

**THE DEVELOPMENT OF MULTI-AGENT SYSTEM FOR BIOLOGICAL  
INFORMATION RETRIEVAL FROM HETEROGENEOUS DISTRIBUTED  
DATA AND KNOWLEDGE SOURCES**

**SAFIE MAT YATIM  
MUHAMAD RAZIB OTHMAN  
SAFAAI DERIS  
ROSLI MD. ILLIAS  
HANY TAHER ALASHWAL**

**FAKULTI SAINS KOMPUTER DAN SISTEM MAKLUMAT  
UNIVERSITI TEKNOLOGI MALAYSIA**

**2006**

## ABSTRAK

Pengagihan, kompleksiti dan kepelbagaian sumber, perisian dan perkakasan adalah cabaran di dalam aplikasi teragih. Agen mobil pintar digunakan untuk menangani data, kawalan, kepakaran dan sumber teragih di dalam aplikasi teragih. Ia adalah atur cara yang berupaya melakukan tindakan berautonomi yang fleksibel, bergerak di antara hos ke hos semasa perlakuan dan menaakul dengan maklumat yang tidak pasti, tidak lengkap dan bertentangan. Kepercayaan-Keinginan-Niat-Bersambung-Mobile UNITY (BDI-MU) kit diperkenalkan untuk membina agen mobil pintar. Ia direka bentuk menggunakan seni bina BDI-MU. Seni bina BDI-MU ialah seni bina terbuka berasaskan taakulan praktikal berautonomi dan keupayaan bersosial menerusi mesej. Ia terdiri daripada pakej sistem dan pakej takrifan pengguna. Pakej sistem menyediakan komponen yang boleh disambung dan diimport untuk membina agen, pelayannya dan domain. Pakej takrifan pengguna ialah komponen boleh dipalam masuk yang terdiri daripada aplikasi pengguna, agen dan pangkalan data. Agen BDI-MU yang dibina oleh BDI-MU kit dibentuk oleh stuktur intra dan inter agen. Struktur intra agen menyediakan kepercayaan, niat, keupayaan dan matlamat untuk taakulan. Ia dibentuk menggunakan autonomi dan seni bina BDI. Struktur inter agen menyediakan fungsi-fungsi untuk berkomunikasi dengan individu lain, menyelaraskan dengan agensi, mencapai sumber dan bergerak di dalam rangkaian. Ia dibentuk menggunakan mesej dan Mobile UNITY. Hirarki ambients dan rangka kerja organisasi holonik digabungkan untuk membina agen BDI-MU yang berkomponen dan berasaskan organisasi. Metodologi dan teknik pemodelan BDI-MU dikemukakan sebagai bahasa pemodelan, peraturan dan amalan untuk menjamin proses pembangunan sistem berasaskan agen BDI-MU yang sistematik. Keputusan ujian menunjukkan perlakuan agen BDI-MU yang cekap menggunakan BDI-MU kit.

## ABSTRACT

Distribution, complexity, heterogeneity of resources, software, and hardware are challenges in distributed applications. Intelligent mobile agent is used to handles distributed data, control, expertise, and resources in distributed applications. It is an executing program that is capable of flexible autonomous action, migrates during execution from host to host, reason with uncertain, incomplete, and contradictory information. Belief-Desire-Intention-Joint-Mobile UNITY (BDI-MU) kit is introduced to construct intelligent mobile agent. It is designed using BDI-MU architecture. BDI-MU architecture is an open architecture based on autonomous practical reasoning and social ability through messages. It consists of system package and user-defined package. The system package provides components that can be extended and imported to develop an agent, its server and domain. The user-defined package is a pluggable component which consists of user applications, agents and databases. The BDI-MU agent constructed by BDI-MU kit is composed by intra agent structure and inter agent structure. The intra agent structure provides belief, intention, capability, and goal for reasoning. Autonomy and BDI architecture is used to construct the intra agent structure. The inter agent structure provides functionality for communicating with other individuals, coordinating with agencies, accessing resources, and mobility within the network. Message and Mobile UNITY is used to construct the inter agent structure. Ambients hierarchy and holonic organization framework is integrated to develop a component and organizational based intelligent mobile agent. To ensure a systematic BDI-MU agent based system development process, a BDI-MU methodology and BDI-MU modeling technique are proposed as a modeling language, rules, and practices. The experimental results show that the BDI-MU agent produced by the BDI-MU kit performed efficiently.

## KANDUNGAN

<b>BAB</b>	<b>PERKARA</b>	<b>MUKA SURAT</b>
	<b>ABSTRAK</b>	ii
	<b>ABSTRACT</b>	iii
	<b>KANDUNGAN</b>	iv
	<b>SENARAI JADUAL</b>	xii
	<b>SENARAI RAJAH</b>	xiii
	<b>SENARAI SINGKATAN</b>	xvii
	<b>SENARAI ISTILAH</b>	xxi
	<b>SENARAI LAMPIRAN</b>	xxxii
<b>BAB I</b>	<b>PENDAHULUAN</b>	1
	1.1 Pengenalan	1
	1.1.1 Agen Mobil Pintar	2
	1.1.2 Aplikasi Teragih dan Sistem Pengurusan Inventori Teragih	2
	1.1.3 Metodologi Berorientasikan Agen Mobil Pintar	3
	1.1.4 Teknik Pemodelan Agen Mobil Pintar	3
	1.1.5 Rangka Kerja Agen Mobil Pintar	4
	1.1.6 Seni Bina Agen Mobil Pintar	4
	1.1.7 Struktur Intra Agen Mobil Pintar	5
	1.1.8 Struktur Inter Agen Mobil Pintar	6
	1.1.9 Spesifikasi Formal	6
	1.2 Motivasi	7
	1.3 Skop	8

1.4	Matlamat dan Objektif	8
1.5	Susunan Tesis	10
<b>BAB II</b>	<b>MASALAH KAJIAN DAN CADANGAN PENYELESAIAN</b>	<b>12</b>
2.1	Pengenalan	12
2.2	Huraian Masalah	13
2.2.1	Masalah Lokasi Inventori	13
2.2.2	Masalah Penjejakan Inventori	14
2.2.3	Masalah Peruntukan Inventori	15
2.3	Aplikasi Teragih Berasaskan Agen Mobil Pintar	16
2.3.1	Sistem Pengurusan Rantaian Bekalan Bersepadu	17
2.3.2	Sistem Reka Bentuk Produk-Proses Bersepadu	18
2.4	Sistem Pengurusan Inventori Teragih	20
2.4.1	Cadangan Penyelesaian Menggunakan Agen Mobil Pintar	21
2.5	Ringkasan	22
<b>BAB III</b>	<b>AGEN MOBIL PINTAR: TEORI DAN APLIKASI</b>	<b>23</b>
3.1	Pengenalan	23
3.2	Agen	24
3.2.1	Definisi Agen	24
3.2.2	Pengkelasan Agen	24
3.2.3	Bidang Ilmu Agen	25
3.3	Agen Mobil Pintar	26
3.3.1	Definisi Agen Mobil Pintar	26
3.3.2	Ciri-Ciri Agen Mobil Pintar	27
3.4	Teori Agen Mobil Pintar	28
3.5	Kejuruteraan Perisian Berasaskan Agen Mobil Pintar	30
3.5.1	Metodologi Berorientasikan Agen Mobil Pintar	30
3.5.2	Teknik Pemodelan Agen Mobil Pintar	31
3.5.3	Rangka Kerja Agen Mobil Pintar	32
3.5.4	Seni Bina Agen Mobil Pintar	33

3.5.5	Spesifikasi Formal	35
3.6	Bahasa Pengaturcaraan	35
3.7	Aplikasi Agen Mobil Pintar	36
3.8	Isu dan Masa Depan Agen Mobil Pintar	37
3.9	Ringkasan	41
<b>BAB IV</b>	<b>METODOLOGI KAJIAN</b>	<b>42</b>
4.1	Pengenalan	42
4.2	Reka Bentuk Kajian	42
4.3	Faedah dan Produk Akhir Kajian	45
4.4	Ringkasan	46
<b>BAB V</b>	<b>METODOLOGI BDI-MU: METODOLOGI RINGAN SEREMPAK UNTUK PEMBANGUNAN SISTEM BERASASKAN AGEN MOBIL PINTAR</b>	<b>47</b>
5.1	Pengenalan	47
5.2	Metodologi Ringan dan Kejuruteraan Serempak	48
5.3	Metodologi BDI-MU	48
5.3.1	Fasa Keperluan	52
5.3.2	Fasa Analisa	53
5.3.3	Fasa Reka Bentuk	54
5.3.3.1	Metodologi Pembangunan Seni Bina Agen	56
5.3.4	Fasa Implementasi	58
5.3.5	Fasa Integrasi	59
5.3.6	Fasa Penempatan	60
5.4	Implementasi Metodologi BDI-MU dalam Pembangunan Sistem Pengurusan Inventori Teragih	60
5.5	Ringkasan	62

<b>BAB VI</b>	<b>TEKNIK PEMODELAN BDI-MU: TEKNIK PEMODELAN</b>	<b>63</b>
	<b>AGEN MOBIL PINTAR MENGGUNAKAN BAHASA</b>	
	<b>PEMODELAN TERGABUNG</b>	
6.1	Pengenalan	63
6.2	Bahasa Pemodelan Tergabung	64
6.3	Teknik Pemodelan BDI-MU	65
6.3.1	Pandangan Domain	66
6.3.2	Pandangan Intra Agen	67
6.3.3	Pandangan Inter Agen	68
6.4	Implementasi Teknik Pemodelan BDI-MU dalam Pemodelan Sistem Pengurusan Inventori Teragih	68
6.4.1	Model Domain	68
6.4.1.1	Gambarajah Kes Gunaan	69
6.4.1.2	Gambarajah Jujukan	69
6.4.1.3	Gambarajah Kelas	71
6.4.1.4	Gambarajah Aktiviti	72
6.4.1.5	Peraturan Pemilihan Agen dan Peraturan Meta Agen	72
6.4.1.6	Gambarajah Kelas Agen	73
6.4.2	Model Intra Agen	74
6.4.2.1	Gambarajah Hirarki Matlamat	75
6.4.2.2	Gambarajah Ontologi	75
6.4.2.3	Gambarajah Jujukan Pelan	76
6.4.2.4	Gambarajah Aliran Data Keupayaan	77
6.4.3	Model Inter Agen	77
6.4.3.1	Gambarajah Jujukan Migrasi	78
6.4.3.2	Gambarajah Jujukan Komunikasi	79
6.5	Ringkasan	79

**BAB VII RANGKA KERJA ORGANISASI HOLONIK DAN  
HIRARKI AMBIENTS UNTUK PEMBANGUNAN SENI  
BINA AGEN MOBIL PINTAR** 80

7.1	Pengenalan	80
7.2	Organisasi Agen	81
7.3	Holonik	82
7.4	Ambients	83
7.5	Rangka Kerja Organisasi Holonik	84
	7.5.1 Lapisan Holon	85
	7.5.2 Lapisan Objek	86
	7.5.3 Lapisan Kelas	87
7.6	Hirarki Ambients	88
7.7	Implementasi Rangka Kerja Organisasi Holonik dan Hirarki Ambients dalam Pembangunan Seni Bina Sistem Pengurusan Inventori Teragih	90
7.8	Ringkasan	92

**BAB VIII SENI BINA BDI-MU: SENI BINA TERBUKA AGEN  
MOBIL PINTAR MENGGUNAKAN CORAK SENI BINA  
HETEROGEN DAN MODEL PANDANGAN “4+1”** 93

8.1	Pengenalan	93
8.2	Corak Seni Bina	94
8.3	Pandangan Seni Bina	95
8.4	Seni Bina BDI-MU	96
	8.4.1 Corak Seni Bina Heterogen	98
	8.4.1.1 Agen, Agen Pintar, Agen Mobil dan Agen Mobil Pintar	98
	8.4.1.2 Persekitaran	101
	8.4.1.3 Konfigurasi, Domain dan Pelayan	102
	8.4.2 Model Pandangan “4+1”	104
	8.4.2.1 Pandangan Kes Gunaan	104
	8.4.2.2 Pandangan Lojikal	104



	8.4.2.3 Pandangan Proses	106
	8.4.2.4 Pandangan Implementasi	106
	8.4.2.5 Pandangan Penempatan	107
8.5	Implementasi Seni Bina BDI-MU dalam Pembangunan Seni Bina Sistem Pengurusan Inventori Teragih	108
8.6	Ringkasan	110
<b>BAB IX</b>	<b>STRUKTUR INTRA AGEN BDI-MU MENGGUNAKAN SENI BINA KEPERCAYAAN-KEINGINAN-NIAT DAN AUTONOMI</b>	111
9.1	Pengenalan	111
9.2	Seni Bina Kepercayaan-Keinginan-Niat	112
9.3	Autonomi	113
9.4	Struktur Intra Agen BDI-MU	114
	9.4.1 Model Dunia	115
	9.4.2 Matlamat	115
	9.4.3 Pelan	116
	9.4.4 Struktur Niat	117
	9.4.5 Pentafsir	117
	9.4.6 Pemerhati	118
	9.4.7 Sintaks, Anatomi dan Bentuk Backus-Naur	119
9.5	Autonomi di dalam Struktur Intra Agen BDI-MU	120
	9.5.1 Autonomi Niat, Keupayaan, Kepercayaan dan Matlamat	120
	9.5.2 Automoni Fleksibel	121
9.6	Keputusan Komputasi	121
9.7	Ringkasan	123
<b>BAB X</b>	<b>STRUKTUR INTER AGEN BDI-MU MENGGUNAKAN MESEJ DAN KONSEP BAHASA MOBILE UNITY</b>	125
10.1	Pengenalan	125
10.2	Mesej	126

10.3	Mobile UNITY	127
10.4	Huraian Masalah Struktur Inter Agen BDI-MU	128
10.4.1	Masalah Komunikasi Teragih	128
10.4.2	Masalah Simulasi Teragih	129
10.5	Penyelesaian Masalah Komunikasi Teragih Menggunakan Mesej	130
10.6	Penyelesaian Masalah Simulasi Teragih Menggunakan Konsep Bahasa Mobile UNITY	131
10.7	Keputusan Komputasi	134
10.8	Ringkasan	136
<b>BAB XI</b>	<b>SPESIFIKASI FORMAL STRUKTUR INTRA DAN INTER AGEN BDI-MU MENGGUNAKAN Z</b>	<b>137</b>
11.1	Pengenalan	137
11.2	Bahasa Formal Z	138
11.3	Alatan Z-Eves	139
11.4	Spesifikasi Formal Struktur Intra Agen BDI-MU	141
11.4.1	Kepercayaan, Matlamat dan Tindakan	141
11.4.2	Pelan, Tika Pelan dan Niat	141
11.4.3	Operasi Agen BDI-MU	143
11.4.4	Autonomi	146
11.5	Spesifikasi Formal Struktur Inter Agen BDI-MU	147
11.5.1	Hubungan dan Kerjasama	147
11.5.2	Komunikasi	148
11.6	Ringkasan	149
<b>BAB XII</b>	<b>BDI-MU KIT: ALATAN PEMBANGUNAN AGEN MOBIL PINTAR MENGGUNAKAN JAVA</b>	<b>150</b>
12.1	Pengenalan	150
12.2	Bahasa Java	151
12.3	Alatan BDI-MU kit	152
12.3.1	Kelas-Kelas Domain BDI-MU	152

12.3.2	Kelas-Kelas Pelayan BDI-MU	153
12.3.3	Fail-Fail Konfigurasi BDI-MU	154
12.3.4	Kelas-Kelas Struktur Intra Agen BDI-MU	154
12.3.5	Kelas-Kelas Struktur Inter Agen BDI-MU	158
12.4	Implementasi BDI-MU kit dalam Pembangunan Sistem Pengurusan Inventori Teragih	161
12.5	Keputusan Implementasi BDI-MU kit dalam Pembangunan Sistem Pengurusan Inventori Teragih	168
12.6	Contoh Implementasi BDI-MU kit dalam Lain-Lain Aplikasi	173
12.7	Perbandingan BDI-MU kit dengan Lain-Lain Alatan Pembangunan Agen	173
12.8	Ringkasan	175
<b>BAB XIII KESIMPULAN</b>		176
13.1	Pengenalan	176
13.2	Kesimpulan	176
13.3	Sumbangan Kajian	179
13.4	Kelemahan Kajian dan Cadangan Kerja Masa Hadapan	181
13.5	Penutup	182
<b>SENARAI PENERBITAN</b>		186
<b>SENARAI DOKUMEN RUJUKAN</b>		188
<b>LAMPIRAN</b>		209

## ENARAI JADUAL

NO. JADUAL	TAJUK	MUKA SURAT
1.1	Objektif kajian	9
3.1	Senarai projek berkaitan dengan agen mobil pintar	39
5.1	Hubungan model dengan pandangan seni bina	56
6.1	Peraturan pemilihan agen	66
6.2	Peraturan meta agen	67
6.3	Contoh agen yang dikenal pasti menggunakan peraturan pemilihan agen	73
8.1	Interaksi di antara proses dalam seni bina BDI-MU	97
9.1	Maklumat bagi data-data sistem pengurusan inventori teragih	122
10.1	Maklumat bagi struktur mesej	131
10.2	Maklumat bagi data-data sistem pengurusan inventori teragih	135
12.1	Ringkasan kelas-kelas domain BDI-MU	153
12.2	Ringkasan fail-fail konfigurasi BDI-MU	154
12.3	Ringkasan kelas-kelas komponen komunikasi dan mobiliti	159
12.4	Operasi asas agen BDI-MU	160
12.5	Persekitaran pengujian	168
12.6	Maklumat bagi data-data sistem pengurusan inventori teragih	169
12.7	Contoh penyelesaian yang dicadangkan oleh agen penjejak stok	170
12.8	Kelakuan agen penjejak stok dengan aras autonomi yang berbeza	171
12.9	Perbandingan BDI-MU kit dengan lain-lain alatan pembangunan agen	174
13.1	Sumbangan kajian	180

## SENARAI RAJAH

NO. RAJAH	TAJUK	MUKA SURAT
1.1	Garis kasar tesis	11
2.1	Struktur stor Tentera Laut DiRaja Malaysia	13
2.2	Pepohon perduaan masalah peruntukan inventori	15
2.3	Pengumpulan dan pengagihan inventori	16
2.4	Fungsi pengurusan rantaian bekalan	18
2.5	Agen ISCM	18
2.6	Pemetaan ruang dalam RAPPID	19
2.7	Agen RAPPID	19
2.8	Sampel rangkaian permintaan dan bekalan inventori	20
2.9	Seni bina sistem pengurusan inventori teragih	21
2.10	Agen mobil pintar dalam sistem pengurusan inventori teragih	22
3.1	Interaksi agen dengan persekitarannya	24
3.2	Pengelasan agen	25
3.3	Bidang ilmu yang berkaitan dengan agen	26
3.4	Perbandingan agen mobil dengan agen mobil pintar dalam perniagaan elektronik	27
3.5	Fasa dalam Tropos	30
3.6	Fasa dalam MAS-CommonKADS	31
3.7	Model dalam AOAD	32
3.8	Rangka kerja Brainstorm/J	32
3.9	Seni bina OAA	33
3.10	Seni bina BDIM	34
3.11	Seni bina INTERRAP	34
3.12	Kos pembangunan perisian dengan spesifikasi formal	35

3.13	Elemen pembangunan agen mobil pintar	38
4.1	Reka bentuk kajian	43
4.2	Produk kajian	45
5.1	Implementasi kejuruteraan serempak ke atas metodologi BDI-MU menggunakan model air terjun	50
5.2	Implementasi kejuruteraan serempak ke atas metodologi BDI-MU menggunakan model putaran	50
5.3	Metodologi BDI-MU	51
5.4	Aktiviti dalam fasa keperluan	52
5.5	Aktiviti dalam fasa analisa	53
5.6	Model-model dalam teknik pemodelan BDI-MU	54
5.7	Aktiviti dalam fasa reka bentuk	55
5.8	Model pandangan “4+1”	56
5.9	Proses pembangunan seni bina agen	58
5.10	Aktiviti dalam fasa implementasi	58
5.11	Aktiviti dalam fasa integrasi	59
5.12	Aktiviti dalam fasa penempatan	60
5.13	Model $i^*$ untuk pengurusan inventori	61
6.1	Teknik pemodelan BDI-MU	65
6.2	Gambarajah kesgunaan sistem pengurusan inventori teragih	69
6.3	Gambarajah jujukan sistem pengurusan inventori teragih	70
6.4	Gambarajah kelas sistem pengurusan inventori teragih	71
6.5	Gambarajah aktiviti sistem pengurusan inventori teragih	72
6.6	Gambarajah kelas agen sistem pengurusan inventori teragih	74
6.7	Gambarajah hirarki matlamat sistem pengurusan inventori teragih	75
6.8	Gambarajah ontologi sistem pengurusan inventori teragih	76
6.9	Gambarajah jujukan pelan sistem pengurusan inventori teragih	76
6.10	Gambarajah aliran data keupayaan sistem pengurusan inventori teragih	77
6.11	Gambarajah jujukan migrasi sistem pengurusan inventori teragih	78
6.12	Gambarajah jujukan komunikasi sistem pengurusan inventori teragih	79
7.1	Organisasi agen	81

7.2	Struktur organisasi agen	82
7.3	Struktur holon dalam sistem kawalan holonik	83
7.4	Rangka kerja organisasi holonik	85
7.5	Lapisan holon	86
7.6	Lapisan objek	87
7.7	Lapisan kelas	88
7.8	Hirarki ambients	88
7.9	Hirarki dan migrasi agen mobil pintar	89
7.10	Transformasi sistem pengurusan inventori teragih	90
7.11	Transformasi agen broker	91
7.12	Struktur agen mobil pintar penjejak stok	91
8.1	Corak seni bina	94
8.2	Seni bina BDI-MU	98
8.3	Struktur agen	99
8.4	Struktur agen pintar	100
8.5	Struktur agen mobil	101
8.6	Struktur persekitaran	102
8.7	Struktur domain	103
8.8	Struktur pelayan	103
8.9	Gambarajah kes gunaan seni bina BDI-MU	104
8.10	Gambarajah pakej seni bina BDI-MU	105
8.11	Gambarajah kelas proses seni bina BDI-MU	106
8.12	Gambarajah komponen seni bina BDI-MU	107
8.13	Gambarajah penempatan seni bina BDI-MU	108
8.14	Seni bina sistem pengurusan inventori teragih	109
9.1	Seni bina kepercayaan-keinginan-niat	113
9.2	Saringan autonomi agen mobil pintar	114
9.3	Graf kesan pusingan pentafsir BDI ke atas masa larian	122
9.4	Graf kesan saringan autonomi ke atas masa larian	123
10.1	Protokol komunikasi KQML	126
10.2	Pergerakan agen mobil pintar berasaskan konsep bahasa Mobile UNITY	128
10.3	Orientasi objek teragih	129
10.4	Penyelesaian masalah simulasi teragih menggunakan konsep	132

	bahasa Mobile UNITY	
10.5	<i>Agen</i> berupaya mengubah lokasi bagi membolehkannya melawati setiap hos sasaran secara pusingan robin	133
10.6	Graf kesan pemindahan agen BDI-MU ke atas masa larian	135
10.7	Graf kesan hebahan mesej ke atas masa larian	135
11.1	Mengompos dan mengedit spesifikasi Z	139
11.2	Menyemak dan membuktikan spesifikasi Z	140
12.1	Kelas-kelas domain BDI-MU	153
12.2	Kelas pelayan BDI-MU	154
12.3	Kelas-kelas komponen model dunia	155
12.4	Kelas-kelas komponen matlamat	155
12.5	Kelas-kelas komponen pelan dan pemerhati	156
12.6	Kelas-kelas komponen struktur niat dan pentafsir	157
12.7	Gelung utama pentafsir BDI	157
12.8	Kelas-kelas komponen komunikasi dan mobiliti	158
12.9	Menghantar mesej	160
12.10	Memindahkan agen BDI-MU	161
12.11	Paparan konfigurasi BDI-MU	164
12.12	Paparan domain BDI-MU	164
12.13	Paparan pelayan BDI-MU	165
12.14	Paparan skrin utama sistem pengurusan inventori teragih	165
12.15	Paparan menginput keperluan kepada agen penjejak stok	166
12.16	Paparan meluluskan permohonan stok yang dibuat oleh agen penjejak stok	166
12.17	Paparan perlaksanaan pentafsir BDI	167
12.18	Paparan penyelesaian yang dicadangkan oleh agen penjejak stok	167
12.19	Graf kesan bilangan larian agen penjejak stok ke atas masa larian	172
13.1	Ringkasan perjalanan kajian	183



## SENARAI SINGKATAN

ACL	-	Agent Communication Language
AHA	-	Asynchronous Hierarchical Agents
AIS	-	Adaptive Intelligent Systems
AOAD	-	Agent-Oriented Analysis and Design
AOP	-	Agent-Oriented Programming
API	-	Application Programming Interface
APL	-	Applicable Plan List
ARCHON	-	Architecture for Cooperating Heterogeneous Online Systems
ATCI	-	Agent Transfer and Communication Interface
AUML	-	Agent Unified Modeling Language
AWIC	-	Agents, World, Interoperability and Coordination
BDI	-	Belief-Desire-Intention
BDIM	-	Belief-Desire-Intention and Message
BDI-MU	-	Belief-Desire-Intention-Joint-Mobile UNITY
BDPAgent	-	Belief, Desire and Plan Agent
BNF	-	Backus-Naur Form
CIDRE	-	Cooperative and Interactive Document Reverse Engineering
CMIPAgent	-	Common Management Information Protocols Agent
CoMoMAS	-	Conceptual Modeling of Multi-Agent Systems
CORBA	-	Common Object Request Broker
CPU	-	Central Processing Unit
DAIS	-	Domain-Adaptive Information Systems

DECAF	-	Distributed Environment Centered Agent Framework
DESIRE	-	Design and Specification of Interacting Reasoning Components
DIA	-	Drivers Information Assistance
DIM	-	Distributed Information Management
dMARS	-	Distributed Multi-Agent Reasoning System
DSDM	-	Dynamic Systems Development Methodology
FIPA	-	Foundation for Intelligent Physical Agents
GRATE	-	Generic Rules and Agent Model Testbed Environment
GVT	-	Global Virtual Time
HCS	-	Holonic Control System
HERMES	-	Heterogeneous Reasoning and Mediator System
HL7	-	Health Level 7
HMS	-	Holonic Manufacturing System
ICL	-	Interagent Communication Language
INTERRAP	-	Integration of Reactive Behavior and Rational Planning
IP	-	Internet Protocol
IRMA	-	Intelligent Resource-Bounded Machine Architecture
ISCM	-	Integrated Supply Chain Management
ITP	-	Institut Teknologi Perisian
JADE	-	Java Agent Development Framework
JAFMAS	-	Java-Based Framework for Multi-Agent System
JAM	-	Java Agent for Meta-Learning
JAT	-	Java Agent Template
JDBC	-	Java Database Connectivity
JDK	-	Java Development Kit
JIAC	-	Java Intelligent Agent Componentware

Jinni	-	Java Inference Engine and Networked Interactor
JVM	-	Java Virtual Machine
KAoS	-	Knowledgeable Agent-Oriented System
KIF	-	Knowledge Interchange Format
KQML	-	Knowledge Query and Manipulation Language
KSE	-	Knowledge Sharing Effort
LAN	-	Local Area Network
LIME	-	Linda in a Mobile Environment
MACRON	-	Multi-Agent Architecture for Cooperative Retrieval Online
MAF	-	Mobile Agent Framework
MAgNET	-	Multi-Agent Negotiation Testbed
MARS	-	Mobile Agent Reactive Space
MAS	-	Multi-Agent Systems
MaSE	-	Multi-Agent Software Engineering
MobiDoc	-	Mobile Documents
NIC	-	Network Interface Card
NP-complete	-	Non-Polynomial-complete
OAA	-	Open Agent Architecture
OASIS	-	Open Architecture for Secure Interworking Services
OCL	-	Object Constraint Language
ODBC	-	Open Database Connectivity
OMG	-	Object Management Group
OMT	-	Object Modeling Technique
OOSE	-	Object-Oriented Software Engineering
OS	-	Operating System
PLACA	-	Planning Communicating Agents
PRS	-	Procedural Reasoning System
RAM	-	Random Access Memory
RAPPID	-	Responsible Agents for Product-Process Integrated Design

RETSINA	-	Reusable Environment for Task Structured Intelligent Network Agents
RMN	-	Royal Malaysian Navy
RPC	-	Remote Procedure Call
RUP	-	Rational Unified Process
TCP	-	Transmission Control Protocol
TLDM	-	Tentera Laut DiRaja Malaysia
TSP	-	Traveling Salesman Problem
UML	-	Unified Modeling Language
UM-PRS	-	University of Michigan Procedural Reasoning System
UTM	-	Universiti Teknologi Malaysia
WSS	-	Wave Secure System

## SENARAI ISTILAH

Adik-beradik	-	Sibling
Agen	-	Agent
Air terjun	-	Waterfall
Aksiom	-	Axiom
Aktiviti	-	Activity
Alat / perkakas	-	Harness
Alatan	-	Tools
Algoritma genetik	-	Genetic algorithms
Aliran kerja	-	Workflow
Anatomi	-	Anatomy
Antara muka	-	Interface
Aplikasi	-	Application
Aras / paras	-	Level
Arbitrari	-	Arbitrary
Argumen	-	Argument
Artifak	-	Artifact
Atomik	-	Atomic
Atribut	-	Attribute
Atur cara	-	Program
Autonomi / berautonomi	-	Autonomy / autonomous
Badan	-	Body
Bahagian	-	Part
Bait	-	Byte
Bajet	-	Budget
Baris	-	Tier
Bas	-	Bus

Bawah-atas	-	Bottom-up
Bebenang	-	Thread
Benar	-	Veracity
Berat	-	Heavyweight
Berpusat	-	Centralized
Bio-informatik	-	Bio-informatics
Boleh diguna semula	-	Reusable
Boleh dipercayai	-	Reliable
Boleh guna / bersesuaian	-	Applicable
Boleh laku / perlakuan	-	Executable / execution
Broker	-	Broker
Cabang	-	Branch
Campur tangan	-	Intervention
Capai / capaian	-	Achieve / access
Carian	-	Searching
Cetusan	-	Trigger
Corak	-	Pattern
Dalam talian	-	Online
Delegasi	-	Delegation
Deria / sensor	-	Sensor
Dinamik	-	Dynamic
Direktori	-	Directory
Diskret	-	Discrete
Domain	-	Domain
Dunia nyata	-	Real world
Efek	-	Effect
Eksplisit	-	Explicit
Ekstrinsik	-	Extrinsic
Fasilitator	-	Facilitator
Fizikal	-	Physical
Formulasi	-	Formulation
Gabungan	-	Integration
Gaya	-	Style
Gelung	-	Loop

Generik	-	Generic
Genom	-	Genome
Hapus	-	Dispose
Hebahan	-	Broadcast
Henti	-	Stop
Heterogen	-	Heterogeneous
Hibrid	-	Hybrid
Hindar	-	Avoid
Hirarki	-	Hierarchy
Holon / holonik	-	Holon / Holonic
Homeostatik	-	Homeostatic
Horizontal	-	Horizontal
Hos	-	Host
Hubungan	-	Relation
Ibu bapa	-	Parent
Implisit	-	Implicit
Inferens	-	Inference
Ingatan	-	Memory
Inter operasi	-	Inter operational
Inter sambungan	-	Interconnection
Interaksi	-	Interaction
Intrinsik	-	Intrinsic
Invokasi	-	Invocation
Isih	-	Sort
Isytihar	-	Declare
Jadual cincangan	-	Hash table
Jadual	-	Table
Jalan	-	Run
Jana	-	Generate / invoke
Jujukan	-	Sequence
Kata putus	-	Decision
Kawalan	-	Control
Keadaan	-	State
Kebersandaran	-	Dependency

Kebetulan	-	Correctness
Kecacatan	-	Defect
Kefungsian	-	Functional
Kejuruteraan perisian	-	Software engineering
Kekangan	-	Constraint
Kekuasaan	-	Authority
Kelakuan	-	Behavior
Kelas	-	Class
Kependaman	-	Latency
Kepercayaan	-	Belief
Keperluan	-	Requirements
Kepintaran buatan	-	Artificial intelligence
Kerahsiaan	-	Privacy
Kerjasama	-	Collaboration
Kernel	-	Kernel
Kesgunaan	-	Use case
Kesedaran	-	Awareness
Kesegaran	-	Fitness
Keselamatan	-	Security
Kesusilaan	-	Ethical
Ketegapan	-	Robustness
Ketepatan	-	Accuracy
Ketersediaan	-	Availability
Keupayaan	-	Capability
Kod pseudo	-	Pseudo code
Kompil	-	Compile
Komponen	-	Component
Kompos	-	Compose
Komputer buku	-	Notebook computer
Komputer meja	-	Desktop computer
Komputer tapak tangan	-	Palmtop computer
Komunikasi	-	Communication
Konfigurasi	-	Configuration
Kontena	-	Container



Kotak hitam	-	Blackboard
Kumpulan	-	Group
Lalai	-	Default
Langganan	-	Bespoke
Lapis	-	Layer
Lebar jalur	-	Bandwidth
Lelaran	-	Iteration
Lengah	-	Delay
Lepasan	-	Release
Lojikal	-	Logical
Masa larian	-	Runtime
Masalah sukar	-	Hard problem
Masa-nyata	-	Real-time
Matlamat	-	Goal
Maya	-	Virtual
Meja kerja	-	Workbench
Melombong	-	Mining
Menaakul	-	Reasoning
Menset	-	Setting
Mental	-	Mental
Menyahkod	-	Decode
Menyambung semula	-	Resume
Mesej	-	Message
Meta	-	Meta
Metodologi	-	Methodology
Migrasi	-	Migration
Mobil / bergerak	-	Mobile
Modaliti	-	Modality
Model dunia	-	World model
Monolitik	-	Monolithic
Motivasi	-	Motivation
Muat turun	-	Download
Mudah dialih	-	Portable
Mudah disambung	-	Extendable

Mudah disenggara	-	Maintainable
Mudah suai / penyesuaian	-	Adaptive
Niat	-	Intention
Objek	-	Object
Omnisains	-	Omniscience
Ontologi	-	Ontology
Operasi	-	Operation
Organisasi	-	Organization
Pakej / bungkusan	-	Package
Palam keluar	-	Plug-out
Palam masuk	-	Plug-in
Pandangan	-	View
Panggilan	-	Call
Pasang	-	Install
Pelaku	-	Actor
Pelan / perancangan	-	Plan
Pelanggan	-	Client
Pelayan	-	Server
Pelet	-	Palette
Pelihara	-	Maintain
Pemandu	-	Driver
Pemasa	-	Timer
Pembelajaran	-	Learning
Pemberi kesan	-	Effectors
Pemberi sandaran	-	Depender
Pembolehkan	-	Variable
Pemecut	-	Accelerator
Pemerhati / pengamatan	-	Observer / observation
Pemilihan	-	Selection
Pemuat	-	Loader
Pemulihan	-	Recovery
Pendengar	-	Listener
Penempatan	-	Deployment
Penemu	-	Finder

Penerapan	-	Assertions
Penerima sandaran	-	Dependee
Penerimaan	-	Acceptance
Penerokaan	-	Discovery
Pengalamatan	-	Addressing
Pengamir	-	Integrator
Pengaruh	-	Influence
Pengecaman	-	Identification
Pengelompokan	-	Clustering
Pengetahuan	-	Knowledge
Penggantian	-	Substitution
Penghalusan	-	Refinement
Penghimpun	-	Assembler
Pengikatan	-	Binding
Pengkomputeran	-	Computing
Pengsirian	-	Serialization
Pengujian / pemeriksaan	-	Testing
Pengukuran	-	Measurement
Pengumpulan	-	Aggregation
Penimbal	-	Buffer
Penjadualan	-	Scheduling
Penjejakan	-	Tracking
Penjelajahan	-	Traveling
Penseketikaan	-	Instantiation
Pentafsir	-	Interpreter
Penunaian	-	Fulfillment
Penuturan	-	Conversation / speech
Penyelarasan	-	Coordination
Penyemak seimbis	-	Browser
Pepohon	-	Tree
Peranan	-	Role
Peraturan	-	Rule
Perbuatan / tindakan	-	Act / action
Perduaan / binari	-	Binary

Peringkasan	-	Abstraction
Peristiwa	-	Event
Perkongsian	-	Sharing
Perlaksanaan	-	Performance
Permulaan / awalan	-	Initially / initial
Pernyataan	-	Expression / proposition
Persekitaran	-	Environment
Perundangan	-	Legal
Perundingan / perbincangan	-	Negotiation
Peruntukan	-	Allocation
Perwakilan	-	Representation
Perwarisan	-	Inheritance
Piawaian	-	Standard
Pilihan	-	Option
Pintar / cerdas	-	Intelligent
Port	-	Port
Pos	-	Post
Praktikal	-	Practical
Predikat	-	Predicate
Proaktif	-	Proactive
Protokol	-	Protocol
Pusingan robin	-	Round robin
Pustaka	-	Library
Putaran	-	Spiral
Ramah pengguna	-	User friendly
Rangka kerja	-	Framework
Rangkaian neural	-	Neural network
Rangkaian	-	Network
Rasional	-	Rational
Reaktif	-	Reactive
Realiti maya	-	Virtual reality
Reka bentuk	-	Design
Relevan	-	Relevant
Rencaman	-	Composition

Rentetan	-	String
Ringan	-	Lightweight
Robotik	-	Robotic
Ruang	-	Space
Rutin	-	Routine
Sains gunaan	-	Applied science
Saling bekerja	-	Inter working
Saling tukar	-	Interchange
Sampukan	-	Interrupt
Sarang	-	Nest
Saring / tapis	-	Filter
Segerak	-	Synchronize
Sekitar	-	Ambients
Sekutuan	-	Association
Selari	-	Parallel
Semakan	-	Revision
Semantik	-	Semantic
Semasa	-	Current
Sementara	-	Transient
Senario	-	Scenario
Seni bina	-	Architecture
Sepunya	-	Common
Serempak / serentak	-	Concurrent
Set kasar	-	Rough set
Setempat / lokal	-	Local
Siber	-	Cyber
Sifat	-	Characteristic / attitude
Simulasi	-	Simulation
Sintaks	-	Syntax
Sintesis	-	Synthesis
Siri	-	Serial
Sistem pelbagai agen	-	Multi-agent system
Sistem pengoperasian	-	Operating system
Situasi	-	Situation

Skima	-	Scheme
Slot	-	Slot
Soket	-	Socket
Spatial	-	Spatial
Spesifikasi formal	-	Formal specification
Storan	-	Storage
Strim	-	Stream
Struktur data	-	Data structure
Struktur inter	-	Inter structure
Struktur intra	-	Intra structure
Suap balik	-	Feedback
Sumber	-	Resource
Sunting	-	Edit
Superkomputer	-	Supercomputer
Taakulan berasaskan kes	-	Case based reasoning
Talian paip	-	Pipeline
Tanda aras	-	Benchmark
Tanggapan / cerapan	-	Perception
Tanpa wayar	-	Wayerless
Tanya	-	Query
Tapak	-	Site
Tatacara / prosidur	-	Procedure
Teknik pemodelan	-	Modeling technique
Tem	-	Term
Tempat simpanan / gudang	-	Repository
Templat	-	Template
Temporal	-	Temporal
Teragih	-	Distributed
Terbuka	-	Open
Tergabung	-	Unified
Tidak segerak	-	Asynchronous
Tidak varian	-	Invariant
Tika	-	Instance
Tindakan penuturan	-	Speech acts

Tindakan	-	Stack
Titik ke titik	-	Point-to-point
Tolakan	-	Deduction
Transformasi / penterjemahan	-	Transformation
Transisi	-	Transition
Transmisi	-	Transmission
Truput	-	Throughput
Tugas	-	Task
Umpuk	-	Assign
Utiliti / kemudahan	-	Utility
Validasi	-	Validation
Variasi	-	Variation
Verifikasi	-	Verification
Versi	-	Version
Vertikal	-	Vertical

**SENARAI LAMPIRAN**

<b>LAMPIRAN</b>	<b>TAJUK</b>	<b>MUKA SURAT</b>
A	Notasi struktur intra agen BDI-MU	209
B	Bentuk Backus-Naur struktur intra agen BDI-MU	210
C	Notasi bahasa Mobile UNITY	213
D	Notasi bahasa Z	214



# **BAB I**

## **PENDAHULUAN**

### **1.1 Pengenalan**

Agen merupakan atur cara yang bekerja bagi pihak penggunanya dan digunakan secara meluas dalam pelbagai aplikasi, bermula daripada tugas-tugas pemprosesan mudah dan penapisan mel elektronik, kepada aplikasi tanpa wayar dan pembantu peribadi pintar, seterusnya kepada sistem yang besar, kompleks dan kritikal seperti kawalan trafik udara, pemantauan pesakit dan pengurusan reaktor kuasa (Jennings et al., 1998). Contoh agen yang telah berjaya dibangunkan adalah seperti RETSINA (Sycara dan Dajun, 1996), MAgNET (Dasgupta et al., 1999) dan Concordia (Koblick, 1999). Bagaimanapun, persekitaran aplikasi yang semakin canggih, teragih dan dinamik memerlukan sokongan agen yang inter operasi, rasional dan berautonomi untuk mengoptimumkan sumber dan berupaya menjalankan pelbagai tugas. Bagi memastikan agen berfungsi secara efektif, seni bina agen mestilah fleksibel, generik dan boleh dipercayai. Oleh itu, kajian ini dijalankan untuk menghasilkan seni bina terbuka untuk agen mobil pintar yang beroperasi di dalam aplikasi teragih. Agen mobil pintar merupakan salah satu klasifikasi agen yang berupaya bergerak dari mesin ke mesin melalui rangkaian, proaktif dan reaktif.

Bab ini memperkenalkan subjek yang menjadi fokus kajian iaitu agen mobil pintar, aplikasi teragih dan sistem pengurusan inventori teragih, metodologi berorientasikan agen mobil pintar, teknik pemodelan agen mobil pintar, rangka kerja agen mobil pintar, seni bina agen mobil pintar, struktur intra agen mobil pintar,

struktur inter agen mobil pintar dan spesifikasi formal. Motivasi, skop dan objektif kajian juga dijelaskan dalam bab ini.

### **1.1.1 Agen Mobil Pintar**

Agen mobil pintar merupakan agen hibrid yang menggabungkan falsafah agen mobil dengan falsafah agen pintar. Agen ini mempunyai keupayaan menukar kedudukan posisinya, berkomunikasi dengan entiti lain dan berautonomi dalam membuat keputusan (Gavalas et al., 2000). Motivasinya adalah untuk menghasilkan agen yang mampu bergerak semasa perlakuan dari mesin ke mesin dalam rangkaian yang heterogen, mencapai maklumat yang teragih (spatial, temporal, semantik atau kefungsiian) bagi pihak penggunaanya dan menaakul dengan maklumat yang tidak lengkap, tidak pasti dan bertentangan.

### **1.1.2 Aplikasi Teragih dan Sistem Pengurusan Inventori Teragih**

Aplikasi teragih ditakrifkan sebagai aplikasi yang terdiri daripada beberapa komponen yang saling tidak bergantung tetapi bekerjasama untuk mencapai matlamat sepunya (Cheriton, 1988). Ciri-ciri aplikasi teragih adalah seperti berikut:

- (a) Sumber data dan maklumat yang heterogen.
- (b) Topologi dan konfigurasi rangkaian yang sukar diramal dan dinamik.
- (c) Keperluan yang berlainan di antara pemilik dengan pengguna sumber.
- (d) Sekatan hubungan dari aspek keselamatan dan protokol komunikasi.

Sistem pengurusan inventori teragih dikelaskan sebagai aplikasi teragih. Sistem ini terdiri daripada satu set stor yang berada di lokasi geografi berbeza tetapi saling berhubungan melalui rangkaian. Setiap stor mempunyai inventori untuk menimbal permintaan luar jangka atau mengekalkan kebolehpayaan bekalan. Oleh kerana kos inventori termasuk kos penyimpanannya adalah mahal, maka mengekalkan paras optimal dan kebolehejakan inventori pada rangkaian stor adalah

penting. Keupayaan agen mobil pintar bergerak di sepanjang rangkaian yang heterogen dan berkemampuan melakukan perundingan bagi pihak penggunanya adalah penyelesaian kepada masalah penggunaan dan penjejakan inventori teragih.

### **1.1.3 Metodologi Berorientasikan Agen Mobil Pintar**

Agen mobil pintar seperti Jinni (Tarau dan Dahl, 2001), Plangent (Ohsuga et al., 1997) dan Colony (Dorigo et al., 1996) telah digunakan untuk mengendalikan operasi dan persekitaran aplikasi yang canggih, besar dan kompleks. Senario ini memerlukan metodologi yang dapat memastikan pengeluaran agen mobil pintar yang boleh dipercayai, mudah disenggara dan mudah diubah suai. Ini kerana proses menghasilkan agen mobil pintar memerlukan 99% teknik kejuruteraan perisian sebagai panduan dan amalan untuk membina agen mobil pintar dan 1% teknik kepintaran buatan untuk menjadikannya perisian yang cerdas (Etzioni, 1997). Oleh itu, metodologi berorientasikan agen mobil pintar merupakan panduan penting untuk membantu pembangun sistem menghasilkan aplikasi berasaskan agen mobil pintar secara sistematik termasuk pengurusan bagi mengurangkan risiko dan ketidakfungsian agen mobil pintar. Metodologi ini harus menerangkan secara terperinci bagaimana agen mobil pintar dispesifikasikan, bagaimana spesifikasi tersebut dijemlakan kepada implementasi yang efisien serta bagaimana agen mobil pintar dan aplikasinya dibuktikan bagi memastikan spesifikasinya dipenuhi mengikut keperluan kejuruteraan perisian berasaskan agen (Wooldridge, 1997).

### **1.1.4 Teknik Pemodelan Agen Mobil Pintar**

Teknik pemodelan objek (*object modeling technique*, OMT) oleh Rumbaugh (1995) merupakan metodologi yang memodelkan sistem berorientasikan objek berdasarkan tiga pandangan yang berlainan tetapi saling berkaitan menerusi model objek, model dinamik dan model kefungsian. Oleh kerana itu, sistem berorientasikan agen mobil pintar juga memerlukan teknik pemodelan yang menyamai OMT dengan

peranan penting untuk mengidentifikasi kelas agen mobil pintar yang terdapat di dalam domain aplikasi dan menghuraikan kelakuan serta hubungannya dengan kelas agen atau kelas objek yang lain. Teknik pemodelan ini meliputi fasa analisa yang melibatkan proses memodelkan domain, mengidentifikasi dan mengklasifikasi agen mobil pintar serta memodelkan agen mobil pintar. Proses pemodelan agen mobil pintar seterusnya dilihat dari tiga sudut pandangan yang berbeza iaitu pandangan domain, pandangan intra agen mobil pintar dan pandangan inter agen mobil pintar.

### **1.1.5 Rangka Kerja Agen Mobil Pintar**

Rangka kerja adalah reka bentuk boleh guna semula sesuatu sistem yang umumnya dijelaskan oleh satu set kelas abstrak dan kerjasama di antara tika kelas tersebut (Johnson dan Foote, 1988). Rangka kerja dibangunkan apabila pelbagai aplikasi dibina untuk domain yang sama supaya kod dan masa dapat dijitamkan. Rangka kerja agen mobil pintar menumpu kepada pembangunan templat yang boleh diguna semula dan disambung untuk membentuk seni bina agen mobil pintar. Rangka kerja ini adalah berasaskan konsep organisasi holonik yang mampu menyelesaikan masalah seni bina agen mobil pintar yang berskala besar dan dapat menyokong pembangunan sistem pelbagai agen. Konsep organisasi merangkumi peraturan organisasi, struktur organisasi dan corak organisasi. Peraturan organisasi menerangkan spesifikasi dan keperluan untuk penseketikaan dan perlakuan agen. Struktur organisasi mentakrifkan komponen organisasi dan hubungan di antara komponen termasuk peranan dan tanggungjawab komponen dalam perancangan dan penyelesaian masalah secara bekerjasama. Corak organisasi pula adalah pratakris struktur organisasi yang boleh diguna semula daripada sistem ke sistem yang lain.

### **1.1.6 Seni Bina Agen Mobil Pintar**

Teori, seni bina dan bahasa memainkan peranan penting dalam pembangunan agen (Wooldridge dan Jennings, 1994). Seni bina merupakan faktor yang

menentukan keterbukaan dan kompleksiti agen (Wooldridge, 1998). Maes (1991) mentakrifkan seni bina agen sebagai:

*“Metodologi khusus untuk membangunkan agen. Ianya menghuraikan bagaimana agen dilerai kepada satu set komponen dan bagaimana komponen-komponen tersebut saling berinteraksi. Keseluruhan komponen dan interaksinya menjelaskan bagaimana deria dan keadaan dalaman agen menentukan tindakan yang akan dilaksanakan. Seni bina agen meliputi teknik dan algoritma yang menyokong metodologi ini.”*

Seni bina membolehkan agen mobil pintar dinilai berasaskan profil (atribut dan operasi), menghurai proses implementasi komponen yang mengurus sesuatu tindakan serta mengidentifikasi data dan aliran kawalan di antara komponen. Salah satu cara untuk menghuraikan seni bina agen mobil pintar adalah melalui corak seni bina dan pandangan seni bina. Corak seni bina menerangkan jenis, konfigurasi dan interaksi di antara komponen yang membentuk sesuatu sistem (Monroe et al., 1993). Pandangan seni bina pula menerangkan ringkasan atribut dan kelakuan komponen dari pelbagai sudut yang berlainan untuk menggambarkan keseluruhan struktur sistem (Plasil dan Stal, 1998).

### **1.1.7 Struktur Intra Agen Mobil Pintar**

Struktur intra agen mobil pintar menerangkan sifat-sifat intrinsik agen mobil pintar sebagai satu perisian secara individu. Sifat-sifat intrinsik tersebut merangkumi matlamat, kepercayaan, keinginan dan keupayaan yang dimiliki oleh agen mobil pintar. Seni bina kepercayaan-keinginan-niat (*belief-desire-intention*, BDI) digabungkan dengan autonomi untuk membentuk struktur intra agen mobil pintar bagi membolehkannya menaakul secara praktikal dan membuat keputusan secara berautonomi. Seni bina BDI (Rao dan Georgeff, 1998) terdiri daripada struktur kepercayaan (maklumat yang dimiliki oleh agen mobil pintar mengenai persekitaran yang mungkin benar atau tidak benar), keinginan (perkara-perkara yang hendak dicapai oleh agen mobil pintar) dan niat (perkara-perkara yang sama ada agen mobil pintar berhasrat untuk melakukannya atau tidak berhasrat melakukannya). Autonomi

pula adalah kemampuan agen mobil pintar melaksanakan sebahagian besar tugas-tugas penyelesaian masalah tanpa campur tangan manusia atau agen lain (Falcone dan Castelfranchi, 2000). Autonomi memberikan agen mobil pintar paras kawalan yang telah ditetapkan ke atas tindakan dan keadaan dalamannya.

### **1.1.8 Struktur Inter Agen Mobil Pintar**

Struktur inter agen mobil pintar menerangkan sifat-sifat ekstrinsik agen mobil pintar sebagai sebahagian daripada komuniti. Sifat-sifat ekstrinsik tersebut melibatkan komunikasi dengan individu lain, penyelarasan dengan agensi, capaian sumber teragih dan penukaran posisi fizikal. Konsep bahasa Mobile UNITY digabungkan dengan mesej untuk membentuk struktur inter agen mobil pintar bagi membolehkannya bergerak dan berkomunikasi. Mobile UNITY (McCann dan Roman, 1998) adalah bahasa untuk menghurai dan menaakul mengenai aplikasi teragih dan serempak yang mengandungi komponen dinamik. Mobile UNITY terdiri daripada beberapa atur cara, seksyen komponen dan seksyen interaksi. Tika atur cara boleh bergerak di sepanjang rangkaian dengan mengumpulkan semula nilai pembolehubah lokasi. Seksyen komponen mentakrifkan tika atur cara yang akan wujud di dalam aplikasi. Seksyen interaksi menghuraikan penyelarasan di antara komponen. Mesej pula adalah elemen komunikasi yang merupakan satu struktur data yang berkemungkinan mengandungi kelakuan agen mobil pintar yang dihantar antara agen mobil pintar melalui medium dan protokol tertentu.

### **1.1.9 Spesifikasi Formal**

Spesifikasi formal ditakrifkan sebagai huraian yang ringkas dan padat mengenai operasi dan atribut sistem yang ditulis dalam bahasa berasaskan matematik untuk menentukan apakah yang sepatutnya dilakukan oleh sistem (Bowen dan Hinchey, 1995). Spesifikasi formal digunakan untuk membuktikan bahawa seni bina agen mobil pintar yang dibangunkan adalah konsisten, lengkap dan betul.

## 1.2 Motivasi

Ledakan kepintaran buatan teragih adalah disebabkan oleh persekitaran pengkomputeran yang terbuka, heterogen, berskala besar dan teragih. Selain itu, aplikasi semasa memerlukan agen untuk bertindak sebagai individu yang mampu membuat perancangan, tugas dan keputusan secara teragih. Keadaan sedemikian menjadi semakin kompleks dengan evolusi sistem pelbagai agen yang memerlukan perkongsian sumber, kawalan teragih, data tidak sentral dan pengkomputeran tidak segerak. Integrasi kedua-dua fenomena ini memerlukan agen mobil pintar yang dapat beroperasi secara berkumpulan (tugas diagihkan, agen bertindak secara individu, mobil dalam domainnya dan bersatu semula), tidak monolitik, pengendalian serempak, berkomunikasi melalui protokol yang piawai dan menaakul mengenai tindakan, pelan dan pengetahuan setempat dan agen lain.

Oleh kerana itu, seni bina agen mobil pintar yang terbuka dengan ciri-ciri fleksibel, generik dan boleh diguna semula diperlukan. Tujuannya ialah untuk memastikan pengeluaran agen mobil pintar yang rasional, berautonomi dan inter operasi di dalam persekitaran teragih yang heterogen dan kompleks dengan mengimport dan menyambung komponen-komponen pintar, autonomi, mobiliti dan komunikasi yang disediakan oleh seni bina tersebut. Walaupun begitu, terdapat persoalan utama yang perlu ditangani bagi merealisasikan idea tersebut, iaitu:

*“Bagaimana menghasilkan seni bina agen mobil pintar yang terbuka untuk aplikasi teragih?”*

Isu-isu yang timbul dari persoalan utama di atas yang perlu dikaji dan diselesaikan dalam kajian ini adalah seperti berikut:

- (a) Apakah metodologi dan teknik pemodelan untuk membangunkan agen mobil pintar dan sistem berasaskannya?
- (b) Apakah rangka kerja, corak seni bina dan pandangan seni bina untuk membangunkan dan menghuraikan seni bina agen mobil pintar?
- (c) Apakah teknik untuk membentuk struktur intra dan inter agen mobil pintar?
- (d) Apakah teknik untuk membuktikan secara formal seni bina agen mobil pintar?

- (e) Apakah teknik untuk mengimplementasi agen mobil pintar serta menggabungkan agen berkenaan ke dalam sistem berasaskannya?

### **1.3 Skop**

Skop yang menjadi batasan dan sempadan dalam melaksanakan kajian ini adalah seperti berikut:

- (a) Kerja-kerja penyelidikan tidak terbatas kepada pembangunan seni bina agen mobil pintar sahaja, tetapi melibatkan kerja-kerja menghasilkan metodologi, alatan pembangunan dan aplikasi berasaskan agen mobil pintar.
- (b) Kajian mengfokus kepada agen dengan sifat mobil, pintar, berautonomi dan berkeupayaan bersosial.
- (c) Pengkhususan domain adalah ke atas aplikasi teragih. Sistem pengurusan inventori teragih yang dibangunkan untuk Tentera Laut DiRaja Malaysia (TLDM) dijadikan kajian kes.
- (d) Tumpuan struktur intra agen mobil pintar adalah ke atas matlamat, kepercayaan, pelan, keupayaan dan autonomi agen mobil pintar. Struktur inter agen mobil pintar pula mengfokus kepada penyelesaian komunikasi dan mobiliti agen mobil pintar.
- (e) Spesifikasi formal menghuraikan atribut, kelakuan dan hubungan di antara komponen-komponen bagi struktur intra dan inter agen mobil pintar.

### **1.4 Matlamat dan Objektif**

Matlamat kajian ini ialah membangunkan seni bina agen mobil pintar untuk aplikasi teragih yang terbuka dengan ciri-ciri mudah disambung, mudah dialih dan mudah diubah suai. Seni bina yang dihasilkan ini dibangunkan menggunakan corak seni bina heterogen dan dihuraikan menggunakan model pandangan “4+1” seperti



yang dibentangkan dalam Bab VIII. Matlamat kajian ini dipenuhi dengan mencapai beberapa objektif seperti yang telah disasarkan pada Jadual 1.1. Pelaksanaan matlamat dan objektif-objektif kajian ini adalah berpandukan kepada metodologi kajian seperti yang dikemukakan dalam Bab IV.

**Jadual 1.1 : Objektif kajian**

<b>No</b>	<b>Objektif</b>	<b>Isu</b>	<b>Teknik</b>	<b>Produk</b>	<b>Bab Berkaitan</b>
1.	Membangunkan metodologi pembangunan agen mobil pintar.	Mengendalikan masalah menganalisa, mereka bentuk dan mengimplementasi agen mobil pintar dan aplikasinya.	Metodologi ringan dan kejuruteraan serempak.	Metodologi BDI-MU.	Bab V dan VI.
2.	Membangunkan struktur intra dan inter agen mobil pintar.	Mengendalikan masalah agen mobil pintar yang berciri teragih, serempak, cerdas dan berupaya bergerak.	Seni bina BDI, autonomi, mesej dan konsep bahasa Mobile UNITY.	Seni bina BDI-MU.	Bab IX dan Bab X.
3.	Membangunkan spesifikasi formal struktur intra dan inter agen mobil pintar.	Mengendalikan masalah konsistensi dan kebenaran reka bentuk seni bina agen mobil pintar.	Bahasa Z.	Spesifikasi formal BDI-MU.	Bab XI.
4.	Membangunkan alatan pembangunan agen mobil pintar.	Mengendalikan masalah membangunkan aplikasi teragih berasaskan agen mobil pintar dalam persekitaran pengkomputeran teragih dan heterogen.	Bahasa Java.	BDI-MU kit.	Bab XII.
5.	Membangunkan aplikasi berasaskan agen mobil pintar.	Mengendalikan masalah menguji keupayaan dan kemampuan agen mobil pintar.	BDI-MU kit.	Sistem pengurusan inventori teragih TLDM.	Bab XII.

Terdapat lima objektif yang akan dicapai di dalam kajian ini, sebagaimana berikut:

- (a) Membangunkan metodologi pembangunan agen mobil pintar menggunakan metodologi ringan dan kejuruteraan serempak.
- (b) Membangunkan struktur intra dan inter agen mobil pintar menggunakan seni bina BDI, autonomi, mesej dan konsep bahasa Mobile UNITY.
- (c) Membangunkan spesifikasi formal struktur intra dan inter agen mobil pintar menggunakan bahasa Z.
- (d) Membangunkan alatan pembangunan agen mobil pintar menggunakan bahasa Java.
- (e) Membangunkan aplikasi berasaskan agen mobil pintar menggunakan BDI-MU kit.

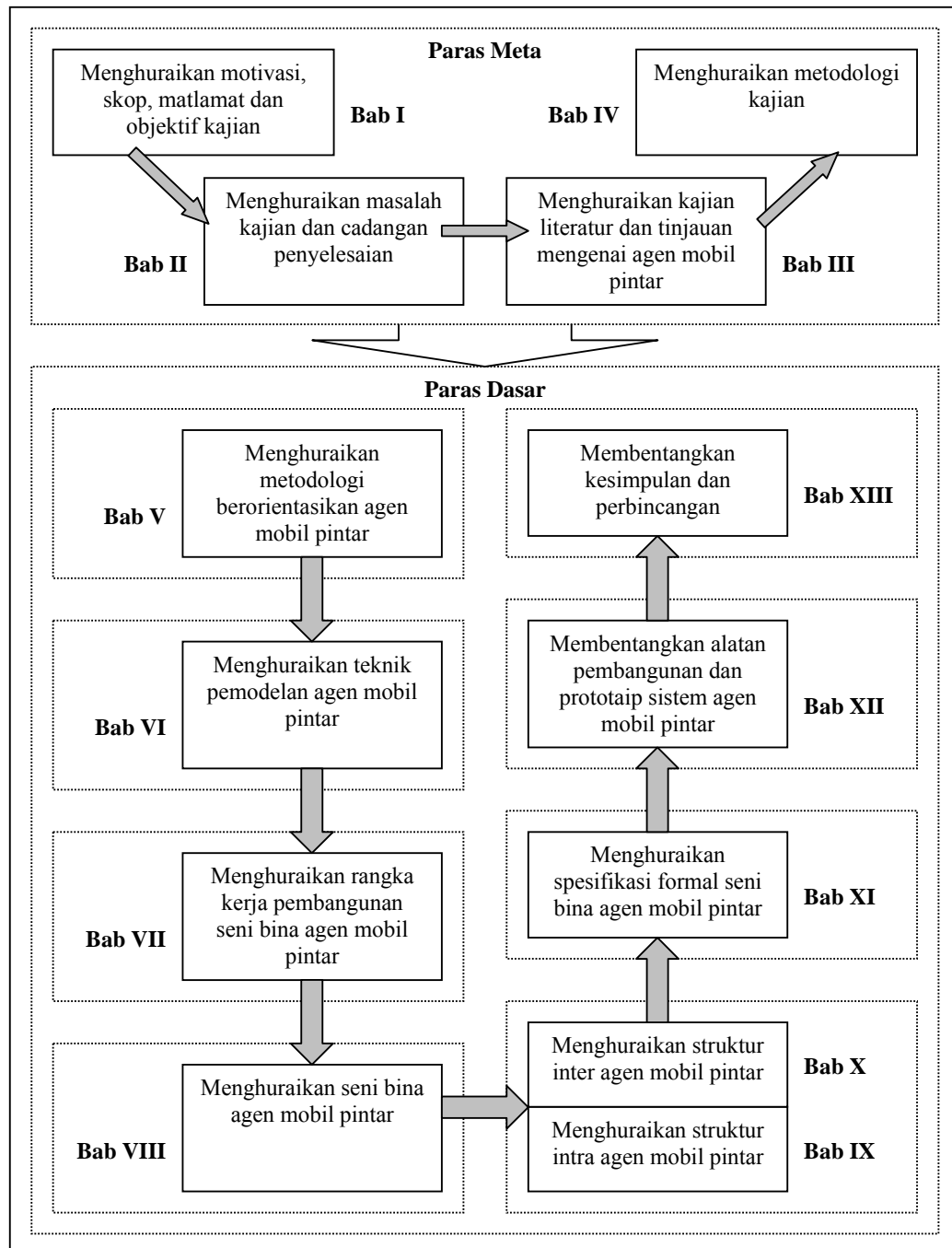
## 1.5 Susunan Tesis

Tesis ini diorganisasi kepada tiga belas bab seperti pada Rajah 1.1. Bab I sehingga Bab IV menghuraikan latar belakang kajian, masalah aplikasi teragih, sebab agen mobil pintar digunakan untuk menangani masalah aplikasi teragih, tinjauan mengenai agen mobil pintar dan metodologi kajian. Bab V sehingga Bab XI menghuraikan kerja-kerja penyelidikan dan pembangunan yang telah dilaksanakan untuk meningkatkan kemampuan sedia ada agen mobil pintar. Ia bertujuan untuk menambah keupayaan agen mobil pintar menyelesaikan masalah aplikasi teragih. Kerja-kerja penyelidikan dan pembangunan yang dijalankan mengikut bab adalah seperti berikut:

- (a) Bab V – menghuraikan metodologi pembangunan agen mobil pintar.
- (b) Bab VI – menghuraikan teknik pemodelan agen mobil pintar.
- (c) Bab VII – menghuraikan rangka kerja pembangunan seni bina agen mobil pintar.
- (d) Bab VIII – menghuraikan seni bina agen mobil pintar.
- (e) Bab IX – menghuraikan struktur intra agen mobil pintar.
- (f) Bab X – menghuraikan struktur inter agen mobil pintar.

- (g) Bab XI – menghuraikan spesifikasi formal seni bina agen mobil pintar.

Bab XII dan Bab XIII pula menghuraikan produk akhir kajian dan kesimpulan yang diperolehi daripada kajian yang telah dijalankan.



**Rajah 1.1 : Garis kasar tesis**

## **BAB II**

### **MASALAH KAJIAN DAN CADANGAN PENYELESAIAN**

#### **2.1 Pengenalan**

Pengurusan masa-nyata rangkaian telekomunikasi (Jakobson dan Weissman, 1993), penjadualan mesyuarat automatik (Sen, 1997), sistem pengangkutan pintar (Palen, 1997) dan membeli-belah elektronik (Lohse dan Spiller, 1998) adalah contoh aplikasi teragih. Persamaan sepunya di antara pelbagai domain aplikasi teragih adalah pengagihan dan kompleksiti semulajadi (Agre, 1995). Pengagihan semulajadi, iaitu maklumat yang hendak diproses teragih dalam bentuk seperti berikut:

- (a) Spatial – sumber terletak pada lokasi geografi yang berbeza.
- (b) Temporal – sumber terbit pada masa yang berbeza.
- (c) Semantik – kelompokan sumber yang memerlukan capaian dengan ontologi dan bahasa yang berbeza.
- (d) Kefungsian – kelompokan sumber yang memerlukan capaian dengan tanggapan, kesan dan keupayaan pengetahuan yang berbeza.

Selain itu, kompleksiti semulajadi, iaitu saiz aplikasi yang berskala besar dan memerlukan teknologi perisian atau perkakasan yang canggih.

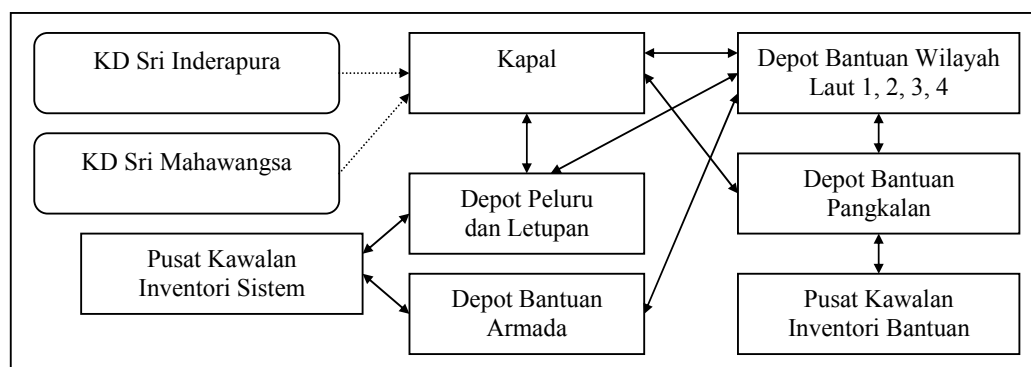
Di dalam bab ini, diuraikan masalah-masalah di dalam aplikasi teragih khususnya pengurusan inventori teragih di TLDM. Selain dijelaskan kepentingan agen mobil pintar dalam menangani masalah aplikasi teragih dan dicadangkan penyelesaian masalah pengurusan inventori teragih menggunakan agen mobil pintar.

## 2.2 Huraian Masalah

### 2.2.1 Masalah Lokasi Inventori

Pengurusan inventori memainkan fungsi penting di TLDM untuk menentukan senggaraan dan pembaikan kapal dapat memenuhi lebih 67% kesiapsiagaan. Ia memerlukan 1/3 kapal dalam sesuatu skuadron disenggara dan 2/3 kapal siap sedia untuk melaksanakan perintah operasi. Skwadron mempunyai 2, 4, 6, 8, ...,  $n$  bilangan kapal bergantung kepada peranan dan fungsi kapal. Kapal TLDM dikelaskan kepada frigat, korvet, pengangkut, penangkis ranjau dan lain-lain. Persediaan alat ganti adalah kritikal untuk memastikan kitar dan penjadualan senggaraan kapal yang licin di limbungan. Kapal yang siap sedia untuk bertugas mestilah mempunyai keperluan bekalan yang secukupnya untuk beroperasi pada bila-bila masa.

TLDM mempunyai lebih daripada 300,000 material inventori yang disimpan dalam dua kategori stor iaitu stor pangkalan (Depot Peluru dan Letupan, Depot Bantuan Armada dan lain-lain) dan stor kapal (KD Sri Inderapura, KD Sri Mahawangsa dan lain-lain) seperti pada Rajah 2.1. Material inventori TLDM dikategorikan kepada alat ganti kapal, bekalan makanan segar dan kering, bekalan am (pakaian, perabot dan alat hidangan), petroleum dan pelincir, peluru dan bahan letupan, minyak dan lain-lain. Faktor-faktor yang diambil kira dalam mengukur prestasi pengurusan inventori TLDM adalah peratus aras layanan yang tinggi, masa lapor pentadbiran yang rendah, nisbah pusingan ganti yang tinggi, peratus penolakan yang rendah dan jangka masa penghantaran yang singkat.



**Rajah 2.1 : Struktur stor Tentera Laut DiRaja Malaysia**

Masalah semasa pengurusan inventori TLDM adalah kelewatan membekal inventori kepada pemohon; kesukaran melakukan rujukan silang dengan peralatan di kapal; penyimpanan, capaian dan aliran data inventori yang tidak rapi; telahan dan kawalan inventori yang tidak komprehensif; kesukaran menyediakan perancangan pengambilan dan penghantaran inventori; kesukaran mengenal pasti inventori yang tidak lagi digunakan atau usang; dan kesukaran mengenal pasti kedudukan dan pergerakan inventori di antara stor. Kelemahan-kelemahan ini mengakibatkan kos penyimpanan dan pembelian meningkat, penggunaan inventori yang tidak optimum dan wujudnya inventori terperap.

## 2.2.2 Masalah Penjejakan Inventori

Masalah penjejakan inventori adalah masalah perancangan untuk menentukan jujukan stor yang perlu dilawati bagi meminimumkan jumlah masa jangkaan sehingga inventori yang dikehendaki dijumpai. Masalah ini diinspirasi oleh masalah penjelajahan jurujual (*traveling salesman problem*, TSP) yang dimodelkan oleh Garey et al. (1976). Masalah penjejakan inventori ditakrifkan seperti berikut:

“Terdapat  $n + 1$  stor,  $s_i$  di mana  $0 \leq i \leq n$ . Kebarangkalian proses penjejakan berjaya dilaksanakan pada sesuatu stor diberikan oleh  $0 \leq p_i \leq 1$ . Kebarangkalian ini saling tidak bergantung di antara satu sama lain. Masa  $t_i > 1$  diperlukan untuk pelaksanaan tugas di stor. Masa penjelajahan atau kependaman yang diambil semasa proses penjejakan di antara stor diberikan oleh  $l_{ij} \geq 0$  untuk bergerak dari  $s_i$  ke  $s_j$ . Apabila proses penjejakan telah selesai, ia akan kembali ke stor permulaan ( $s_0$ ). Bagi stor permulaan,  $p_0 = 0$  dan  $t_0 = 0$ .”

Masa jangkaan yang diperlukan untuk menjelajah setiap stor,  $T = \langle i_1, i_2, \dots, i_n \rangle$  adalah:

$$C_T = l_{0i_1} + t_{i_1} + p_{i_1}l_{i_1 0} + \sum_{k=2}^n \left\{ \left( \prod_{j=1}^{k-1} (1 - p_{i_j}) \right) \right\} (l_{i_{k-1}i_k} + t_{i_k} + p_{i_k}l_{i_k 0}) + \prod_{j=1}^n (1 - p_{i_j}) l_{n 0}$$

Masa  $l_{0i_1}$  diperlukan untuk menjelajah dari  $s_0$  ke stor pertama,  $s_{i_1}$ . Masa  $t_{i_1}$  pula diperlukan untuk melaksanakan tugas di stor pertama. Kebarangkalian  $p_{i_1}$  tugas

berjaya dilaksanakan dan kembali ke stor permulaan dengan masa  $l_{i0}$ . Bahagian kedua rangkaian algebra adalah masa untuk menjelajah dari stor  $i_1$  sehingga ke stor  $i_n$  termasuk masa pelaksanaan tugas di setiap stor yang boleh diumumkan sebagai:

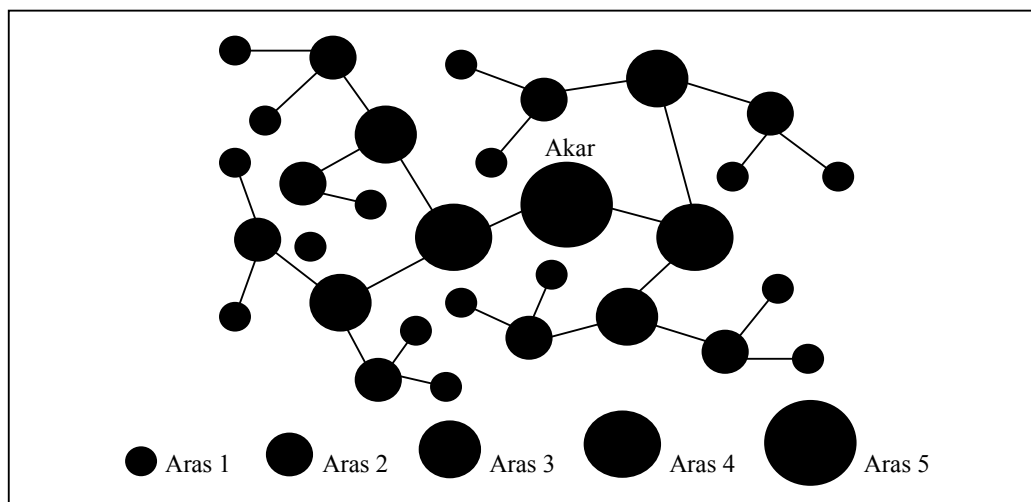
$$(\text{kebarangkalian kegagalan di stor } k - 1) \times (\text{masa kejayaan di stor } i_k)$$

Bahagian terakhir rangkaian algebra pula adalah kegagalan menjelajah semua stor dan kembali ke stor permulaan.

Masalah penjejakan inventori boleh dikategorikan sebagai masalah *NP-complete* (Moizumi, 1998). *NP-complete* adalah satu set masalah yang tidak mempunyai algoritma polinomial masa dan merupakan masalah sukar yang hanya mempunyai penyelesaian optimum.

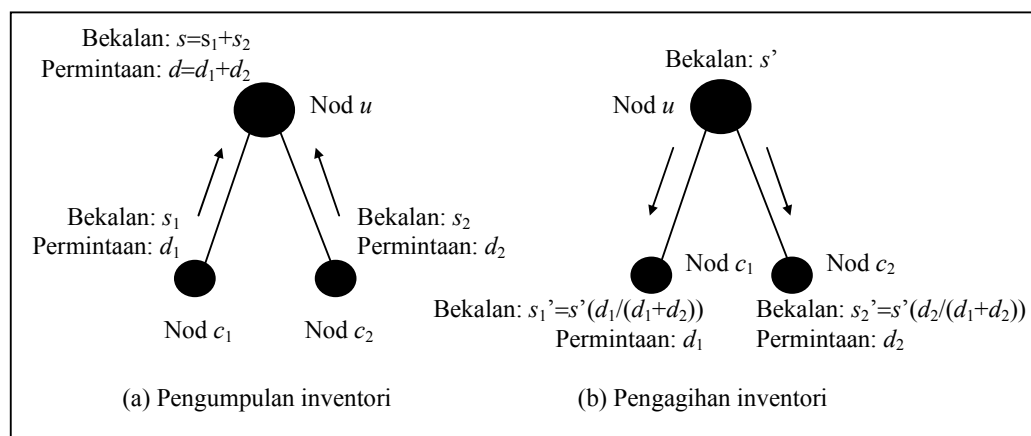
### 2.2.3 Masalah Peruntukan Inventori

Masalah peruntukan inventori dimodelkan sebagai pepohon perduaan dengan nod  $i = 1, 2, \dots, n$ , bekalan  $s_i$  dan permintaan  $d_i$  pada setiap nod,  $i$ . Matlamatnya adalah untuk mengumpulkan bekalan  $s'_i$  kepada nod-nod untuk memenuhi permintaan,  $d_i$  di setiap nod. Rajah 2.2 menunjukkan pepohon perduaan dengan aras,  $n = 5$  bagi menggambarkan masalah peruntukan inventori.



**Rajah 2.2 : Pepohon perduaan masalah peruntukan inventori**

Setiap nod memproses bekalan dan permintaan inventori. Bekalan dan permintaan dirambat ke nod aras atas melalui pengumpulan seperti pada Rajah 2.3 (a). Di nod akar, bekalan diagihkan ke nod aras bawah melalui pemecahan berkadar berdasarkan kepada permintaan seperti pada Rajah 2.3 (b). Andaikan nod  $u$  mempunyai nod anak  $c_1$  dan  $c_2$  dengan bekalan  $s_1$  dan  $s_2$  serta permintaan  $d_1$  dan  $d_2$ . Bekalan berkadar di nod  $u$  adalah  $s_1 + s_2$  dan permintaan berkadar adalah  $d_1 + d_2$ . Bekalan dan permintaan dirambat ke nod aras atas kecuali jika nod  $u$  adalah nod akar. Inventori diagihkan berdasarkan peraturan pemecahan berkadar iaitu jika nod  $u$  diperuntukan dengan bekalan  $s'$  yang mungkin berbeza dengan  $d_1 + d_2$ , maka  $c_1$  menerima  $\frac{s'd_1}{d_1 + d_2}$  dan  $c_2$  menerima  $\frac{s'd_2}{d_1 + d_2}$ .



**Rajah 2.3 : Pengumpulan dan pengagihan inventori**

### 2.3 Aplikasi Teragih Berasaskan Agen Mobil Pintar

Aplikasi teragih merupakan satu set komponen serempak yang saling berhubungan dan bekerjasama. Ciri-ciri utama aplikasi teragih adalah seperti berikut:

- Kepelbagaian – mempunyai berbagai-bagai sistem pengoperasian, perkakasan dan protokol rangkaian.
- Perpindahan proses – proses boleh dipindahkan ke mesin lain tanpa perlu memulakan semula atur cara.
- Boleh diskalakan – kos pengkomputeran yang minimum melalui perkongsian maklumat, ingatan dan storan.



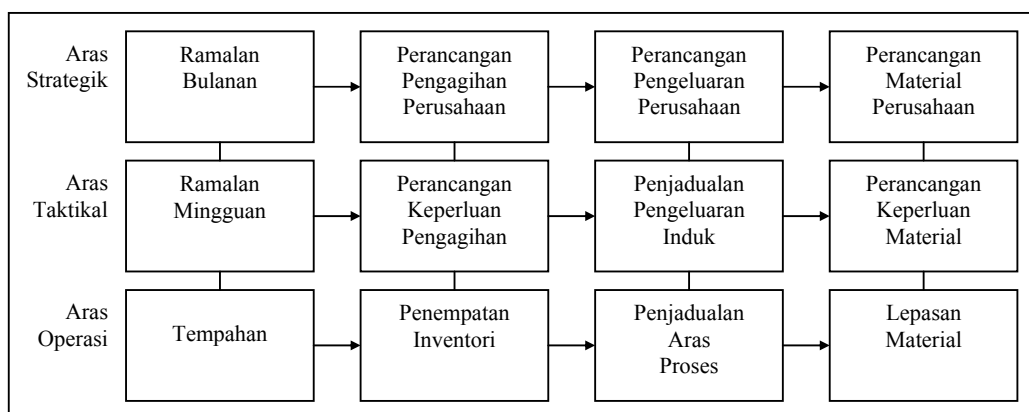
Agen mobil pintar diimplementasi ke dalam aplikasi teragih bukan sekadar untuk menyelesaikan masalah maklumat teragih dan kompleksiti domain. Ia disebabkan oleh kemampuannya mengendalikan aplikasi dengan entiti-entiti penyelesaian masalah yang teragih secara fizikal atau lojikal dari aspek data, kawalan, kepakaran atau sumber (Jennings dan Wooldridge, 1995). Contohnya, sistem penjagaan kesihatan teragih memerlukan penyelarasan di antara doktor dan jururawat di hospital dengan pembantu kesihatan dan pembantu farmasi di pusat kesihatan untuk merawat pesakit. Senario sebegini mewujudkan situasi seperti berikut:

- (a) Data teragih – pembantu kesihatan mempunyai rekod yang berlainan dengan jururawat walaupun merujuk kepada pesakit yang sama.
- (b) Kawalan teragih – setiap pelaku mempunyai peranan dengan set tugas yang berlainan.
- (c) Kepakaran teragih – doktor mempunyai pengetahuan yang berlainan dengan pembantu kesihatan.
- (d) Sumber teragih – pembantu farmasi bertanggungjawab menyediakan ubat-ubatan. Jururawat pula bertanggungjawab menyediakan kelengkapan bilik rawatan sekiranya keadaan pesakit adalah kronik.

Contoh-contoh agen mobil pintar yang dibangunkan dan diimplementasi ke dalam aplikasi teragih adalah seperti DAIS untuk pengurusan maklumat mudah suai teragih (Kotz et al., 2002), HERMES untuk pengurusan perpustakaan digital teragih (Faulstich dan Spiliopoulou, 2000) dan CMIPAgent untuk pengurusan rangkaian teragih (Covaci et al., 1997).

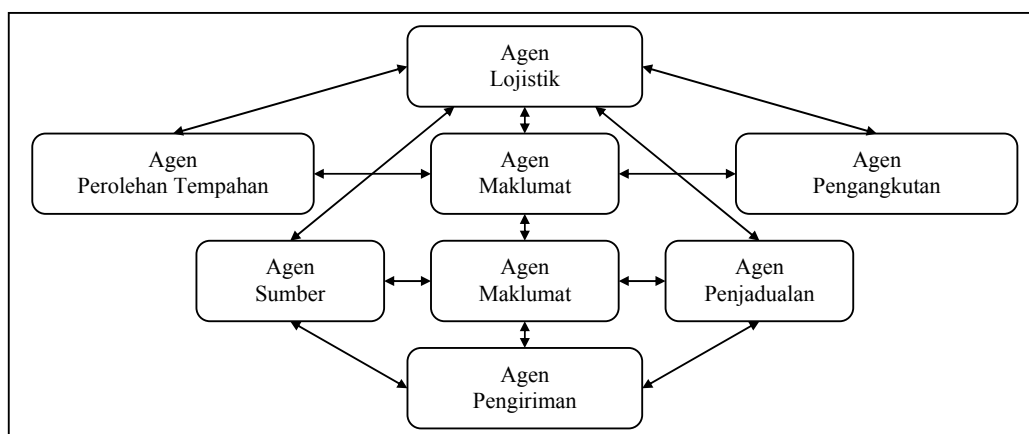
### **2.3.1 Sistem Pengurusan Rantaian Bekalan Bersepadu**

Fox et al. (2000) telah membangunkan sistem pengurusan rantaian bekalan bersepadu (*integrated supply chain management*, ISCM) bermula dari tempahan dan penerimaan material mentah kepada pengilangan produk sehingga pembekalan kepada pengguna. Terdapat tiga aras fungsi operasi ISCM seperti pada Rajah 2.4 iaitu aras strategik, aras taktikal dan aras operasi.



**Rajah 2.4 : Fungsi pengurusan rantai bekalan**

Ciri-ciri ISCM adalah teragih, bersepadu, mudah konfigurasi, interaktif dan mudah suai. Agen mobil pintar digunakan untuk menyokong persekitaran ISCM yang dinamik. Sebagaimana Rajah 2.5, setiap agen mobil pintar mewakili entiti yang spesifik seperti agen pengangkutan berperanan menentukan dan menjadual sumber pengangkutan. Agen sumber pula berperanan mengurus inventori dan belian.

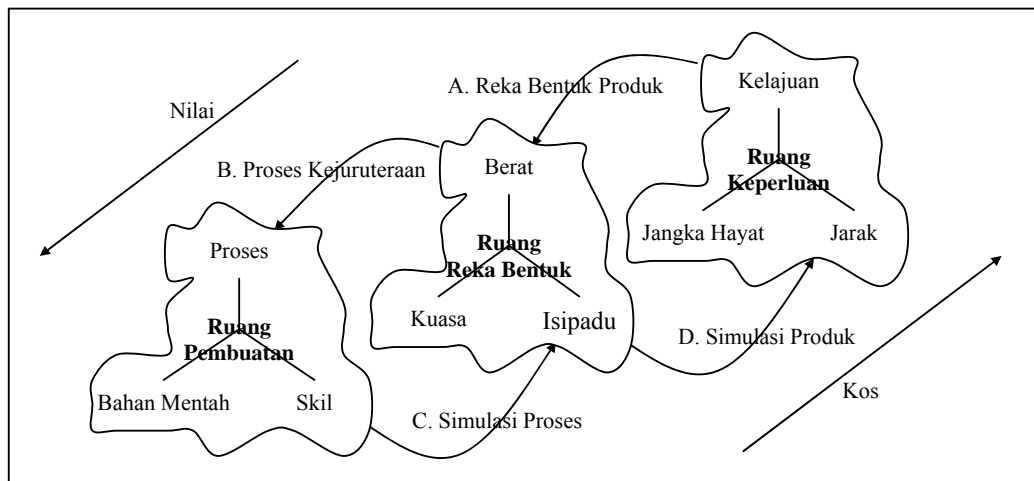


**Rajah 2.5 : Agen ISCM**

### 2.3.2 Sistem Reka Bentuk Produk-Proses Bersepadu

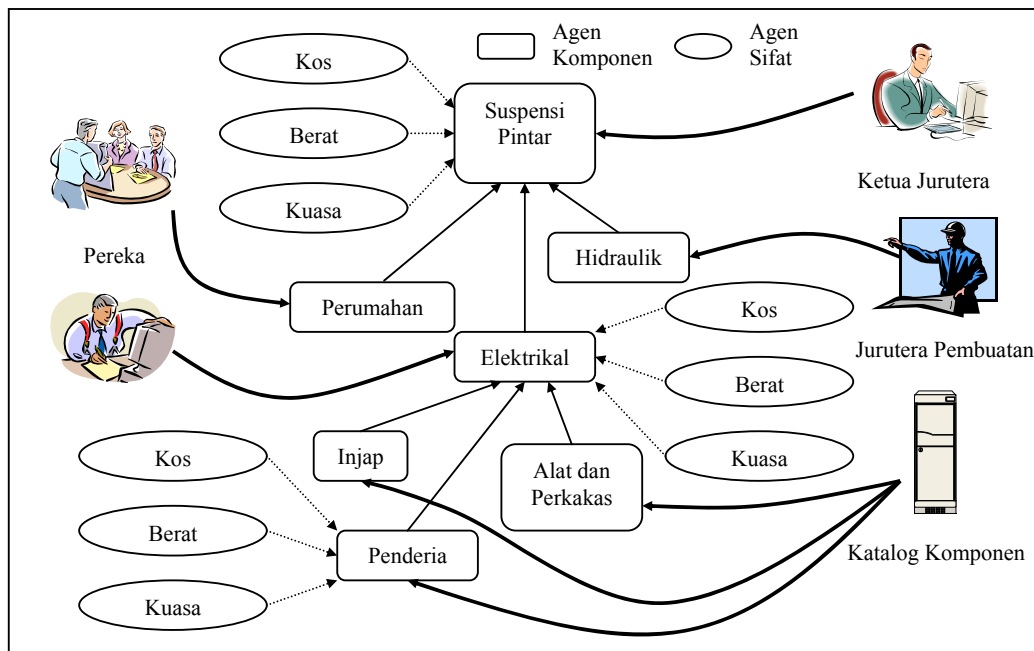
Sistem reka bentuk produk-proses bersepadu menemukan pereka dan jurutera dari lokasi dan syarikat yang berbeza untuk mereka bentuk komponen dan subsistem produk yang kompleks. Parunak et al. (1999) telah membangunkan RAPPID untuk

menyelesaikan masalah pemetaan ruang keperluan, ruang reka bentuk dan ruang pembuatan di dalam sistem reka bentuk produk-proses seperti pada Rajah 2.6.



**Rajah 2.6 : Pemetaan ruang dalam RAPPID**

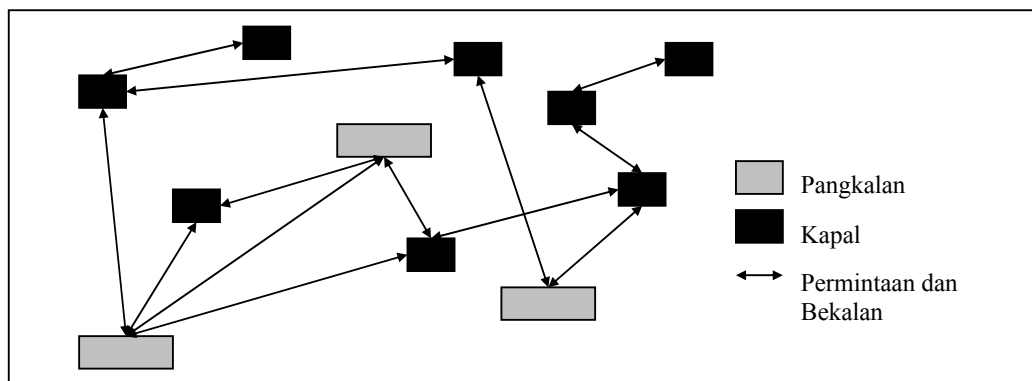
Di dalam RAPPID, agen mobil pintar dikategorikan kepada agen komponen dan agen sifat seperti pada Rajah 2.7. Agen komponen mewakili bahagian-bahagian produk. Agen sifat pula mewakili kekangan produk.



**Rajah 2.7 : Agen RAPPID**

## 2.4 Sistem Pengurusan Inventori Teragih

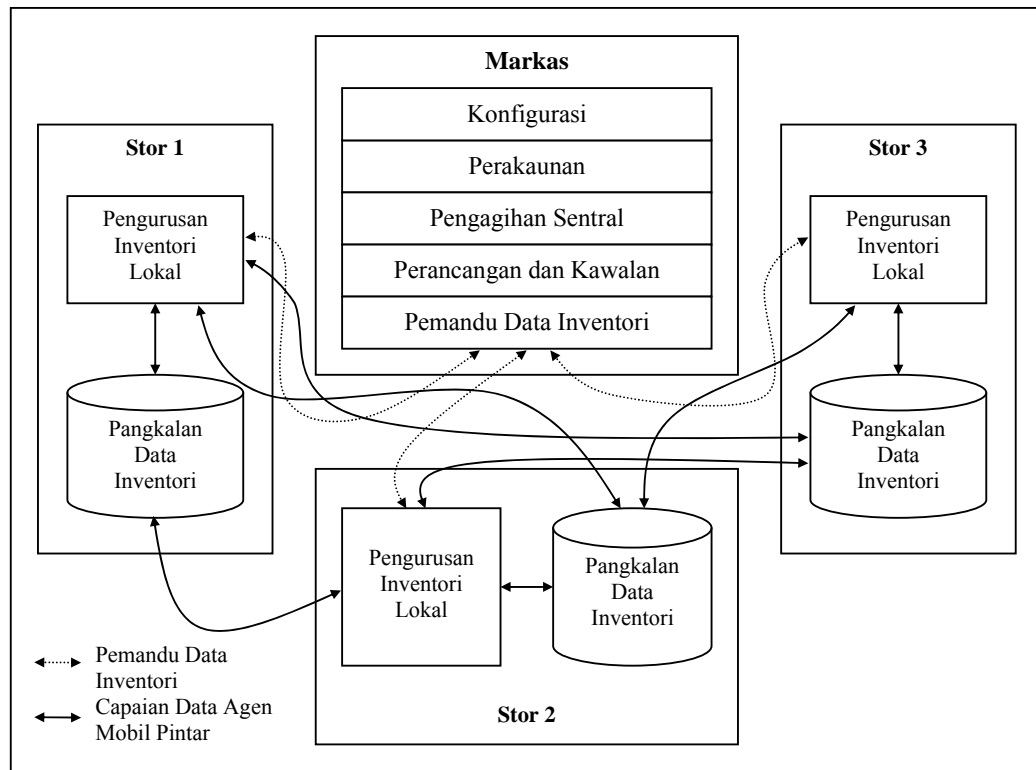
Berdasarkan kepada dua aplikasi teragih yang dibincangkan pada Seksyen 2.3, masalah pengurusan inventori teragih di TLDM sebagaimana yang diuraikan dalam Seksyen 2.2 boleh diselesaikan menggunakan agen mobil pintar. Rajah 2.8 menggambarkan contoh situasi permintaan dan bekalan inventori yang berlaku di TLDM. Terdapat beberapa buah stor yang saling terangkai melalui internet yang dikelaskan kepada stor pangkalan dan stor kapal. Bagaimanapun, tidak semua kapal mempunyai stor. Setiap kapal atau pangkalan boleh membuat permintaan ke atas sesuatu stor tanpa mengira kategori. Bekalan dibuat berdasarkan kekangan ketersediaan inventori dan keutamaan permintaan.



**Rajah 2.8 : Sampel rangkaian permintaan dan bekalan inventori**

Seni bina sistem pengurusan inventori teragih yang akan digunakan oleh TLDM adalah seperti pada Rajah 2.9. Setiap stor mempunyai pangkalan data yang menyimpan maklumat inventori. Markas berperanan mengendalikan satu set rangkaian stor. Sistem ini dikomposisi oleh lima modul seperti berikut:

- (a) Modul konfigurasi – mengurus kod palang material inventori.
- (b) Modul perakaunan – mengurus aktiviti perakaunan dan belian.
- (c) Modul pengagihan sentral – mengurus agihan inventori ke stor-stor.
- (d) Modul perancangan dan kawalan – merancang dan mengawal aliran inventori termasuk meramalkan keperluan inventori.
- (e) Modul pengurusan inventori lokal – mengurus aktiviti inventori setempat seperti penerimaan, permohonan dan pengeluaran, pelupusan, pemindahan dan lain-lain.

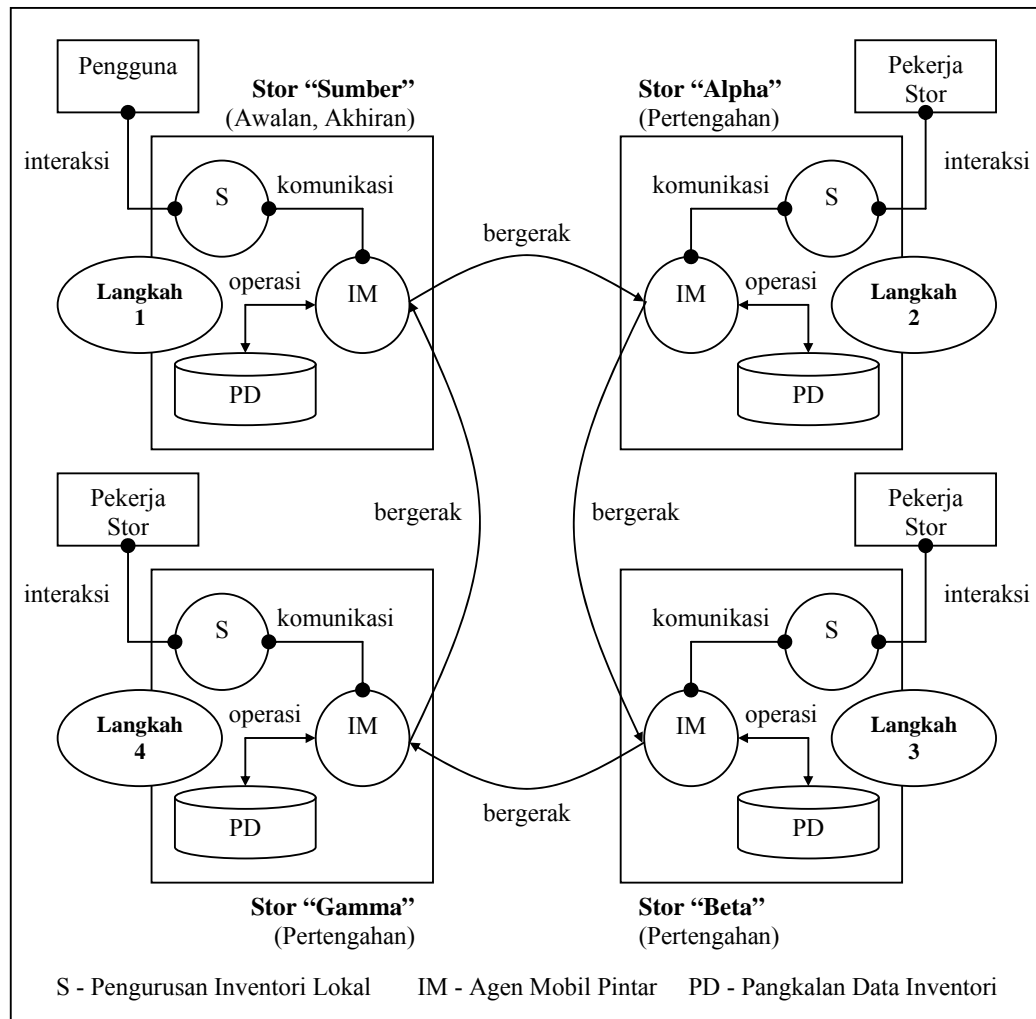


**Rajah 2.9 : Seni bina sistem pengurusan inventori teragih**

Pemandu data inventori berfungsi mengumpul data inventori daripada setiap stor, memprosesnya dan menghantarnya kepada modul perancangan dan kawalan. Kesemua modul-modul berkenaan menggunakan agen mobil pintar yang boleh mencapai pangkalan data inventori setiap stor seperti yang akan diuraikan dalam Seksyen 2.4.1.

#### **2.4.1 Cadangan Penyelesaian Menggunakan Agen Mobil Pintar**

Perlaksanaan sistem pengurusan inventori teragih berasaskan agen mobil pintar secara kasarnya adalah seperti pada Rajah 2.10. Pengguna berinteraksi dengan agen mobil pintar bagi menginput butiran keperluan, kemudian agen mobil pintar melaksanakan proses penjejakan dan permohonan stok. Akhirnya menjana laporan dan mengemukakan penyelesaian berdasarkan maklumat yang dikumpul oleh agen berkenaan.



**Rajah 2.10 : Agen mobil pintar dalam sistem pengurusan inventori teragih**

## 2.5 Ringkasan

Agensi mobil pintar telah diimplementasikan dalam pelbagai domain aplikasi teragih. Ciri-ciri agensi mobil pintar yang berupaya bergerak di sepanjang rangkaian, melaksanakan tugasnya secara berautonomi dan keupayaan berinteraksi dengan agensi atau entiti lain merupakan pemetaan yang ideal dengan aplikasi teragih. Bab ini telah mengupas mengenai aplikasi teragih dan mencadangkannya untuk digunakan dalam pengurusan inventori teragih di TLDM. Gambaran umum peranan agensi mobil pintar dalam sistem pengurusan inventori teragih turut diterangkan dalam bab ini.

## **BAB III**

### **AGEN MOBIL PINTAR: TEORI DAN APLIKASI**

#### **3.1 Pengenalan**

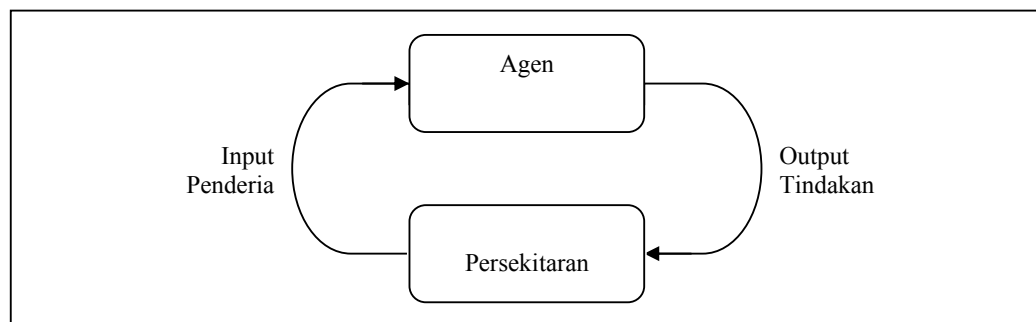
Teknologi agen mobil pintar merupakan salah satu paradigma baru dalam kejuruteraan perisian khususnya sistem pelbagai agen. Bagaimanapun, ianya telah dikaji lebih awal oleh penyelidik dalam bidang kepintaran buatan. Perkembangan teknologi jaringan komputer dan keperluan membangunkan perisian yang dapat menjalankan tugas yang didelegasikan kepadanya secara berautonomi, pintar dan berkeupayaan mengumpul maklumat yang teragih adalah diperlukan. Oleh kerana itu, peranan agen mobil pintar untuk menyelesaikan masalah tersebut tidak dapat dielakkan. Secara tidak langsung, agen mobil pintar perlu memiliki kredibiliti yang optimum bersesuaian dengan domain aplikasi yang semakin besar dan kompleks terutamanya dalam perniagaan elektronik, telekomunikasi, pengeluaran, pemantauan pesakit dan permainan komputer.

Objektif bab ini adalah untuk meninjau kemajuan yang telah dicapai dalam penyelidikan dan implementasi agen mobil pintar bermula daripada teori sehingga kepada praktik. Ia merupakan kesinambungan tinjauan yang telah dilakukan oleh Sierra et al. (2000), Nwana (1996), Jennings dan Wooldridge (1996) serta Genesereth dan Ketchpel (1994) yang hanya memberi tumpuan kepada agen secara keseluruhan dan menyentuh tentang agen mobil pintar secara umum. Kajian literatur secara mendalam dan panjang lebar juga dibincangkan pada permulaan Bab V sehingga Bab XII.

## 3.2 Agen

### 3.2.1 Definisi Agen

Agen telah diperkenalkan oleh Hewitt (1977) melalui model pelaku serempak. Dalam modelnya, beliau mengemukakan teori tentang objek yang dipanggil pelaku yang mempunyai ciri-ciri berautonomi, interaktif dan reaktif. Agen ditakrifkan sebagai entiti yang mengamati persekitarannya melalui penderia dan bertindak ke atas persekitaran melalui pemberi kesan (Russell, 1997). Wooldridge (1998) pula mentakrifkan agen sebagai sistem komputer yang ditempatkan pada sesuatu persekitaran dan berupaya bertindak secara berautonomi bagi mencapai matlamat yang telah direka. Hubungan di antara agen dengan persekitarannya ditunjukkan pada Rajah 3.1. Agen mendapatkan input mengenai persekitaran melalui penderia dan memprosesnya bagi menghasilkan output yang berupa satu tindakan.



**Rajah 3.1 : Interaksi agen dengan persekitarannya**

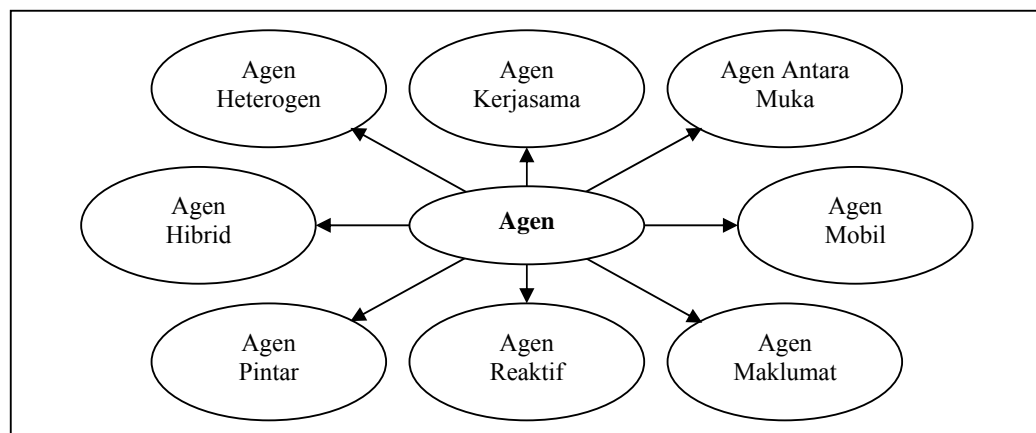
### 3.2.2 Pengelasan Agen

Pengelasan agen menurut Nwana (1996) dibahagikan kepada lapan kategori seperti pada Rajah 3.2, iaitu:

- (a) Agen kerjasama – agen yang memiliki kemampuan melakukan kerjasama dan penyelarasan dengan agen lain.
- (b) Agen antara muka – agen yang memiliki kemampuan bekerjasama dan saling berinteraksi dengan pengguna.



- (c) Agen mobil – agen yang memiliki kemampuan bergerak daripada satu mesin ke mesin yang lain melalui rangkaian.
- (d) Agen maklumat – agen yang memiliki kemampuan mengumpul, memperguna dan mengurus maklumat yang wujud di dalam internet.
- (e) Agen reaktif – agen yang memiliki kemampuan bertindak secara rangsangan dan tindak balas ke atas keadaan semasa persekitarannya.
- (f) Agen pintar – agen yang memiliki keupayaan melaksanakan tugasnya secara berautonomi, berinteraksi dengan persekitaran dan menaakul untuk menentukan tindakan.
- (g) Agen hibrid – agen yang merupakan gabungan dua atau lebih kelas agen sama ada agen pintar, agen reaktif, agen maklumat, agen mobil, agen antara muka atau agen kerjasama sebagai satu agen tunggal.
- (h) Agen heterogen – agen yang merupakan gabungan dua atau lebih agen hibrid sebagai satu agen tunggal.

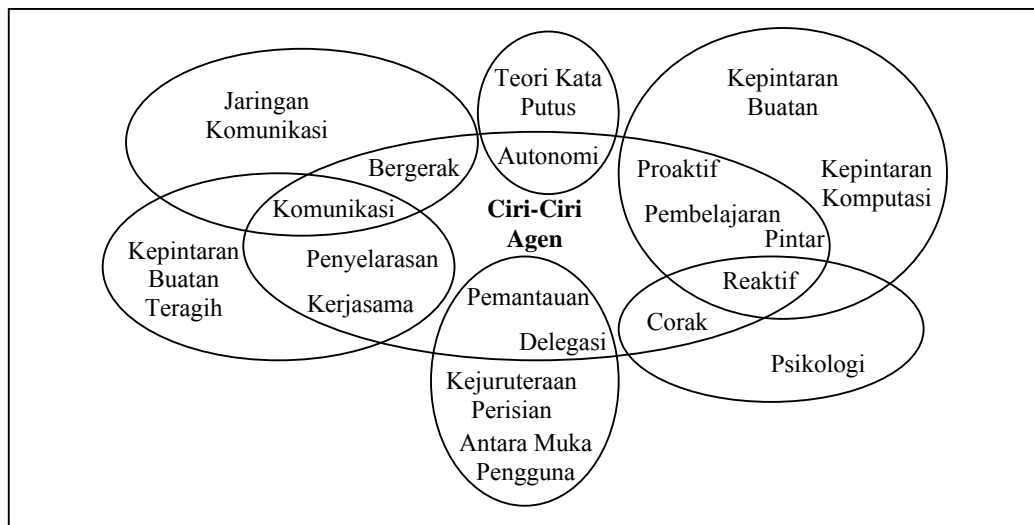


**Rajah 3.2 : Pengkelasan agen**

### 3.2.3 Bidang Ilmu Agen

Masalah pembelajaran, kepintaran, proaktif dan reaktif adalah intisari bidang kepintaran buatan. Bidang kepintaran buatan teragih pula berkisar kepada masalah penyelarasan, kerjasama dan komunikasi di antara agen dalam sistem pelbagai agen. Bidang komunikasi dan rangkaian teragih pula membawa kepada kewujudan agen

yang mempunyai kemampuan bergerak dan berkomunikasi dengan agen lain. Perkembangan agen yang drastik tidak terlepas dari pengaruh bidang psikologi yang mengupas agen secara teori dan falsafah. Seterusnya bidang kejuruteraan perisian berperanan menyediakan metodologi pembangunan, kaedah analisa dan reka bentuk serta implementasi agen. Akhir sekali ialah bidang teori kata putus yang menekankan bagaimana agen harus menentukan strategi dan menjalankan tugas secara berautonomi. Kaitan bidang ilmu berkenaan dengan agen dapat digambarkan seperti pada Rajah 3.3 (Chaib-Draa et al., 1992).



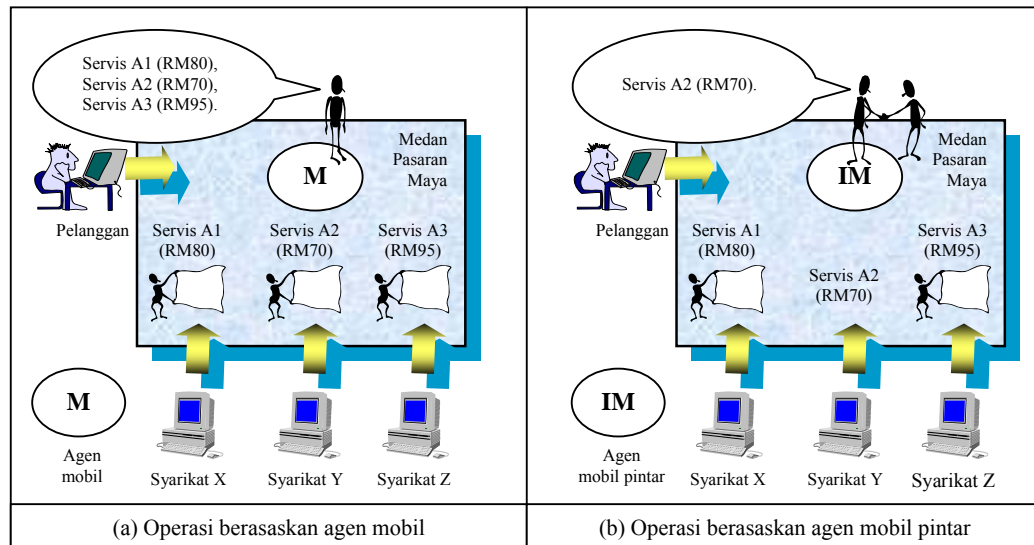
**Rajah 3.3 : Bidang ilmu yang berkaitan dengan agen**

### 3.3 Agen Mobil Pintar

#### 3.3.1 Definisi Agen Mobil Pintar

Agen mobil pintar ditakrifkan sebagai agen yang berupaya mengubah kedudukan posisinya dalam rangkaian yang heterogen dan beroperasi melalui pelaksanaan tiga fungsi utama iaitu mengamati situasi yang dinamik pada persekitarannya, bertindak kepada rangsangan situasi dan menaakul untuk menterjemah pengamatan, memilih inferens, menyelesaikan masalah dan menentukan tindakan. Perbezaan operasi di antara agen mobil pintar berbanding

dengan agen mobil dari aspek bantuan penyelesaian dalam perniagaan elektronik ditunjukkan pada Rajah 3.4. Agen mobil pintar dapat melaksanakan proses perundingan dan mendapatkan servis ideal untuk pelanggan berdasarkan matlamat yang spesifik (Rajah 3.4 (b)). Agen mobil pula membawa pulang semua hasil carian kepada pelanggan tanpa memberi bantuan penyelesaian (Rajah 3.4 (a)).



**Rajah 3.4 : Perbandingan agen mobil dengan agen mobil pintar dalam perniagaan elektronik**

### 3.3.2 Ciri-Ciri Agen Mobil Pintar

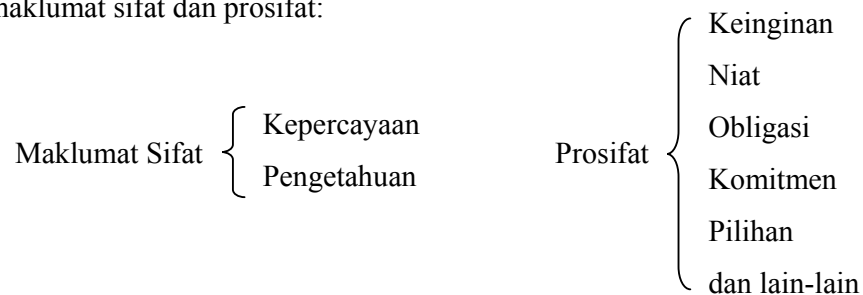
Ciri-ciri yang perlu ada pada agen mobil pintar berdasarkan peranan dan fungsinya adalah seperti berikut:

- (a) Autonomi – keupayaan agen mobil pintar melaksanakan tugas secara sendiri tanpa dipengaruhi oleh pengguna, agen lain atau persekitaran.
- (b) Keupayaan bersosial – keupayaan agen mobil pintar berinteraksi dan bekerjasama dengan agen atau manusia untuk mencapai matlamatnya.
- (c) Reaktif – keupayaan agen mobil pintar mengamati persekitarannya dan bertindak balas ke atas perubahan yang berlaku.
- (d) Proaktif – keupayaan agen mobil pintar mempamerkan kelakuan terarah matlamat dengan melakukan inisiatif tertentu.

- (e) Mudah suai – keupayaan agen mobil pintar mengubah kelakuan atau pengetahuannya untuk bertindak balas ke atas persekitaran dinamik.
- (f) Mobil – keupayaan agen mobil pintar bergerak di sepanjang rangkaian untuk meningkatkan kemampuannya menyelesaikan masalah.
- (g) Benar – andaian agen mobil pintar tidak berkemampuan untuk berkomunikasi secara palsu.
- (h) Rasional – andaian agen mobil pintar bertindak untuk mencapai matlamatnya dan tidak bertindak menggagalkan matlamat tersebut.

### 3.4 Teori Agen Mobil Pintar

Teori agen mobil pintar adalah spesifikasi atau huraian konseptual mengenai struktur agen mobil pintar yang ditakrif secara formal. Struktur agen mobil pintar dikategorikan kepada struktur intra agen dan struktur inter agen. Struktur intra agen adalah sifat-sifat yang menentukan tindakan agen mobil pintar sebagai satu perisian individu. Struktur inter agen pula adalah sifat-sifat yang menentukan tindakan agen mobil pintar sebagai sebahagian daripada komuniti. Struktur intra agen dibahagikan kepada maklumat sifat dan prosifat:



Maklumat sifat adalah informasi yang dimiliki oleh agen mobil pintar mengenai dunia penempatannya. Prosifat pula adalah perangsang kepada tindakan agen mobil pintar. Contoh penyelidikan teori agen mobil pintar yang dilakukan ke atas struktur intra agen adalah:

- (a) Pengetahuan dan tindakan (Moore, 1996) – kajian ke atas prasyarat pengetahuan terhadap tindakan agen mobil pintar. Ia mengfokus kepada pengetahuan yang perlu dimiliki oleh agen mobil pintar untuk membolehkannya melaksanakan beberapa tindakan tertentu.

- (b) Kepercayaan dan kesedaran (Lakemeyer, 1994) – pengetahuan agen mobil pintar dibahagikan kepada set kepercayaan eksplisit dan set kepercayaan implisit untuk menyelesaikan masalah omnisains.
- (c) Seni bina BDI (Rao dan Georgeff, 1998) – menentukan apakah matlamat yang akan dicapai dan bagaimana ia perlu dicapai melalui kepercayaan, keinginan dan niat.
- (d) Rangka kerja agensi dan autonomi (Luck dan d’Inverno, 2001) – agensi adalah hirarki tiga entiti yang mencakupi objek, agen (objek dengan matlamat) dan agen berautonomi (agen dengan motivasi).

Penyelarasan, kerjasama, pertandingan dan perbincangan adalah sebahagian daripada struktur inter agen. Komunikasi dan mobiliti adalah keupayaan yang diperlukan untuk memastikan struktur inter agen berfungsi secara efektif. Contoh teknologi mobiliti adalah:

- (a) Ambient calculus (Cardelli dan Gordon, 2000) – pergerakan sekitar (unit yang sedang dilarikan dan bersarang) dan subkomponennya di antara komputer melalui pemindahan persekitaran dan komputasi sekitar.
- (b) Klaim (De Nicola et al., 1998) – pergerakan proses (unit yang sedang dilarikan) di antara persekitaran pengkomputeran dengan melaksanakan pengambilan dan penghantaran kod.
- (c) Mobile UNITY (McCann dan Roman, 1998) – pergerakan dan penseketikaan atur cara di sepanjang rangkaian dengan mengumpukkan semula nilai pembolehkan lokasi.
- (d) Obliq (Cardelli, 1995) – pergerakan objek di antara tapak secara pengklonan dengan melaksanakan alias ke atas objek asal.

Mobiliti menerangkan bagaimana kod-kod atur cara agen mobil pintar berpindah di antara mesin yang berlainan.

Komunikasi pula menerangkan bagaimana maklumat saling bertukar ganti di antara agen mobil pintar dan persekitarannya. Contoh teknologi komunikasi adalah:

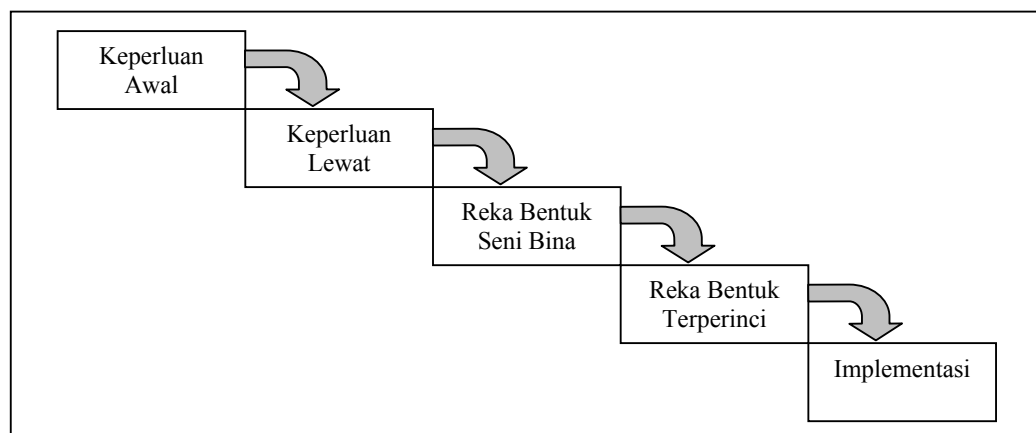
- (a) Mesej (Huhns dan Singh, 1997) – komunikasi di antara agen mobil pintar dikelaskan kepada dua iaitu penerapan atau pertanyaan.

- (b) Tindakan penuturan (Cohen dan Levesque, 1990) – komunikasi di antara agen mobil pintar dianggap sebagai tindakan yang merupakan permohonan, cadangan, maklum balas atau komitmen.
- (c) KQML (Finin et al., 1994) – protokol komunikasi yang membenarkan agen mobil pintar berkongsi maklumat dan pengetahuan. Komunikasi berlaku secara segerak atau tidak segerak seperti pelayan-pelanggan.

### 3.5 Kejuruteraan Perisian Berasaskan Agen Mobil Pintar

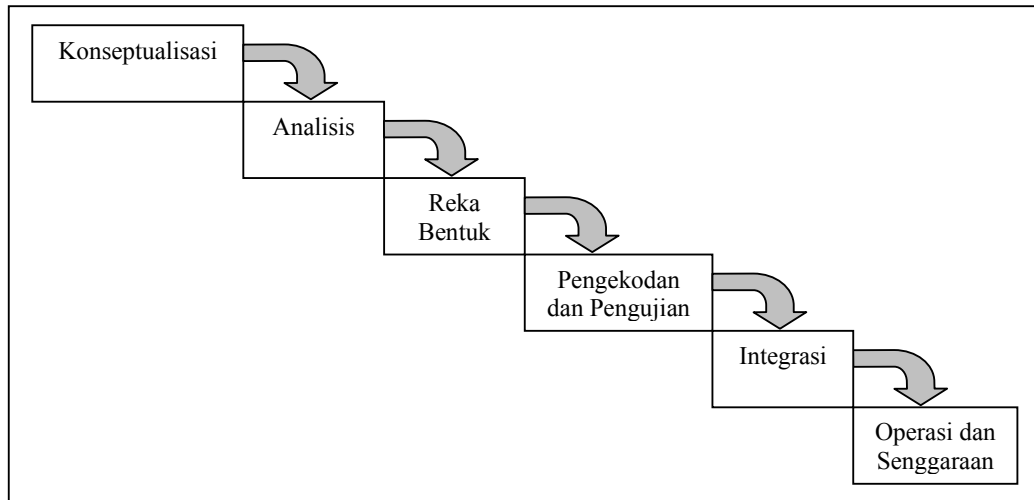
#### 3.5.1 Metodologi Berorientasikan Agen Mobil Pintar

Metodologi berorientasikan agen mobil pintar dibahagikan kepada dua kelompok iaitu yang menyambung metodologi berorientasikan objek atau metodologi berorientasikan kejuruteraan pengetahuan. Metodologi berorientasikan objek digunakan kerana agen mobil pintar dianggap sebagai objek aktif dan objek yang mempunyai keadaan mental (Burmeister, 1996). Tropos (Castro et al., 2002), MaSE (Deloach et al., 2001) dan Gaia (Wooldridge et al., 2000) adalah contoh metodologi berorientasikan objek. Rajah 3.5 menggambarkan fasa-fasa di dalam Tropos yang terdiri daripada keperluan awal, keperluan lewat, reka bentuk seni bina, reka bentuk terperinci dan implementasi.



**Rajah 3.5 : Fasa dalam Tropos**

Metodologi berorientasikan kejuruteraan pengetahuan pula digunakan kerana agen mobil pintar adalah perisian yang mempunyai kecerdikan dengan pengetahuan dan kepercayaan (Iglesias et al., 1998). MAS-CommonKADS (Iglesias et al., 1996) dan CoMoMAS (Glaser, 1996) adalah contoh metodologi berorientasikan kejuruteraan pengetahuan. Rajah 3.6 menggambarkan fasa-fasa di dalam MAS-CommonKADS yang terdiri daripada konseptualisasi, analisis, reka bentuk, pengekodan dan pengujian, integrasi serta operasi dan senggaraan.

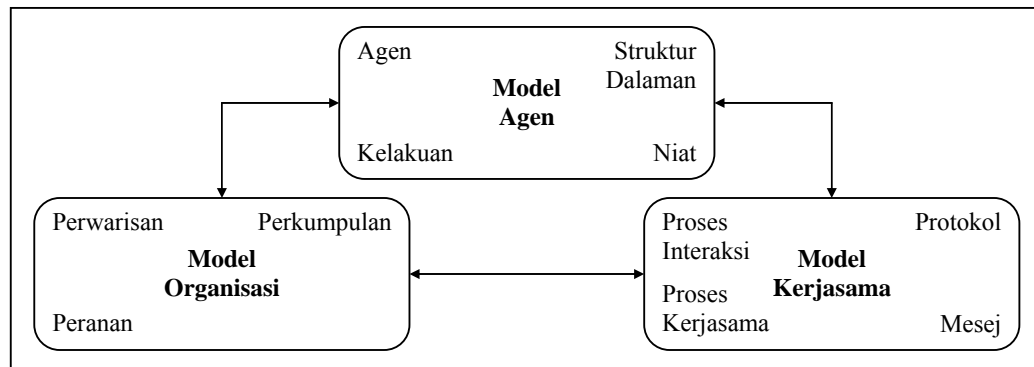


**Rajah 3.6 : Fasa dalam MAS-CommonKADS**

### 3.5.2 Teknik Pemodelan Agen Mobil Pintar

AUML (Bauer et al., 2001), AWIC (Muller, 1997), AOAD (Burmeister, 1996) dan teknik pemodelan agen BDI (Kinny et al., 1996) adalah sebahagian teknik pemodelan yang boleh digunakan untuk memodelkan agen mobil pintar. Terdapat tiga jenis model yang digunakan dalam AOAD seperti pada Rajah 3.7, iaitu:

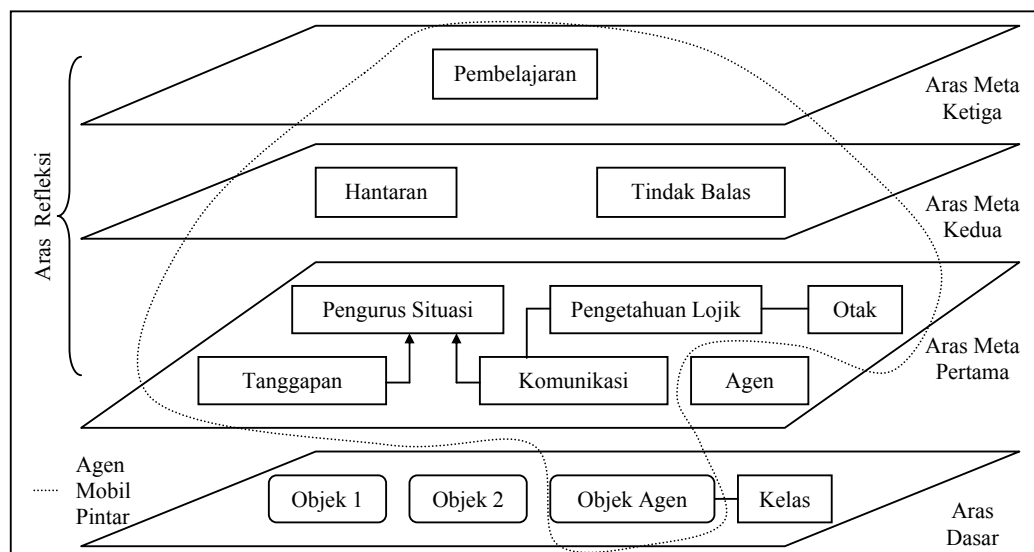
- (a) Model agen – menerangkan struktur dalaman agen mobil pintar seperti kepercayaan, pelan, matlamat dan kelakuan.
- (b) Model organisasi – menerangkan hubungan agen mobil pintar dengan agen atau objek lain sama ada perwarisan, peranan atau perkumpulan.
- (c) Model kerjasama – menerangkan interaksi agen mobil pintar dengan agen lain termasuk protokol, mesej, proses kerjasama dan interaksi.



**Rajah 3.7 : Model dalam AOAD**

### 3.5.3 Rangka Kerja Agen Mobil Pintar

Faktor boleh guna semula merupakan matlamat utama pembangunan sesuatu rangka kerja. DECAF (Graham dan Decker, 1999) dan DESIRE (Brazier et al., 1997) serta usaha yang dilakukan oleh Lejter dan Dean (1996), Bellavista et al. (1999) dan Yim et al. (2000) adalah sebahagian rangka kerja yang boleh diguna semula untuk membangunkan seni bina agen mobil pintar. Rajah 3.8 menunjukkan rangka kerja Brainstorm/J yang dipelopori oleh Zunino dan Amandi (2000), di mana agen mobil pintar dibentuk oleh empat aras objek meta.



**Rajah 3.8 : Rangka kerja Brainstorm/J**

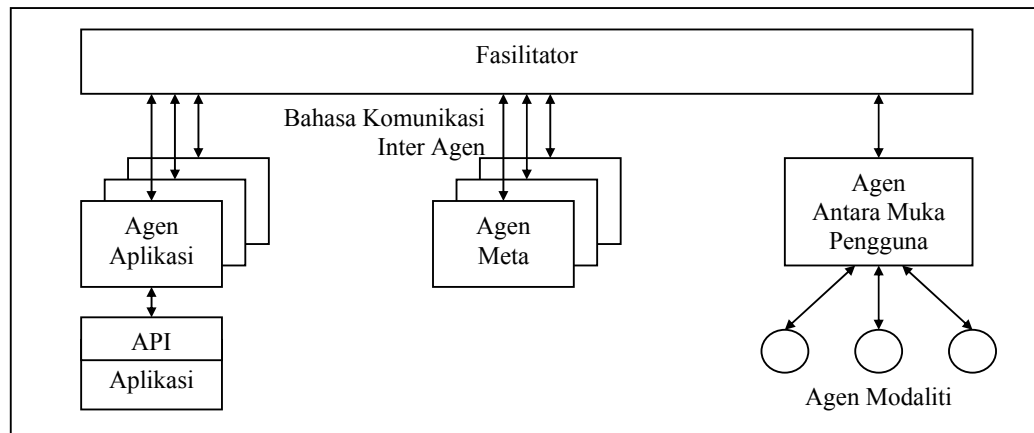


### 3.5.4 Seni Bina Agen Mobil Pintar

Seni bina agen mobil pintar boleh diguguskan kepada empat kategori berdasarkan ciri-ciri dan keadaan dalaman agen mobil pintar, iaitu:

- (a) Reaktif – proses membuat kata putus berlandaskan pemetaan secara langsung daripada situasi kepada tindakan.
- (b) BDI – proses membuat kata putus berlandaskan tafsiran struktur data kepercayaan, keinginan dan niat.
- (c) Berlapis – proses membuat kata putus berlandaskan pelbagai lapisan perisian. Setiap lapisan bertaakul secara jelas mengenai persekitaran daripada paparan peringkasan yang berlainan.
- (d) Berasaskan logik – proses membuat kata putus berlandaskan tolakan logikal.

Seni bina OAA (Martin et al., 1999) seperti pada Rajah 3.9 adalah contoh seni bina agen mobil pintar reaktif. Agen Aplikasi, Agen Meta dan Agen Antara Muka Pengguna merupakan sumber pengetahuan. Fasilitator pula bertindak sebagai kotak hitam.

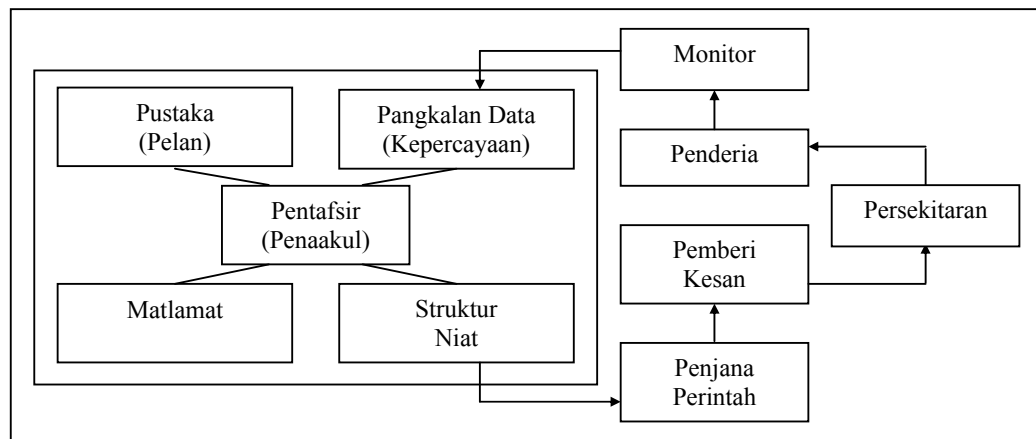


**Rajah 3.9 : Seni bina OAA**

Seni bina agen mobil pintar BDI seperti pada Rajah 3.10 dibangunkan oleh Busetta dan Ramamohanarao (1998) untuk agen BDIM. Pentafsir berfungsi sebagai otak yang akan melaksanakan pelan berdasarkan niat, pelan, matlamat dan

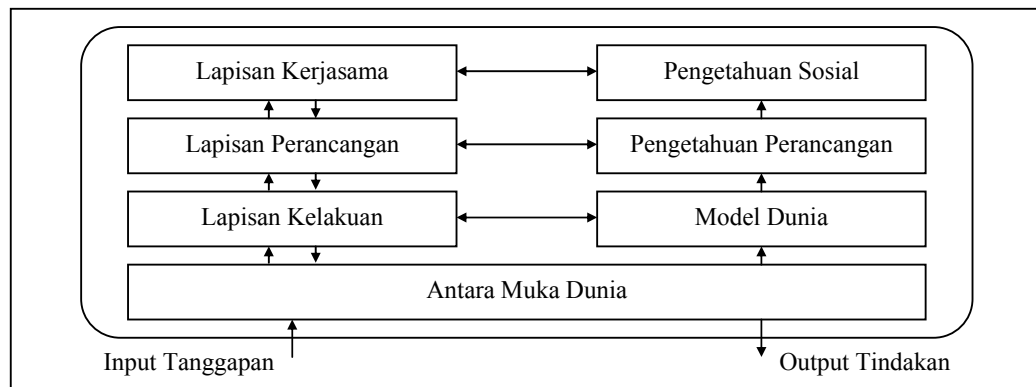
kepercayaan tentang situasi semasa persekitaran yang diinput oleh penerima.

Tindakan yang merupakan output agen akan dilaksanakan oleh pemberi kesan.



**Rajah 3.10 : Seni bina BDIM**

Seni bina INTERRAP (Muller, 1996) mengandungi tiga lapisan kawalan iaitu lapisan kerjasama, lapisan perancangan dan lapisan kelakuan dengan pangkalan pengetahuan tersendiri. Seni bina INTERRAP seperti yang ditunjukkan pada Rajah 3.11 adalah contoh seni bina agen mobil pintar berlapis.

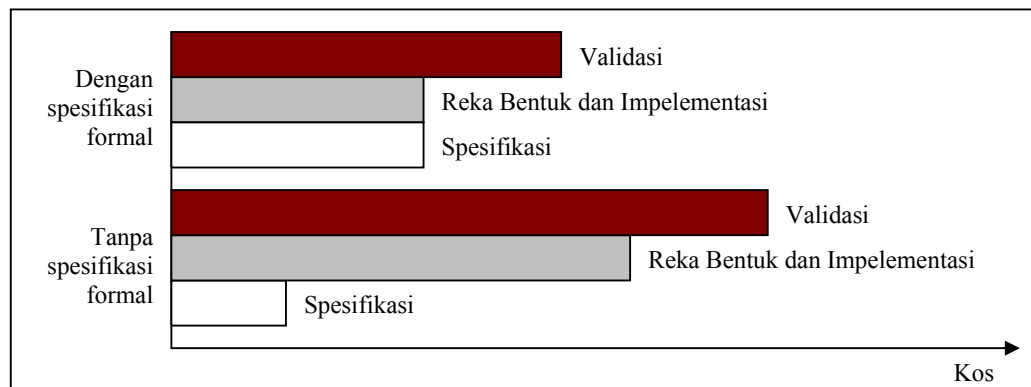


**Rajah 3.11 : Seni bina INTERRAP**

Contoh seni bina agen mobil pintar berasaskan lojik adalah Concurrent METATEM (Fisher, 1996) yang bersandarkan spesifikasi lojik temporal. Seni bina agen mobil pintar dibangunkan untuk menerangkan struktur intra dan inter agen mobil pintar. Selain itu, seni bina agen mobil pintar dapat menghuraikan tujuan, teknologi dan interaksi agen mobil pintar dengan agen lain dan persekitarannya.

### 3.5.5 Spesifikasi Formal

Implikasi spesifikasi formal ke atas kos pembangunan perisian telah dikemukakan oleh Bowen dan Hinchey (1995) sebagaimana pada Rajah 3.12. Dalam pembangunan konvensional, kos validasi adalah 50% lebih dari kos pembangunan. Kos reka bentuk dan implementasi pula adalah lebih dua kali ganda dari kos spesifikasi. Tetapi dengan penggunaan spesifikasi formal, kos spesifikasi dengan reka bentuk dan implementasi adalah sama. Sedangkan, kos validasi dikurangkan.



**Rajah 3.12 : Kos pembangunan perisian dengan spesifikasi formal**

Contoh implementasi spesifikasi formal dalam pembangunan sistem agen mobil pintar adalah dMARS (Kinny dan Georgeff, 1997) menggunakan Z, VirtualCart (Roman et al., 1997) menggunakan UNITY dan Bayou (Terry et al., 1995) menggunakan Z.

### 3.6 Bahasa Pengaturcaraan

Ciri-ciri yang perlu ada pada sesuatu bahasa pengaturcaraan untuk memastikan keupayaannya menyokong pembangunan agen mobil pintar adalah ia mestilah bebas platform, kod mudah difahami dan digunakan, berciri keselamatan, berorientasikan objek dan menyokong pembangunan komponen berautonomi, cerdik dan bergerak. Contoh bahasa pengaturcaraan yang boleh digunakan untuk membangunkan agen mobil pintar adalah Java (Bigus dan Bigus, 1999), Agent Tcl

(Kotz et al., 1997), Telescript (Tardo dan Valente, 1996), AOP (Shoham, 1993) dan PLACA (Thomas, 1993).

### **3.7 Aplikasi Agen Mobil Pintar**

Pada masa kini, teknologi agen mobil pintar telah diimplementasi secara meluas untuk menyelesaikan masalah sebenar di dalam dunia perindustrian dan komersial. Ia merangkumi teknologi pelayan-pelanggan, intranet, internet dan ekstranet.

Aplikasi perindustrian merupakan perintis pembangunan agen mobil pintar dan digunakan dalam pelbagai aplikasi industri seperti:

- (a) Pengeluaran – mengoperasi dan mengawal mesin-mesin pengilangan komponen produk yang teragih dan canggih seperti pengisar, pengecat, penghimpun dan lain-lain.
- (b) Telekomunikasi – memantau dan mengurus komponen telekomunikasi yang teragih, berskala besar dan saling berhubung.

Di dalam aplikasi komersial, hala tuju pemasaran dan transaksi perniagaan yang semakin kompleks adalah bersesuaian dengan peranan agen mobil pintar seperti:

- (a) Perniagaan elektronik – mengautomasi proses jual beli di internet termasuk tempahan, tawar-menawar dan kaedah pembayaran.
- (b) Pengurusan proses perniagaan – meningkatkan kerjasama di dalam rantaian organisasi yang teragih dan memberi sokongan keputusan.

Aplikasi hiburan adalah industri yang mencabar dan menguntungkan, di mana agen mobil pintar boleh digunakan untuk mewakili watak animasi di dalam:

- (a) Permainan komputer – membentuk pelaku-pelaku yang berautonomi dan cerdas di dalam persekitaran simulasi yang teragih.

- (b) Realiti maya – menghidupkan pelaku-pelaku di dalam realiti maya yang teragih supaya ia mempunyai ciri-ciri dan berkelakuan seperti makhluk sebenar.

Perkembangan pesat aplikasi perubatan pula membolehkan agen mobil pintar digunakan dalam aplikasi:

- (a) Pemantauan pesakit – mewakili pelaku-pelaku seperti pakar, doktor, jururawat dan lain-lain yang berkomunikasi dan bekerjasama pada lokasi yang teragih untuk merawat pesakit.
- (b) Bio-informatik – mengumpul data-data genom di dalam pangkalan data yang teragih untuk ditapis dan dikelompokkan.

### **3.8 Isu dan Masa Depan Agen Mobil Pintar**

Isu-isu yang perlu diselesaikan oleh penyelidik dan pembangun agen mobil pintar ialah:

- (a) Kerahsiaan – memastikan agen mobil pintar menjaga hak persendirian pengguna apabila bertindak mewakili pihak penggunanya.
- (b) Kekuasaan – memastikan agen mobil pintar menggunakan kuasa dan melaksanakan tanggungjawab dengan betul dan tidak disalahgunakan.
- (c) Perundangan – memastikan agen mobil pintar tidak melanggar lunas-lunas perundangan seperti melakukan pencerobohan atau menyebarkan virus.
- (d) Kesusilaan – memastikan agen mobil pintar mematuhi norma-norma positif seperti berkongsi maklumat, memohon kebenaran dan bermuafakat.

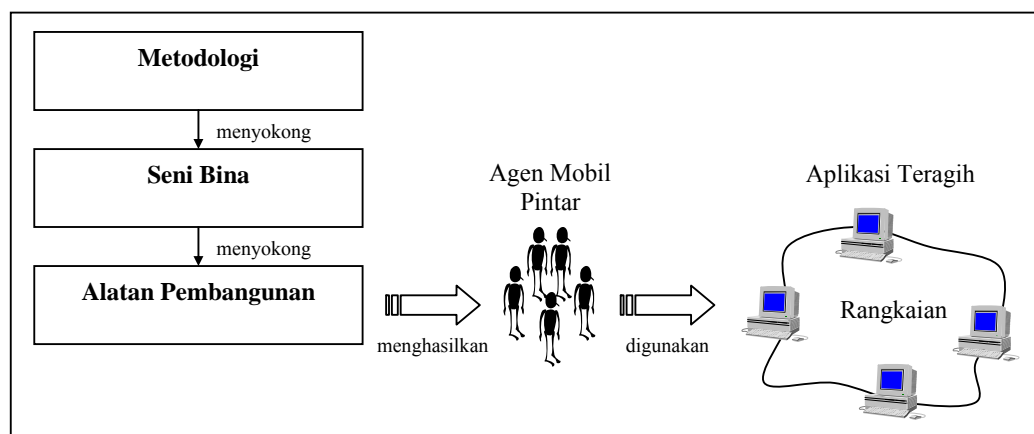
Kerja-kerja masa depan yang perlu diberi perhatian oleh penyelidik dan pembangun agen mobil pintar ialah:

- (a) Membangunkan metodologi dan teknik pemodelan yang boleh digunakan untuk menganalisa dan mereka bentuk agen mobil pintar dan aplikasinya secara sistematik.

- (b) Membangunkan rangka kerja yang boleh digunakan untuk menghasilkan seni bina agen mobil pintar yang fleksibel dan generik.
- (c) Membangunkan seni bina agen mobil pintar yang terbuka dengan mengambil kira struktur intra dan inter agen mobil pintar yang boleh digunakan untuk membina agen mobil pintar yang berkomponen, berautonomi, cerdas dan berkeupayaan bersosial.
- (d) Membuktikan struktur intra dan inter agen mobil pintar secara matematik untuk memastikan kebenaran reka bentuk seni bina agen mobil pintar.
- (e) Membangunkan piawaian dan tanda aras penilaian agen mobil pintar.
- (f) Membangunkan alatan pembangunan agen mobil pintar yang boleh digunakan untuk menghasilkan agen mobil pintar yang inter operasi di dalam aplikasi teragih.

Berdasarkan kepada isu-isu dan kerja-kerja masa depan yang telah dikupas di atas serta kepentingan agen mobil pintar dalam menangani masalah aplikasi teragih seperti yang telah dibincangkan di dalam Bab II, maka tiga elemen seperti pada Rajah 3.13 diperlukan untuk meningkatkan keupayaan agen mobil pintar. Elemen-elemen tersebut adalah seperti berikut:

- (a) Metodologi – menyediakan peraturan dan amalan pembangunan.
- (b) Seni bina – menyediakan komponen-komponen pembangunan.
- (c) Alatan pembangunan – menyediakan persekitaran dan platform pembangunan.



**Rajah 3.13 : Elemen pembangunan agen mobil pintar**

Jadual 3.1 menyenaraikan projek-projek penyelidikan dan pembangunan yang berkaitan dengan agen mobil pintar. Tujuannya adalah untuk:

- (a) Mengkaji konsistensi dan gandingan di antara elemen-elemen pembangunan agen mobil pintar.
- (b) Mengkaji teknik yang digunakan untuk membentuk struktur intra dan inter agen mobil pintar.

**Jadual 3.1 : Senarai projek berkaitan dengan agen mobil pintar**

<b>Projek</b>	<b>Metodologi</b>	<b>Seni Bina</b>	<b>Struktur Intra Ajen</b>	<b>Struktur Inter Ajen</b>	<b>Alatan Pembangunan Ajen</b>
BDIM (Busetta dan Ramamohanarao, 1998)	Tiada	BDI	BDI	Mesej	BDIM Toolkit (Java)
DECAF (Graham dan Decker, 1999)	Tiada	Reaktif	Perancangan	KQML	DECAF 2.2 (Java)
dMARS (Kinny dan Georgeff, 1997)	Metodologi BDI	BDI	BDI	Mesej	dMARS 1.6.12 (C++)
JADE (Bellifemine et al., 2001)	Tiada	Berlapis	Ontologi	FIPA	JADE 2.61 (Java)
JAFMAS (Chauhan, 1997)	Metodologi JAFMAS	Berlapis	Kerjasama	Tindakan Penuturan	JAFMAS 1.0 (Java)
JATLite (Jeon et al., 2000)	Tiada	Berlapis	Ontologi	KQML	JATLite V.0.4 (Java)
JIAC (Fricke et al., 2001)	Tiada	Berlapis	Pengetahuan Berasaskan Peraturan	KQML	JIAC 3.5 (Java)
Jinni (Tarau dan Dahl, 2001)	Tiada	Reaktif	Kotak Hitam	Mesej	Jinni 2002 (Java)
OAA (Martin et al., 1999)	Tiada	Reaktif	Kotak Hitam	ICL	OAA 2.2 (Java)
Plangent (Ohsuga et al., 1997)	Tiada	Reaktif	Perancangan	Mesej	Plangent Development Kit 1.0 (Java)

Berdasarkan kepada fakta yang telah dibentangkan pada Jadual 3.1, terdapat tiga ruang kajian yang belum diterokai, iaitu:

- (a) Ketiadaan perwakilan struktur intra agen mobil pintar berasaskan taakulan praktikal berautonomi untuk menghuraikan proses membuat kata putus yang dilakukan oleh agen mobil pintar.
- (b) Ketiadaan perwakilan struktur inter agen mobil pintar berasaskan komunikasi dan mobiliti bermesej untuk menghuraikan keupayaan bersosial agen mobil pintar.
- (c) Ketiadaan metodologi, seni bina dan alatan pembangunan agen mobil pintar yang konsisten untuk membentuk struktur intra dan inter agen mobil pintar bermula daripada penganalisan sehingga pengekodan.

Oleh kerana itu, kajian ini dijalankan untuk menghasilkan agen mobil pintar yang dibentuk oleh seni bina dengan struktur intra agen mobil pintar berasaskan taakulan praktikal berautonomi dan struktur inter agen mobil pintar berasaskan komunikasi dan mobiliti bermesej. Taakulan praktikal berautonomi adalah proses memutuskan apakah yang perlu dibuat bersamaan dengan taakulan yang dilakukan oleh manusia dalam kehidupan harian untuk mencapai sesuatu matlamat secara sendirian. Ia dibentuk oleh gabungan seni bina BDI dan autonomi. Komunikasi dan mobiliti bermesej pula dibentuk oleh gabungan konsep bahasa Mobile UNITY dan mesej.

Struktur intra agen mobil pintar diperlukan untuk membolehkan agen mobil pintar beroperasi dalam persekitaran dinamik, di mana kata putus dibuat melalui implementasi taakulan praktikal berdasarkan pentafsiran secara berautonomi terhadap struktur data kepercayaan, keinginan dan niat. Struktur inter agen mobil pintar pula diperlukan untuk membolehkan agen mobil pintar beroperasi dalam persekitaran teragih, di mana komunikasi dan mobiliti dibuat melalui hebahan mesej dan implementasi konsep atur cara, komponen dan interaksi. Tujuan struktur intra dan inter agen mobil pintar adalah untuk mengendalikan ciri-ciri agen mobil pintar di dalam aplikasi teragih iaitu serempak, cerdas, berkemampuan bergerak dan persekitaran penempatan yang teragih dan heterogen. Tujuan metodologi, seni bina dan alatan pembangunan agen mobil pintar pula adalah untuk memastikan pembinaan agen mobil pintar yang konsisten dan teratur bermula daripada penganalisan, penukaran kepada reka bentuk dan penjelmaan kepada kod-kod atur cara yang boleh dilarikan mengikut acuan struktur intra dan inter agen mobil pintar.



### **3.9 Ringkasan**

Bab ini memberi tinjauan mengenai agen mobil pintar dari aspek teori mahupun praktik dan menjelaskan aplikasi agen mobil pintar dalam pelbagai pengkhususan. Tinjauan menunjukkan bahawa penyelidikan dan pembangunan agen mobil pintar telah memberi implikasi secara langsung kepada bidang kejuruteraan perisian dan kepintaran buatan. Ini kerana agen mobil pintar perlu dibangunkan berlandaskan langkah-langkah pembangunan perisian yang sistematik dan dibenamkan teknik-teknik pintar untuk menjamin kualiti dan keupayaan operasinya. Kesimpulan yang boleh dibuat daripada tinjauan ini adalah pentingnya untuk menghasilkan agen mobil pintar yang cerdas, saling bekerjasama dan inter operasi dalam persekitaran pengkomputeran yang heterogen dan teragih. Oleh itu, kajian ini dijalankan untuk menghasilkan seni bina agen mobil pintar yang terbuka, mudah disambung, mudah dialih dan mudah diubah suai dengan mengambil kira struktur intra dan inter agen mobil pintar bagi mengatasi kekurangan sedia ada. Tujuannya untuk meningkatkan kemampuan agen mobil pintar dalam menangani masalah aplikasi teragih sebagaimana yang telah dibincangkan pada Bab II.

## **BAB IV**

### **METODOLOGI KAJIAN**

#### **4.1 Pengenalan**

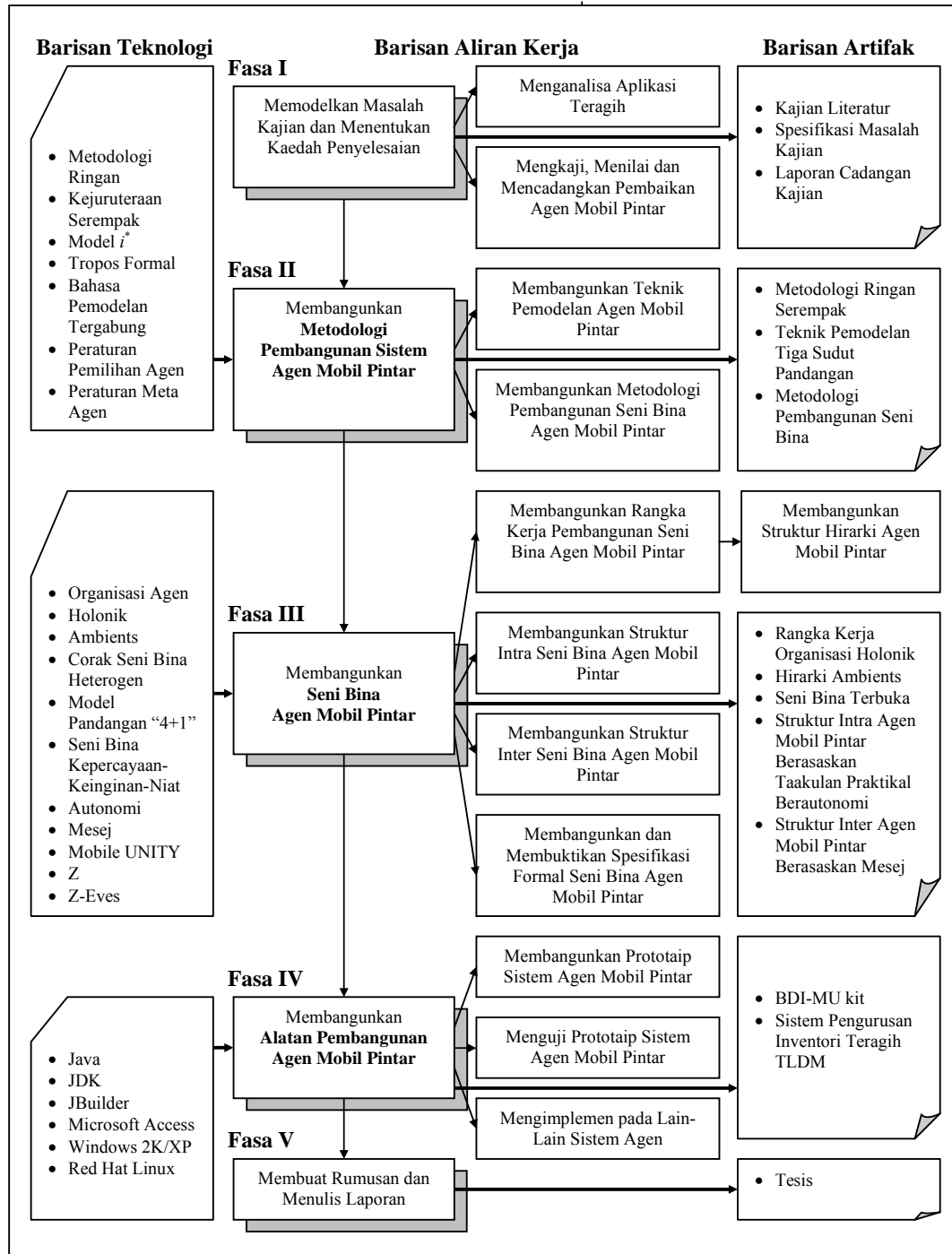
Berdasarkan kepada kajian literatur yang telah dibincangkan dalam Bab III, keperluan utama adalah untuk membangunkan seni bina terbuka agen mobil pintar bagi aplikasi teragih. Sokongan bidang kejuruteraan perisian dan kepintaran buatan diperlukan untuk membangunkan seni bina ini. Keperluan bidang kejuruteraan perisian adalah metodologi, teknik pemodelan, rangka kerja pembangunan seni bina, corak dan pandangan seni bina, spesifikasi formal dan bahasa pengaturcaraan untuk memandu proses pembangunan seni bina. Keperluan bidang kepintaran buatan pula adalah taakulan dan autonomi untuk membina struktur intra seni bina serta komunikasi dan mobiliti untuk membina struktur inter seni bina.

Secara am, bab ini menerangkan metodologi kajian yang perlu digunakan untuk mencapai matlamat kajian. Selain itu, ia juga menerangkan teknologi yang digunakan, faedah yang diperolehi dan artifak yang dihasilkan.

#### **4.2 Reka Bentuk Kajian**

Reka bentuk kajian adalah seperti ditunjukkan pada Rajah 4.1. Ia mengandungi tiga barisan iaitu barisan teknologi, barisan aliran kerja dan barisan

artifak. Barisan teknologi merangkumi pendekatan, kaedah, teknik, model dan bahasa yang digunakan untuk melaksanakan barisan aliran kerja. Barisan aliran kerja merupakan jujukan proses bersistematik yang dijalankan untuk menghasilkan barisan artifak. Barisan artifak pula adalah produk dan sumbangan kajian.



Rajah 4.1 : Reka bentuk kajian

Terdapat lima fasa yang akan dilaksanakan di dalam kajian ini, sebagaimana berikut:

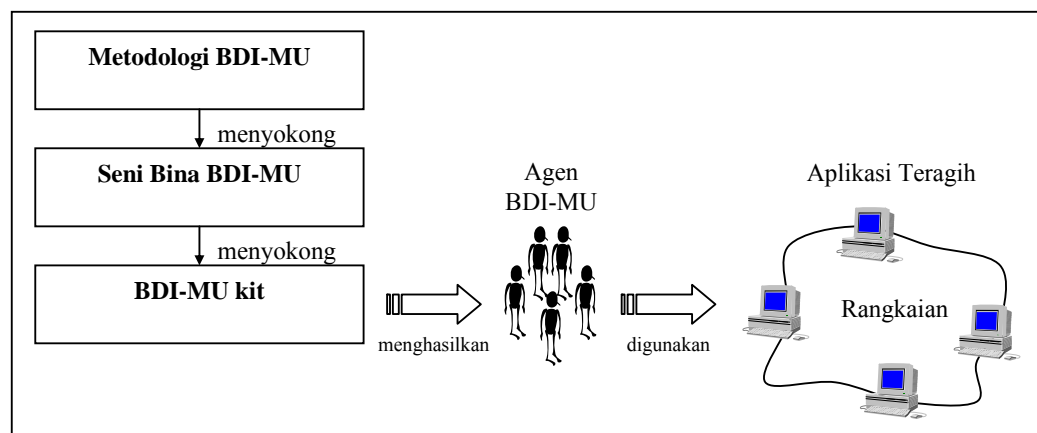
- (a) Fasa I – memodelkan masalah kajian dan menentukan kaedah penyelesaian. Tujuannya adalah untuk memahami ciri-ciri masalah aplikasi teragih, mengenal pasti kaedah penyelesaian semasa, memilih penyelesaian yang berkeupayaan mengendalikan masalah aplikasi teragih dan mencadangkan pembaikan bagi meningkatkan keupayaan kaedah yang dipilih.
- (b) Fasa II – membangunkan metodologi pembangunan agen mobil pintar. Tujuannya adalah untuk menghasilkan metodologi yang mengandungi peraturan dan amalan pembangunan sistem agen mobil pintar yang sistematik. Metodologi tersebut boleh digunakan untuk melaksanakan proses-proses pembangunan sistem agen mobil pintar bermula daripada mengenal pasti keperluan sehingga menempatkan sistem agen mobil pintar di dalam persekitaran operasi sebenar.
- (c) Fasa III – membangunkan seni bina agen mobil pintar. Tujuannya adalah untuk menghasilkan seni bina yang mengandungi pakej-pakej dan kelas-kelas yang boleh diimport dan disambung untuk membentuk struktur intra dan inter agen mobil pintar. Seni bina tersebut juga adalah terbuka bagi membolehkannya digunakan ke atas pelbagai domain aplikasi teragih berasaskan agen mobil pintar.
- (d) Fasa IV – membangunkan alatan pembangunan agen mobil pintar. Tujuannya adalah untuk menghasilkan alatan pembangunan agen mobil pintar yang boleh digunakan untuk mengekod agen mobil pintar dalam bentuk kod-kod atur cara. Alatan pembangunan tersebut juga menyediakan persekitaran operasi untuk agen mobil pintar.
- (e) Fasa V – membuat rumusan dan menulis laporan. Tujuannya adalah untuk menguji, menilai dan menyimpulkan kemampuan metodologi, seni bina dan alatan pembangunan yang dihasilkan dalam mengendalikan masalah aplikasi teragih.

Kajian ini dijalankan berasaskan konsep sains gunaan, di mana kajian kes adalah sistem pengurusan inventori teragih dan domain kajian pula adalah seperti berikut:

- (a) Domain inventori yang difokuskan kepada proses pengurusan inventori teragih bagi material kapal TLDM.
- (b) Domain sains komputer yang difokuskan kepada bidang kepintaran buatan dan kejuruteraan perisian. Pengkhususan kepintaran buatan adalah dalam taakulan, mobiliti dan komunikasi agen mobil pintar. Pengkhususan bidang kejuruteraan perisian pula adalah dalam proses pembangunan agen mobil pintar dan sistem berasaskannya.

### 4.3 Faedah dan Produk Akhir Kajian

Produk akhir kajian adalah metodologi BDI-MU, seni bina BDI-MU dan BDI-MU kit sepertimana yang ditunjukkan pada Rajah 4.2.



**Rajah 4.2 : Produk kajian**

Produk pertama adalah metodologi BDI-MU yang berkelas ringan dan berjenis serempak. Ia mengandungi enam fasa iaitu keperluan, analisa, reka bentuk, implementasi, integrasi dan penempatan yang boleh digunakan untuk membangunkan sistem berasaskan agen mobil pintar untuk aplikasi teragih. Produk kedua adalah seni bina BDI-MU yang berkategori terbuka. Ia mengandungi lapan pakej iaitu domain, pelayan, konfigurasi, agen, MobileUNITY, BDI, aplikasi dan pangkalan data yang boleh diimport, disambung dan diubah suai untuk membentuk seni bina sistem berasaskan agen mobil pintar. Produk ketiga adalah BDI-MU kit

yang dibangunkan menggunakan Java. Kelebihan BDI-MU kit berbanding dengan alatan lain seperti JATLite, JADE, AgentBuilder, Zeus, Concordia dan lain-lain adalah seperti berikut:

- (a) Berasaskan taakulan praktikal dan menyokong pembangunan komponen dinamik yang bergerak, serempak dan saling berinteraksi.
- (b) Menyokong pembangunan sistem pelbagai agen yang beroperasi dalam persekitaran teragih dan heterogen untuk pelbagai domain.
- (c) Boleh dihibridkan dengan teknik pintar seperti algoritma genetik, logik lalai, set kasar, rangkaian neural, taakulan berasaskan kes dan lain-lain untuk meningkatkan keupayaan agen mobil pintar.

#### **4.4 Ringkasan**

Metodologi kajian yang digunakan adalah seperti pada Rajah 4.1, ia merangkumi teknologi yang dipakai, aliran kerja yang dilaksanakan dan artifak yang dihasilkan. Implementasi setiap fasa dan kaitannya dengan struktur tesis adalah seperti berikut:

- (a) Fasa I – dibincangkan dalam Bab I sehingga Bab III.
- (b) Fasa II – dibincangkan dalam Bab V dan Bab VI.
- (c) Fasa III – dibincangkan dalam Bab VII sehingga Bab XI.
- (d) Fasa IV – dibincangkan dalam Bab XII.
- (e) Fasa V – dibincangkan dalam Bab XIII.

## **BAB V**

### **METODOLOGI BDI-MU: METODOLOGI RINGAN SEREMPAK UNTUK PEMBANGUNAN SISTEM BERASASKAN AGEN MOBIL PINTAR**

#### **5.1 Pengenalan**

Perkembangan teknologi agen telah mencetuskan kejuruteraan perisian berasaskan agen (Wooldridge, 1997) dan kejuruteraan sistem pelbagai agen (Muller, 1995) yang memberi anjakan di dalam bidang kejuruteraan perisian khususnya ke atas pembangunan agen dan sistem agen. Seterusnya ia membawa kepada era teknologi agen pintar (Wooldridge dan Jennings, 1994), agen mobil (Straber dan Rothermel, 1998) dan agen mobil pintar (McGrath et al., 2000). Keadaan ini mendesak kepada penyelidikan yang lebih intensif untuk menghasilkan metodologi berorientasikan agen sebagaimana yang dinyatakan oleh Jennings dan Wooldridge (1995), Fisher et al. (1997) dan Jennings (2001). Metodologi yang dibina seharusnya berupaya menyokong pembangunan agen yang akan diimplementasi dalam platform dan domain yang berlainan dengan pelbagai skala, kompleksiti dan senario.

Objektif bab ini adalah untuk mengetengahkan metodologi ringan serempak berorientasikan agen mobil pintar sebagai satu langkah permulaan ke arah cetusan istilah kejuruteraan perisian berasaskan agen mobil pintar. Satu bidang yang diramal akan menjadi penting dan relevan dengan teknologi masa hadapan selari dengan pernyataan oleh Jennings et al. (1998). Di dalam bab ini juga, beberapa contoh diberikan untuk menunjukkan bagaimana metodologi yang dihasilkan dapat digunakan dalam sistem pengurusan inventori teragih berasaskan agen mobil pintar.

## 5.2 Metodologi Ringan dan Kejuruteraan Serempak

Metodologi adalah satu set peraturan dan amalan yang digunakan untuk membangunkan perisian. Metodologi dikategorikan kepada metodologi berat dan metodologi ringan (Mayhew, 1998). Metodologi berat mempunyai peraturan, amalan dan artifak yang banyak serta memerlukan disiplin dan masa yang lebih untuk mengikutinya dengan betul. Artifak adalah produk kerja akhir atau pertengahan yang dihasilkan dan digunakan semasa pembangunan perisian. Artifak mengandungi maklumat-maklumat projek dan digunakan untuk memudahkan pemahaman mengenai perisian yang dibangunkan. Artifak terdiri daripada berikut:

- (a) Dokumentasi seperti dokumen spesifikasi keperluan perisian, dokumen seni bina perisian dan manual pengguna.
- (b) Model seperti model kesgunaan, model data dan model pengujian.
- (c) Komponen seperti kod sumber, fail boleh laku dan pangkalan data.

Metodologi ringan pula mempunyai peraturan, amalan dan artifak yang sedikit serta mudah diikuti. Metodologi ringan sesuai untuk projek dengan jangka masa yang singkat dan kos pembangunan yang terhad. Extreme Programming (Paulk, 2001), SCRUM (Rising dan Janoff, 2000) dan DSDM (Millington dan Stapleton, 1995) adalah contoh metodologi ringan yang telah diimplementasi untuk menghasilkan perisian yang tegap, boleh dipercayai dan boleh beroperasi dalam pelbagai domain.

Kejuruteraan serempak adalah pendekatan sistematik untuk pembangunan perisian secara bersepadu dan serentak (Pohl dan Jacobs, 1994). Kombinasi metodologi ringan dengan kejuruteraan serempak adalah gabungan yang sepadan untuk menangani pembangunan perisian dengan jangka masa yang singkat. Di dalam kejuruteraan serempak, fasa-fasa boleh saling bertindih bagi membenarkan aktiviti-aktiviti tertentu dalam fasa yang berlainan dijalankan secara serentak.

## 5.3 Metodologi BDI-MU

Terdapat enam fasa di dalam metodologi BDI-MU yang perlu dilaksanakan untuk membangunkan sistem berasaskan agen mobil pintar, iaitu:



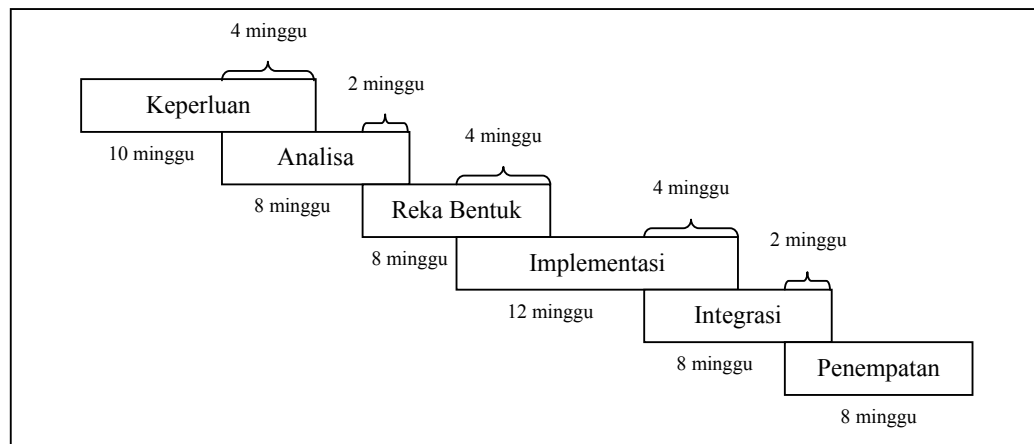
- (a) Keperluan – keperluan pengguna, keperluan kefungsiian dan keperluan bukan kefungsiian ditakrifkan dalam bentuk yang difahami oleh pengguna dan pembangun sistem agen.
- (b) Analisa – domain agen dimodelkan dan difahami, agen diidentifikasi dan diklasifikasikan serta model intra dan inter agen dibangunkan.
- (c) Reka bentuk – seni bina agen dibangunkan dan setiap komponen agen ditakrifkan sehingga dapat dijelmakan kepada atur cara yang boleh dilarikan.
- (d) Implementasi – reka bentuk agen diterjemah kepada satu set atur cara atau unit atur cara dan setiap satunya diuji secara berasingan.
- (e) Integrasi – unit atur cara agen dan bukan agen digabungkan dan diuji sebagai satu sistem agen yang lengkap.
- (f) Penempatan – sistem agen ditempatkan dan digunakan secara praktikal. Senggaraan dilakukan untuk membaiki ralat, meningkatkan kemampuan sistem agen atau menambah fungsi sistem agen.

Rajah 5.1 menggambarkan implementasi kejuruteraan serempak ke atas metodologi BDI-MU dengan menggunakan model air terjun (Royce, 1970), di mana masa pembangunan projek adalah terhad dan perlu disiapkan dalam masa 5 bulan atau hampir 22 minggu. Kejuruteraan serempak membolehkan aktiviti dalam fasa yang berlainan dilaksanakan secara serentak. Contohnya, aktiviti mengurus skop keperluan, menyaring keperluan dan mengurus perubahan keperluan dilaksanakan bersama-sama dengan aktiviti memodelkan domain. Senario ini membolehkan jangka masa bagi pelaksanaan fasa keperluan dan analisa disingkatkan selama 4 minggu daripada 18 minggu yang telah diperuntukkan. Secara umumnya, pelaksanaan kejuruteraan serempak dalam metodologi BDI-MU bertujuan untuk menangani kekangan masa pembangunan dan kos projek. Ia merangkumi prinsip-prinsip asas seperti berikut:

- (a) Kerjasama dan komunikasi berterusan di antara pembangun sistem agen dengan pelbagai kumpulan pengguna sistem agen yang teragih. Tujuannya untuk mengesahkan dan membaiki spesifikasi, reka bentuk dan larian sistem agen dengan segera.
- (b) Pelaksanaan aktiviti secara selari dengan mengintegrasikan aktiviti-aktiviti fasa yang berlainan. Tujuannya untuk menghapuskan

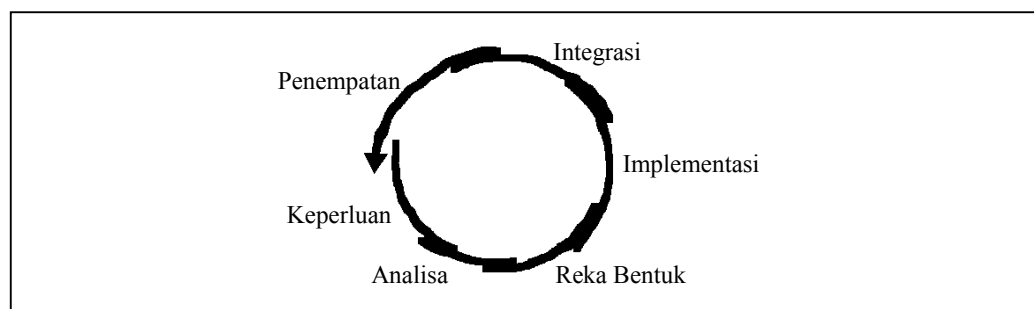
pertindihan tugas yang sama dan mengurangkan masa lengah ketika proses pemindahan tugas.

- (c) Proses pengemaskinian dilaksanakan secara berterusan melalui beberapa kitaran versi dan pertambahan fungsi yang sentiasa berkembang. Tujuannya untuk membolehkan fungsi-fungsi kritikal sistem agen dibangunkan terlebih dahulu dan diuji pelaksanaannya.
- (d) Pemeriksaan semula kemajuan kerja dan memperbaiki perancangan kerja melalui batu tanda dan pemecahan masalah secara bermodular. Tujuannya untuk membolehkan masalah yang kompleks dilarikan kepada beberapa submasalah dan ditentukan keutamaannya.



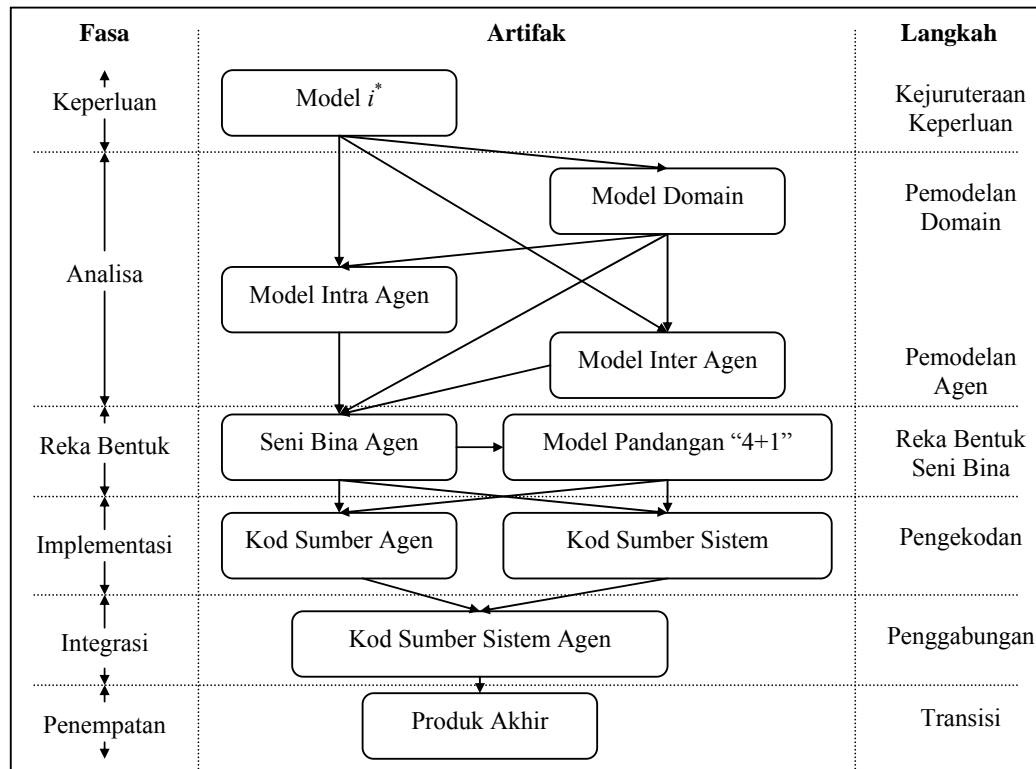
**Rajah 5.1 : Implementasi kejuruteraan serempak ke atas metodologi BDI-MU menggunakan model air terjun**

Rajah 5.2 pula menunjukkan implementasi kejuruteraan serempak ke atas metodologi BDI-MU jika menggunakan model putaran (Boehm, 1988).



**Rajah 5.2 : Implementasi kejuruteraan serempak ke atas metodologi BDI-MU menggunakan model putaran**

Gambaran keseluruhan metodologi BDI-MU adalah seperti pada Rajah 5.3. Semasa fasa analisa, artifak seperti model domain, model intra agen dan model inter agen dihasilkan. Di dalam fasa reka bentuk pula seni bina agen dan model pandangan “4+1” dibangunkan.



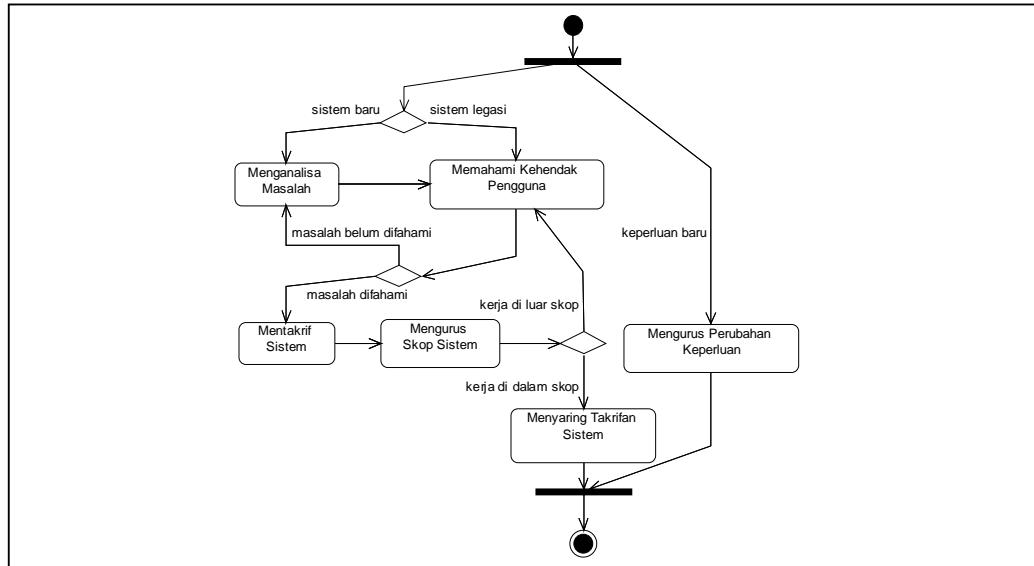
**Rajah 5.3 : Metodologi BDI-MU**

Proses pengesahan ke atas artifak yang dihasilkan pada setiap fasa dalam metodologi BDI-MU dibuat melalui cara berikut:

- (a) Penyemakan dengan artifak yang dihasilkan pada fasa sebelumnya. Contohnya, fungsi dan larian sistem agen yang telah dikodkan oleh pembangun sistem akan disemak dengan keperluan kefungsiian.
- (b) Pengesahan secara berterusan oleh pengguna sistem agen. Contohnya, penambahan atau perubahan keperluan dalam model  $i^*$  akan disahkan oleh pengguna sistem agen.
- (c) Pengesahan secara formal menggunakan spesifikasi Z. Contohnya, spesifikasi seni bina agen ditulis menggunakan bahasa Z untuk disahkan sintaks dan semantiknya menggunakan alatan Z-Eves.

### 5.3.1 Fasa Keperluan

Di dalam fasa keperluan, fokus diberikan kepada usaha mengumpulkan keperluan pengguna dan sistem serta menentukan skop sistem seperti pada Rajah 5.4.



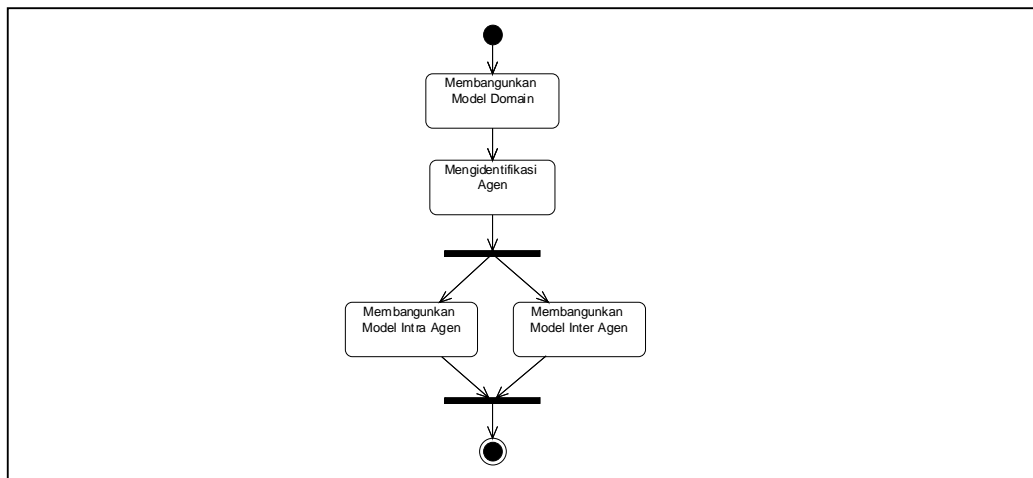
**Rajah 5.4 : Aktiviti dalam fasa keperluan**

Model  $i^*$  (Yu, 1995) digunakan untuk memodelkan keperluan. Model  $i^*$  adalah rangka kerja pemodelan dengan konsep pelaku dan kebersandaran sosial seperti kebersandaran matlamat, matlamat lembut, tugas dan sumber. Pelaku merupakan agen, posisi atau peranan yang wujud dalam organisasi. Kebersandaran matlamat mewakili delegasi atau tanggungjawab pelaku untuk mencapai matlamat. Kebersandaran matlamat lembut menyamai kebersandaran matlamat tetapi pencapaiannya tidak dapat ditakrifkan secara tepat tetapi subjektif. Kebersandaran tugas adalah situasi di mana pelaku yang menerima sandaran dikehendaki melaksanakan aktiviti yang telah diberikan kepadanya. Kebersandaran sumber pula adalah keadaan di mana pelaku yang menerima sandaran perlu menyediakan sumber kepada pelaku yang bersandar. Melalui model  $i^*$ , sistem dilihat sebagai satu organisasi dengan pelaku-pelaku yang mempunyai matlamat tersendiri dan saling bergantung untuk mencapai matlamat-matlamat tersebut. Model  $i^*$  telah diimplementasi dalam kejuruteraan keperluan (Mylopoulos et al., 1999), proses perisian (Mylopoulos et al., 1997) dan kejuruteraan proses perniagaan (Yu dan

Mylopoulos, 1996). Setelah dimodelkan, keperluan akan ditulis menggunakan Tropos Formal (Fuxman, 2001).

### 5.3.2 Fasa Analisa

Matlamat fasa analisa adalah untuk memodelkan domain agen bagi membolehkan aktiviti mengidentifikasi dan mengklasifikasi agen dapat dilakukan. Seterusnya memudahkan aktiviti menganalisa struktur intra dan inter agen. Keseluruhan aktiviti di dalam fasa analisa ditunjukkan pada Rajah 5.5.



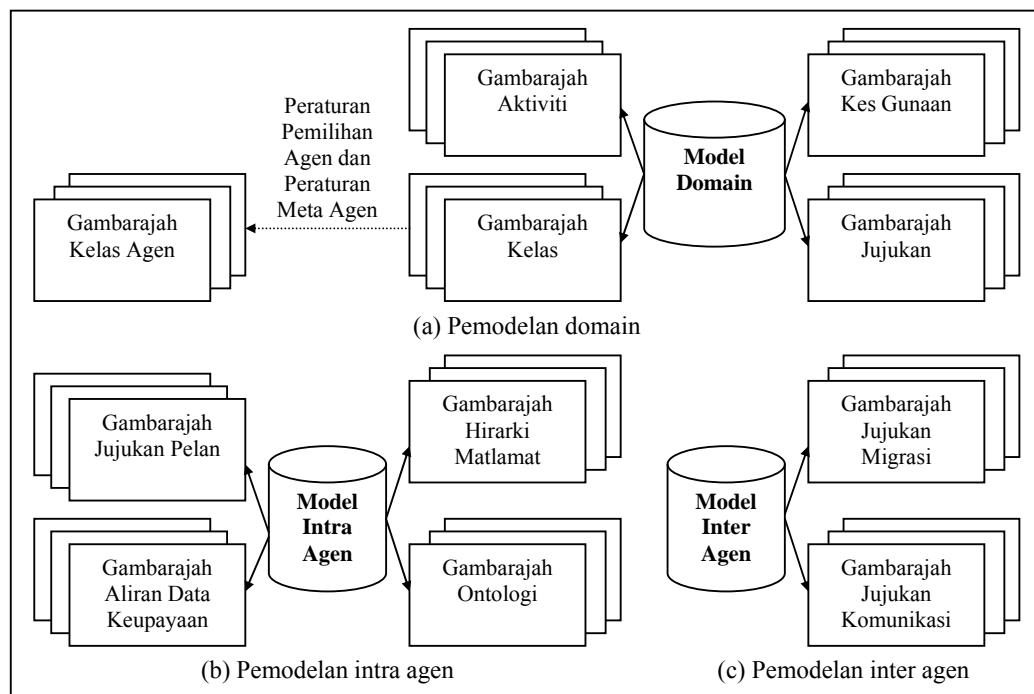
**Rajah 5.5 : Aktiviti dalam fasa analisa**

Teknik pemodelan BDI-MU digunakan secara konstruktif sebagai teknik pemodelan dalam fasa analisa. Teknik pemodelan ini menganalisa sistem agen dari sudut pandangan domain, pandangan intra agen dan pandangan inter agen. Tiga model utama seperti pada Rajah 5.6 iaitu model domain, model intra agen dan model inter agen akan dibangunkan untuk menggambarkan setiap pandangan tersebut:

- (a) Model domain (Rajah 5.6 (a)) – menerangkan atribut dan kelakuan sistem agen dari aspek yang statik dan dinamik serta menerangkan bagaimana proses pengecaman agen dilaksanakan. Gambarajah kesgunaan, gambarajah jujukan, gambarajah aktiviti dan gambarajah kelas dibangunkan untuk memodelkan domain agen. Peraturan

pemilihan agen dan peraturan meta agen digunakan untuk mengecam agen. Gambarajah kelas agen pula dibina untuk menerangkan atribut dan operasi agen serta hubungan di antara agen dengan objek.

- (b) Model intra agen (Rajah 5.6 (b)) – menerangkan struktur dalaman agen seperti matlamat, pelan, kepercayaan dan keupayaan. Gambarajah hirarki matlamat, gambarajah jujukan pelan, gambarajah ontologi dan gambarajah aliran data keupayaan dibangunkan untuk memodelkan struktur intra agen.
- (c) Model inter agen (Rajah 5.6 (c)) – menerangkan mobiliti dan komunikasi agen melalui pembangunan gambarajah jujukan migrasi dan gambarajah jujukan komunikasi.

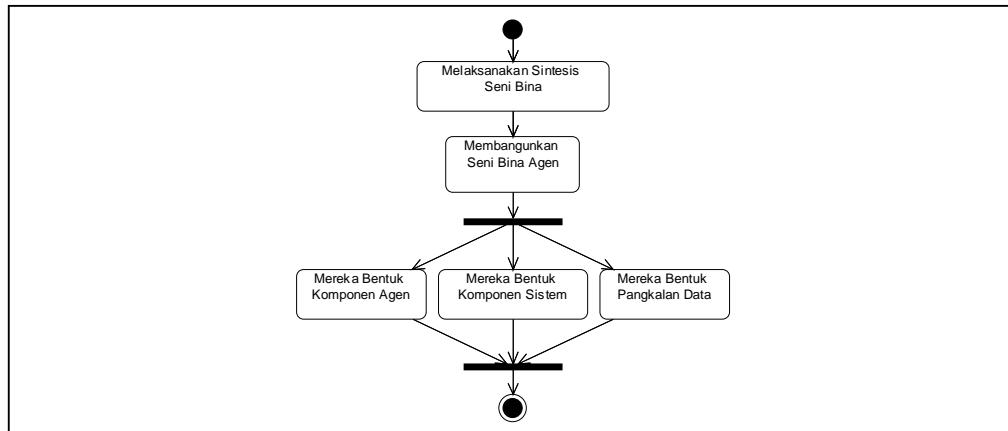


**Rajah 5.6 : Model-model dalam teknik pemodelan BDI-MU**

### 5.3.3 Fasa Reka Bentuk

Aktiviti utama di dalam fasa reka bentuk adalah membangunkan seni bina agen. Seni bina agen digunakan untuk menghuraikan struktur sistem agen yang

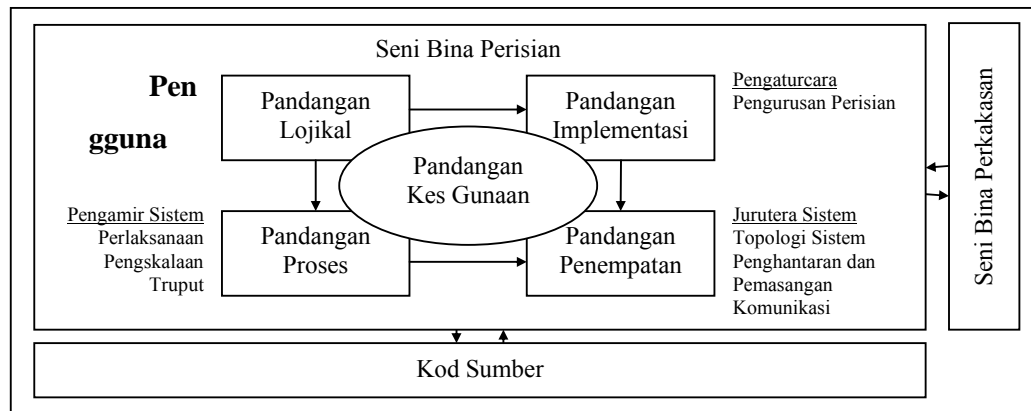
terdiri daripada komponen agen dan komponen bukan agen, atribut dan operasi komponen serta hubungan di antara komponen. Aktiviti-aktiviti yang perlu dilaksanakan dalam fasa reka bentuk adalah seperti pada Rajah 5.7.



**Rajah 5.7 : Aktiviti dalam fasa reka bentuk**

Model pandangan “4+1” (Kruchten, 1995) yang diimplementasi oleh proses tergabung rasional (*rational unified process*, RUP) sebagai pandangan seni bina digunakan untuk menghuraikan seni bina agen. Model pandangan “4+1” terdiri daripada pandangan lojikal, pandangan proses, pandangan implementasi, pandangan penempatan dan pandangan kes guna seperti pada Rajah 5.8:

- (a) Pandangan lojikal – menerangkan keperluan kefungsi-an agen dan sistem agen.
- (b) Pandangan proses – menerangkan aspek serempak dan segerak yang wujud di dalam agen dan sistem agen seperti tugas, proses dan bebenang termasuk interaksi di antaranya.
- (c) Pandangan implementasi – menerangkan organisasi modul-modul statik agen dan sistem agen seperti kod sumber, fail data dan fail boleh laku.
- (d) Pandangan penempatan – menerangkan bagaimana komponen boleh laku dan masa larian dipetakan kepada platform dan nod-nod komputasi.
- (e) Pandangan kes guna – menerangkan senario agen dan sistem agen termasuk peranan pelaku-pelaku yang terlibat.



**Rajah 5.8 : Model pandangan “4+1”**

Model pandangan “4+1” dapat digambarkan oleh beberapa model seperti pada Jadual 5.1, iaitu model reka bentuk, model proses, model implementasi, model penempatan dan model kesgunaan (Egyed dan Kruchten, 1999).

**Jadual 5.1 : Hubungan model dengan pandangan seni bina**

Model	Pandangan seni bina
Model reka bentuk	Pandangan lojikal
Model proses	Pandangan proses
Model implementasi	Pandangan implementasi
Model penempatan	Pandangan penempatan
Model kesgunaan	Pandangan kesgunaan

### 5.3.3.1 Metodologi Pembangunan Seni Bina Ajen

Seni bina agen adalah artifak penting dalam pembangunan sistem agen yang berskala besar dan kritikal. Oleh kerana itu, metodologi pembangunan seni bina agen diperlukan dalam metodologi BDI-MU. Ia bertujuan untuk memberi pemahaman tentang fungsi dan kegunaan sistem agen, memahami bagaimana sistem agen bekerja, memudahkan penyambungan sistem agen dan membolehkan komponen sistem agen diguna semula untuk membina yang baru. Terdapat beberapa metodologi yang boleh digunakan sebagai garis panduan untuk membangunkan seni bina agen

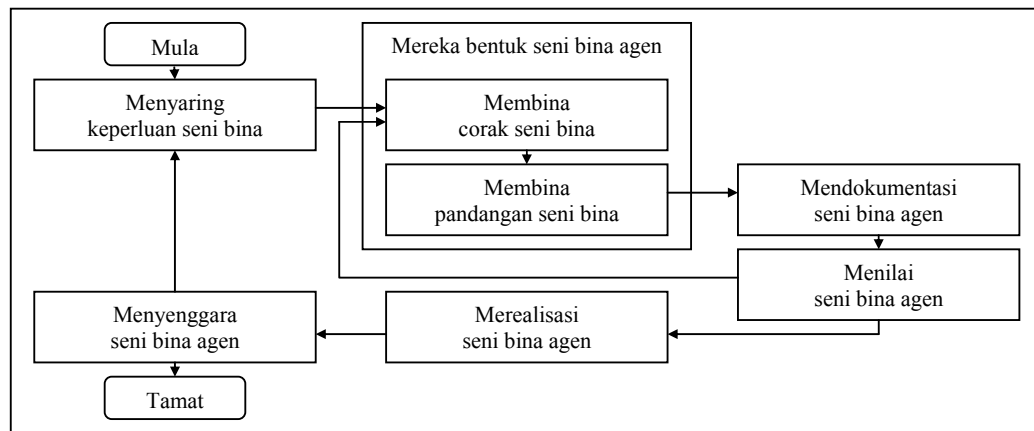


seperti analisa berasaskan kes gunaan (Jacobson et al., 1995), analisa berasaskan senario (Kazman et al., 1999) dan reka bentuk berorientasikan objek (Bryson, 2000). Bagaimanapun, metodologi tersebut tidak memberi penekanan kepada proses reka bentuk dan senggaraan serta tidak menjelaskan bagaimana seni bina agen didokumentasi, dinilai dan direalisasikan.

Selari dengan keperluan di atas, metodologi pembangunan seni bina agen seperti pada Rajah 5.9 dihasilkan. Ia merangkumi proses-proses seperti berikut:

- (a) Menyaring keperluan seni bina – bertujuan mengasingkan keperluan seni bina agen dengan keperluan sistem agen. Inputnya adalah keperluan pengguna, persekitaran teknikal dan pengalaman arkitek sistem.
- (b) Mereka bentuk seni bina agen – bertujuan membina struktur seni bina agen mengikut corak seni bina tertentu dan menghuraikannya mengikut pandangan seni bina. Inputnya adalah senario kualiti, gaya seni bina, corak reka bentuk dan keperluan seni bina.
- (c) Mendokumentasi seni bina agen – bertujuan memudahkan senggaraan dan komunikasi di antara pengguna, arkitek sistem dan pengaturcara. Inputnya adalah komunikasi dan analisa oleh pengguna, arkitek sistem dan pengaturcara.
- (d) Menilai seni bina agen – bertujuan menyemak seni bina agen, mengidentifikasi risiko dan memastikan keperluan kualiti seperti kebolegunaan, kebolehpayaan, kebolehppercayaan dan kebolehsenggaraan dipenuhi. Inputnya adalah penyemak luaran, pengguna dan keperluan kualiti.
- (e) Merealisasi seni bina agen – bertujuan menterjemah seni bina agen kepada kod-kod atur cara yang boleh dilarikan. Ia bagi membolehkan komponen-komponen seni bina agen diuji, dikonfigurasi dan dibetulkan perlaksanaannya. Proses ini dilaksanakan bersama-sama dengan fasa implementasi yang dibincangkan dalam Seksyen 5.3.4.
- (f) Menyenggara seni bina agen – bertujuan menyelaras di antara reka bentuk dengan implementasi. Ia bagi memastikan seni bina agen adalah konsisten, rasional dan berkualiti. Meja kerja Dali (Kazman dan Carriere, 1999) digunakan sebagai alatan untuk mendokumentasi

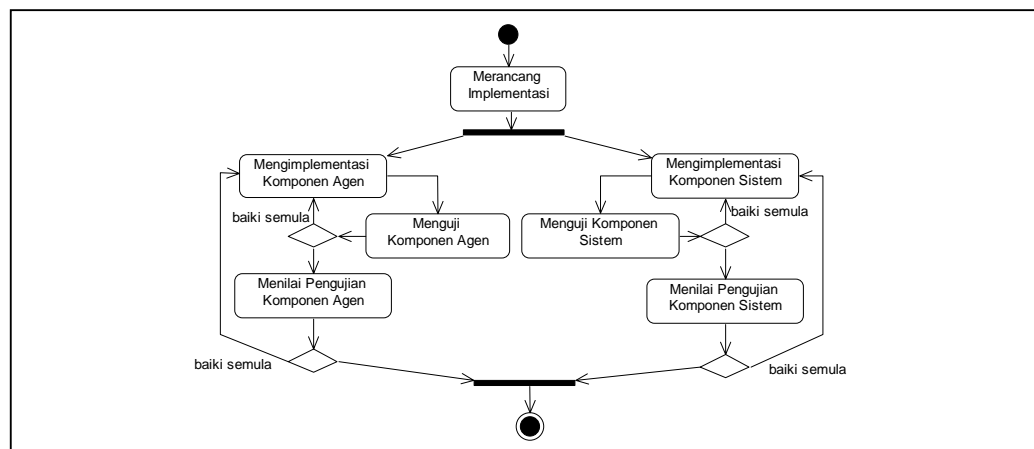
semula seni bina agen daripada kod sumber bagi memudahkan senggaraan seni bina agen.



**Rajah 5.9 : Proses pembangunan seni bina agen**

#### 5.3.4 Fasa Implementasi

Fasa implementasi dijalankan untuk menukarkan reka bentuk komponen agen dan komponen sistem kepada unit-unit atur cara yang boleh dilarikan. Setiap unit atur cara akan diuji dan dinilai bagi memastikan ia bebas daripada kecacatan. Rajah 5.10 menunjukkan aktiviti-aktiviti yang terlibat dalam fasa implementasi.

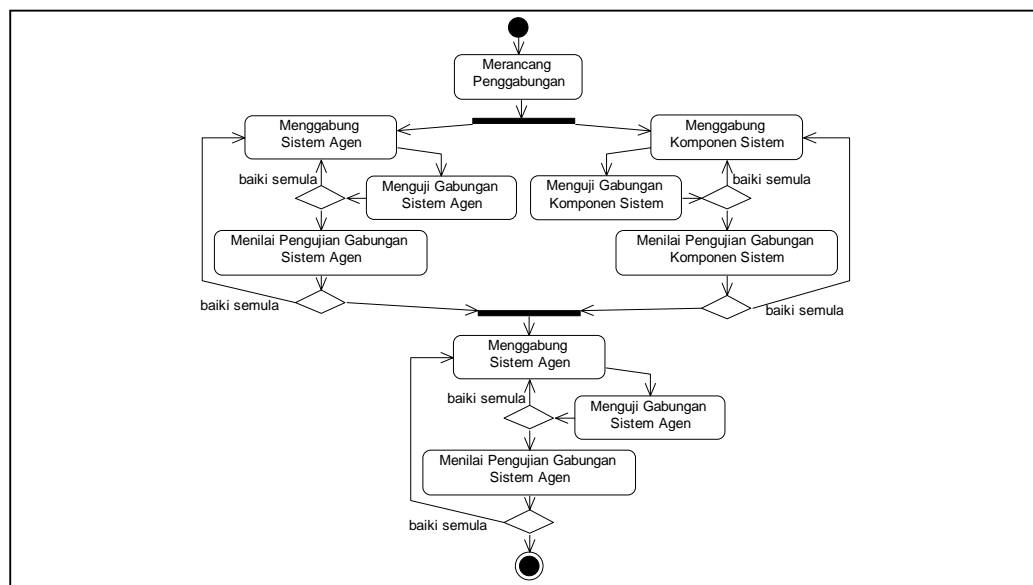


**Rajah 5.10 : Aktiviti dalam fasa implementasi**

Komponen agen dikodkan menggunakan Java, manakala kod sistem boleh dikodkan menggunakan sebarang bahasa pengaturcaraan berorientasikan objek. Java dipilih untuk membina agen kerana kod sumber Java (\*.java) tidak diterjemahkan kepada arahan mesin tetapi dikompil kepada kod bait (\*.class) yang mudah dialih dan boleh dilarikan pada sistem komputer yang mempunyai mesin maya Java (*Java virtual machine*, JVM). Keadaan ini membolehkan kod atur cara ditulis sekali, dipindahkan di sepanjang rangkaian dan dilarikan pada sebarang sistem pengoperasian. Java juga menyokong pembangunan komponen berautonomi dan cerdas melalui konsep pelbagai bebenang, pengisirian dan objek serta pakej keadaan, pemerhati dan rangkaian yang boleh diimport dan disambung.

### 5.3.5 Fasa Integrasi

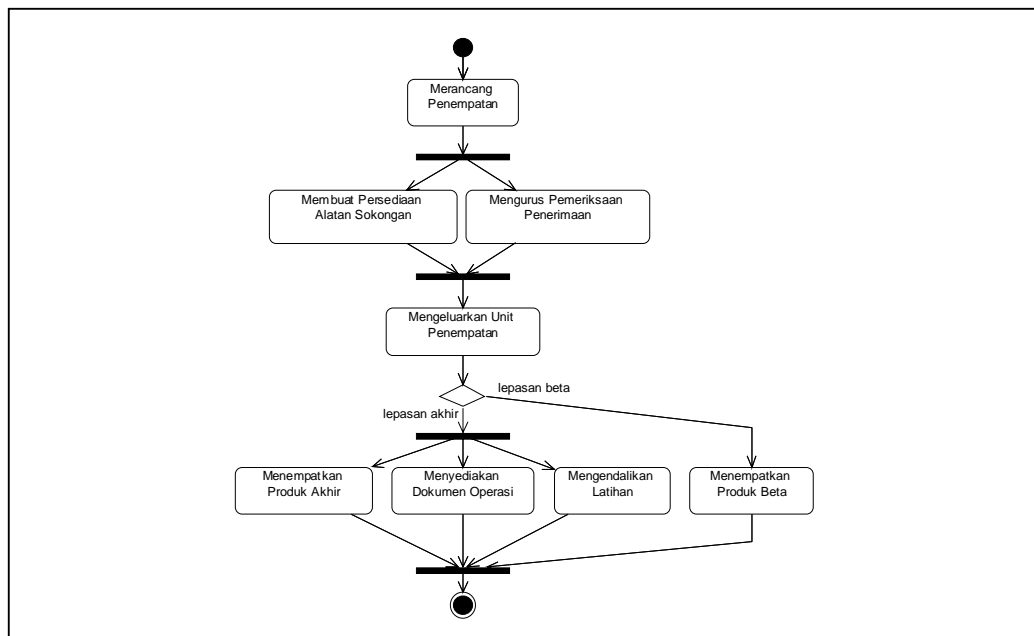
Komponen agen dan komponen sistem digabungkan dan dinilai sebagai sistem agen dalam fasa integrasi. Proses penggabungan dilaksanakan secara bawah-atas dari unit-unit yang kecil dikumulatif kepada sistem agen yang lengkap. Ia bagi menjamin konsistensi sistem agen dan memastikan komponen berkeutamaan tinggi berfungsi dengan sempurna. Aktiviti penggabungan adalah seperti pada Rajah 5.11.



Rajah 5.11 : Aktiviti dalam fasa integrasi

### 5.3.6 Fasa Penempatan

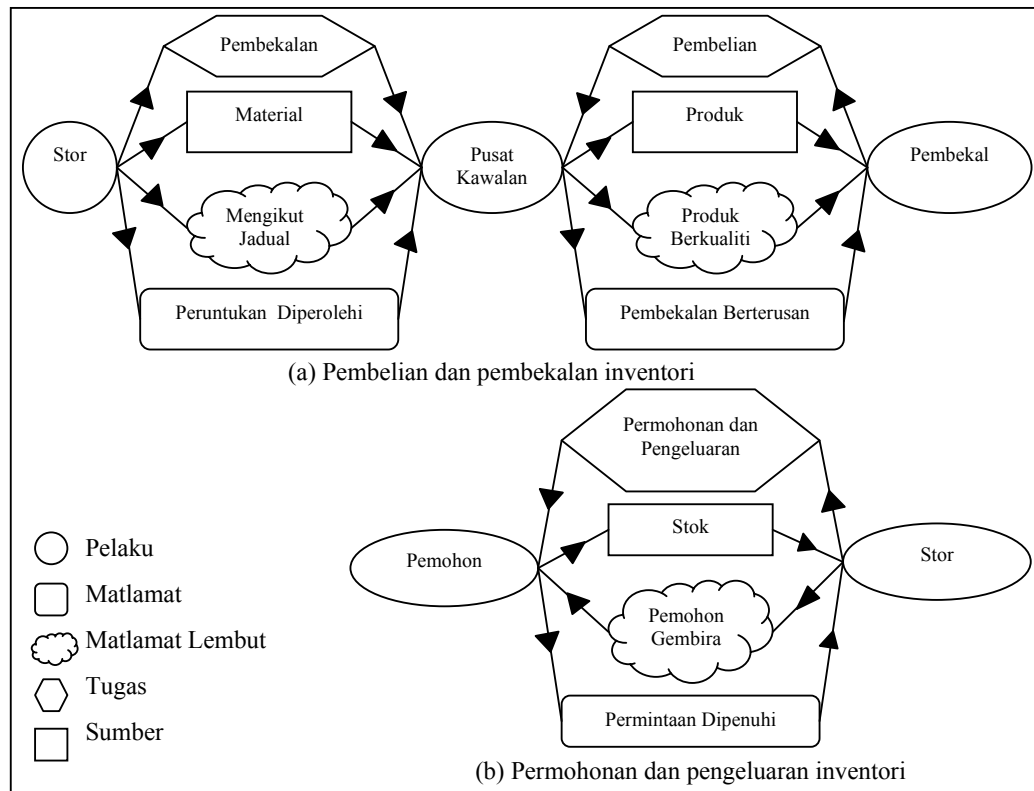
Fasa penempatan berfungsi untuk memastikan sistem agen berfungsi dan sedia digunakan oleh pengguna. Semasa sistem agen ditempatkan secara praktikal, penekanan dibuat ke atas menguji sistem agen di persekitaran sebenar. Tujuannya untuk menjadikan sistem agen bertambah teguh dan matang. Rajah 5.12 menunjukkan aktiviti-aktiviti dalam fasa penempatan.



Rajah 5.12 : Aktiviti dalam fasa penempatan

## 5.4 Implementasi Metodologi BDI-MU dalam Pembangunan Sistem Pengurusan Inventori Teragih

Rajah 5.13 menggambarkan model  $i^*$  untuk pengurusan inventori bagi fungsi pembelian, pembekalan, permohonan dan pengeluaran inventori. Pembekal, pusat kawalan, stor dan pemohon adalah pelaku-pelaku yang telah dikenal pasti. Pemohon bersandar kepada stor untuk memenuhi matlamatnya iaitu permintaan dipenuhi. Stor pula bersandar kepada pemohon untuk membuat permohonan dan pengeluaran stok serta memastikan pemohon gembira.



**Rajah 5.13 : Model  $i^*$  untuk pengurusan inventori**

Contoh takrifan keperluan bagi pembekalan inventori menggunakan Tropos

Formal adalah seperti berikut:

<i>Entiti</i>	Produk
<i>Atribut</i>	kod : rentetan, jenis : JenisProduk, harga : nombor, ...
<i>Kebersandaran</i>	Pembekalan Berterusan
<i>Jenis matlamat</i>	Capai
<i>Pemberi sandaran</i>	Pusat Kawalan
<i>Penerima sandaran</i>	Pembekal
<i>Atribut</i>	item : Produk
<i>Penunaian</i>	
<i>Syarat kepada pemberi sandaran</i>	$\forall \text{produk} : \text{Produk}(\text{item.jenis} =$ $\text{produk.jenis} \rightarrow \text{item.harga} \leq \text{produk.harga})$ (harga yang berpatutan dijangkakan dari pembekal)

Perlaksanaan fasa-fasa yang lain pula dibentangkan dalam Bab VI, Bab VII, Bab VIII dan Bab XII seperti berikut:

- (a) Analisa – contoh implementasi teknik pemodelan BDI-MU ke atas analisa sistem pengurusan inventori teragih dibentangkan secara terperinci dalam Bab VI.

- (b) Reka bentuk – contoh pembangunan seni bina sistem pengurusan inventori teragih dan implementasi model pandangan “4+1” dibentangkan secara terperinci dalam Bab VII dan Bab VIII.
- (c) Implementasi – contoh implementasi BDI-MU kit dan Java ke atas pembangunan komponen agen dan sistem pengurusan inventori teragih dibentangkan secara terperinci dalam Bab XII.
- (d) Integrasi – contoh penggabungan komponen agen dan komponen bukan agen bagi sistem pengurusan inventori teragih dibentangkan secara terperinci dalam Bab XII.
- (e) Penempatan – contoh versi akhir sistem pengurusan inventori teragih yang berasaskan agen mobil pintar dibentangkan secara terperinci dalam Bab XII.

## 5.5 Ringkasan

Metodologi BDI-MU yang merupakan metodologi ringan serempak berorientasikan agen mobil pintar yang diketengahkan bukan sahaja dikhususkan kepada pembangunan agen mobil pintar tetapi boleh digunakan untuk sistem pelbagai agen. Metodologi ini menyediakan panduan peraturan dan amalan kepada pembangun sistem agen pada sepanjang proses pembangunan bermula daripada mengumpul keperluan sehingga menempatkan sistem agen untuk kegunaan praktikal. Ia turut menyerapkan beberapa model seperti model  $t^*$  untuk kejuruteraan keperluan, model domain untuk memodelkan domain agen, model intra dan inter agen untuk memodelkan agen serta model pandangan “4+1” untuk menghuraikan seni bina agen. Metodologi ini telah diimplementasi ke atas pembangunan sistem pengurusan inventori teragih untuk membuktikan keupayaannya menyokong pembangunan aplikasi berasaskan agen mobil pintar yang beroperasi dalam persekitaran pengkomputeran heterogen dan teragih.

## **BAB VI**

### **TEKNIK PEMODELAN BDI-MU: TEKNIK PEMODELAN AGEN MOBIL PINTAR MENGGUNAKAN BAHASA PEMODELAN TERGABUNG**

#### **6.1 Pengenalan**

Penggunaan agen mobil pintar berkembang selari dengan peningkatan teknologi internet dan aplikasi teragih. Agen mobil pintar dibenam dalam sesuatu perisian kerana fungsinya diperlukan untuk menyokong operasi perisian ke atas domain yang semakin kompleks dan berskala besar. Mengurus kompleksiti dan risiko merupakan tugas yang mencabar dalam pembangunan sistem agen mobil pintar. Kaedah efektif untuk membangunkan sistem agen mobil pintar yang boleh dipercayai, mudah disenggara, mudah disambung dan menuruti spesifikasi adalah melalui teknik pemodelan. Ini adalah kerana teknik pemodelan menyediakan peringkasan, penstrukturan, perwarisan, modulariti dan lain-lain mekanisme untuk mengurus kompleksiti dan risiko semulajadi dalam membangunkan sistem agen mobil pintar.

Objektif bab ini adalah untuk mengetengahkan teknik pemodelan BDI-MU berasaskan UML yang dibangunkan untuk memodelkan domain agen mobil pintar dan struktur intra serta inter agen mobil pintar. Penggunaan UML bagi tujuan pemodelan turut diaplikasikan oleh FIPA (Odell et al., 2001), AUML (Bauer et al., 2001) dan MAS-CommonKADS (Iglesias et al., 1997). Sistem pengurusan inventori teragih dimodelkan menggunakan teknik pemodelan ini untuk menggambarkan dan menghuraikan sistem tersebut secara lebih terperinci.

## 6.2 Bahasa Pemodelan Tergabung

UML (Booch et al., 1999) adalah bahasa pemodelan dengan notasi yang digunakan untuk menghuraikan, memvisualisasikan, membangunkan dan mendokumentasikan analisis dan reka bentuk sistem. UML telah diperkenalkan oleh Kumpulan Pengurusan Objek (*Object Management Group*, OMG) hasil gabungan kejuruteraan perisian berasaskan objek (*object oriented software engineering*, OOSE) oleh Jacobson (1993), kaedah Booch oleh Booch (1994) dan teknik pemodelan objek (*object modeling technique*, OMT) oleh Rumbaugh (1995).

UML menyokong analisa dan reka bentuk berorientasikan objek. Di antara jenis model yang terdapat dalam UML adalah seperti berikut:

- (a) Model statik – menerangkan struktur statik data dan mesej di dalam sistem. Model statik digambarkan oleh gambarajah kelas. Gambarajah kelas boleh digunakan untuk dua tujuan. Tujuan pertama adalah untuk memodelkan masalah domain secara konseptual yang boleh difahami oleh pelanggan. Tujuan kedua adalah sebagai reka bentuk terperinci untuk memandu pembangun sistem menterjemah komponen sistem kepada kod-kod atur cara yang boleh dilarikan.
- (b) Model dinamik – menggambarkan kelakuan sistem yang bersandaran dengan masa. Model dinamik dibentuk oleh gambarajah interaksi (gambarajah jujukan dan gambarajah kerjasama), gambarajah keadaan dan gambarajah aktiviti.
- (c) Model kes gunaan – menerangkan keperluan sistem dan jujukan tindakan yang akan dilaksanakan oleh sistem termasuk interaksi sistem dengan pelaku. Gambarajah kes gunaan digunakan untuk menggambarkan model kes gunaan.
- (d) Model implementasi – melibatkan penghasilan gambarajah komponen untuk menerangkan komponen-komponen sistem seperti fail boleh laku, kod sumber dan sebagainya. Selain pembangunan gambarajah penempatan untuk menerangkan pengagihan komponen dalam persekitaran.
- (e) Bahasa kekangan objek (*object constraint language*, OCL) – bahasa formal untuk mentakrifkan kekangan model, tidak varian, prasyarat



dan pascasyarat operasi serta laluan pelayaran di antara rangkaian objek.

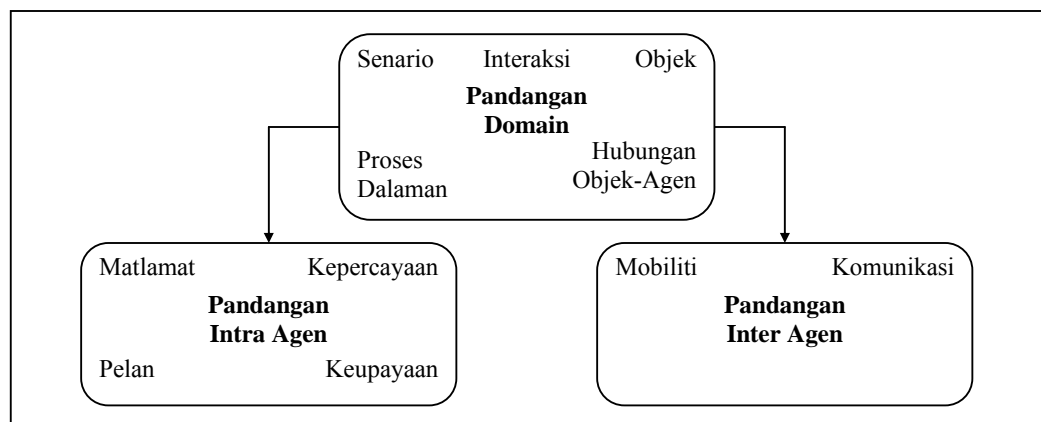
### 6.3 Teknik Pemodelan BDI-MU

Dalam melaksanakan aktiviti pemodelan ke atas domain dunia nyata, agen yang wujud di dalam domain tersebut diandaikan mempunyai ciri-ciri berikut:

- (a) Agen dan objek boleh wujud bersama dan mempunyai hubungan di antara satu sama lain.
- (b) Objek yang aktif boleh dianggap sebagai agen (Jennings et al., 1998).
- (c) Agen bertindak secara serentak dan tidak segerak.
- (d) Agen berinteraksi melalui pertukaran mesej.

Berdasarkan anggapan di atas, teknik pemodelan BDI-MU diperkenalkan. Teknik pemodelan ini menghuraikan sistem agen dari tiga sudut pandangan atau perspektif yang berbeza seperti pada Rajah 6.1, iaitu:

- (a) Pandangan domain – menerangkan aspek statik dan dinamik sistem agen serta hubungan di antara objek dengan agen.
- (b) Pandangan intra agen – menerangkan matlamat, kepercayaan, pelan dan keupayaan agen.
- (c) Pandangan inter agen – menerangkan mobiliti dan komunikasi agen.



Rajah 6.1 : Teknik pemodelan BDI-MU

### 6.3.1 Pandangan Domain

Pandangan domain melibatkan dua proses utama iaitu pemodelan dan pemahaman masalah domain serta identifikasi dan penentuan agen. Pemodelan dan pemahaman masalah domain membabitkan pembangunan artifak seperti berikut:

- (a) Gambarajah kesgunaan – menggambarkan satu set kesgunaan, pelaku dan hubungan di antaranya. Ianya membolehkan keperluan sistem dikenal pasti.
- (b) Gambarajah jujukan – menggambarkan senario dan interaksi di antara objek yang bersandaran dengan masa. Ianya membolehkan peranan kelas dan antara muka dikenal pasti.
- (c) Gambarajah kelas – menggambarkan satu set kelas, antara muka, kerjasama dan hubungan di antaranya. Ianya membolehkan atribut dan operasi objek dikenal pasti.
- (d) Gambarajah aktiviti – menggambarkan aliran kerja bagi sesuatu proses perniagaan. Ianya membolehkan proses dalaman dikenal pasti.

Identifikasi dan penentuan agen pula melibatkan aktiviti mengecam agen daripada satu set objek menggunakan peraturan pemilihan agen seperti pada Jadual 6.1. Peraturan meta agen (Nwana, 1996) pula seperti pada Jadual 6.2 digunakan untuk mengelaskan agen.

**Jadual 6.1 : Peraturan pemilihan agen**

1. Autonomi.
1.1. Adakah ianya memerlukan pengetahuan dalaman?
1.2. Adakah ianya membuat keputusan sendiri?
1.3. Bolehkah ianya bertoleransi dengan input yang tidak benar atau tidak diduga?
2. Penyesuaian.
2.1. Adakah ianya memerlukan pengetahuan dalaman?
2.2. Adakah pengetahuannya dikemaskini secara berterusan?
2.3. Adakah ianya berinteraksi dengan entiti luaran?
3. Kerjasama.
3.1. Adakah ianya berinteraksi dengan entiti luaran?
3.1.1. Adakah ianya beroperasi secara bekerjasama?
3.1.2. Adakah ianya beroperasi secara pelbagai beban?
4. Mobiliti.
4.1. Adakah ianya berupaya mengubah kedudukan fizikal?

**Jadual 6.2 : Peraturan meta agen**

1. Sekiranya agen mempunyai autonomi, penyesuaian dan kerjasama, ianya dikelaskan sebagai agen pintar.
2. Sekiranya agen mempunyai penyesuaian dan kerjasama, ianya dikelaskan sebagai agen pembelajaran kerjasama.
3. Sekiranya agen mempunyai autonomi dan kerjasama, ianya dikelaskan sebagai agen kerjasama.
4. Sekiranya agen mempunyai autonomi dan penyesuaian, ianya dikelaskan sebagai agen antara muka.
5. Sekiranya agen mempunyai mobiliti, ianya dikelaskan sebagai agen mobil.
6. Sekiranya agen mempunyai autonomi, penyesuaian, kerjasama dan mobiliti, ianya dikelaskan sebagai agen mobil pintar.

Agen-agen yang telah diterbitkan dan objek-objek yang masih tinggal di dalam gambarajah kelas kemudian dicantumkan dalam gambarajah kelas agen. Gambarajah ini menerangkan hubungan di antara agen termasuk objek bukan agen yang membentuk sesuatu sistem agen. Gambarajah kelas agen juga menggambarkan struktur dalaman agen yang merangkumi atribut matlamat, kepercayaan, pelan dan keupayaan.

### 6.3.2 Pandangan Intra Agen

Pandangan intra agen menerangkan sifat-sifat dalaman agen berasaskan seni bina BDI. Sifat-sifat dalaman agen terdiri daripada empat komponen utama, iaitu:

- (a) Matlamat – merupakan objektif muktamad yang perlu dicapai oleh agen. Dardenne et al. (1993) menggambarkan matlamat dalam bentuk Corak\_Matlamat (*Pattern\_of\_Goal*) yang dikategorikan kepada lima kumpulan iaitu Capai, Henti, Pelihara, Hindar dan Optima. Gambarajah hirarki matlamat digunakan untuk memodelkan matlamat agen.
- (b) Kepercayaan – merupakan maklumat yang dimiliki oleh agen mengenai persekitaran atau agen itu sendiri. Maklumat tersebut digunakan untuk membentuk pangkalan pengetahuan atau ontologi agen. Gambarajah ontologi digunakan untuk memodelkan kepercayaan agen menggunakan Protégé (Noy et al., 2001).

- (c) Pelan – mempamerkan kelakuan agen untuk mencapai matlamat. Ia mengfokus kepada kelakuan agen yang berubah bersandarkan masa. Gambarajah jujukan pelan digunakan untuk memodelkan pelan agen.
- (d) Keupayaan – satu set operasi yang boleh dilaksanakan oleh agen. Ianya turut menjelaskan proses dalaman bagaimana input ditukarkan kepada output. Gambarajah aliran data keupayaan digunakan untuk memodelkan keupayaan agen.

### **6.3.3 Pandangan Inter Agen**

Pandangan inter agen memperlihatkan pergerakan agen di antara hos dan pertukaran mesej di antara agen dalam sistem teragih dan sistem pelbagai agen. Gambarajah jujukan migrasi digunakan untuk memodelkan pergerakan agen. Ianya menggambarkan bagaimana agen bergerak di antara hos melalui rangkaian untuk melaksanakan tugas yang spesifik. Gambarajah jujukan komunikasi pula digunakan untuk memodelkan komunikasi agen dan menjelaskan bagaimana mesej dihantar kepada agen atau objek bukan agen.

## **6.4 Implementasi Teknik Pemodelan BDI-MU dalam Pemodelan Sistem Pengurusan Inventori Teragih**

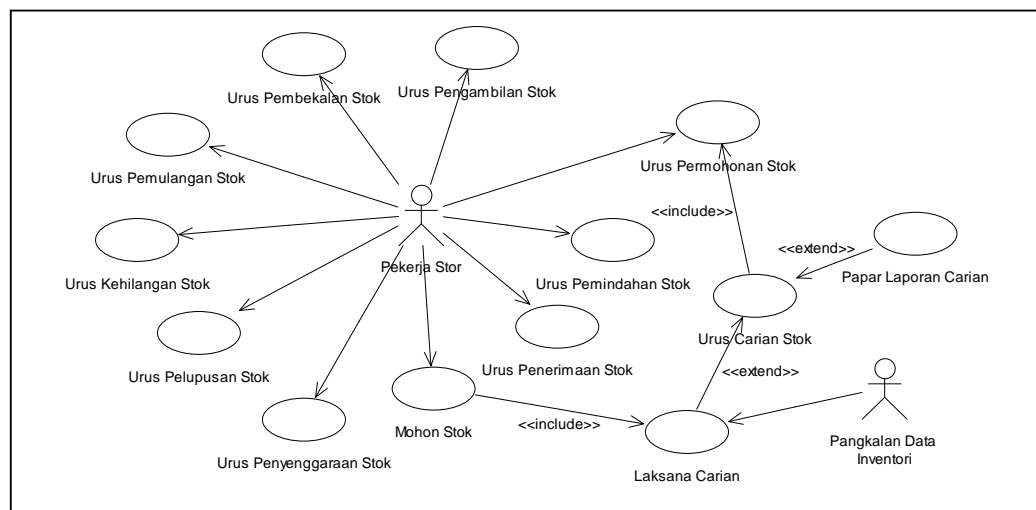
### **6.4.1 Model Domain**

Pandangan domain menggambarkan bagaimana domain yang menjadi penempatan agen dianalisa dan bagaimana agen dikenal pasti. Proses ini dapat ditunjukkan oleh model domain. Hasil penganalisan domain agen dibentangkan oleh gambarajah kes gunaan, gambarajah jujukan, gambarajah kelas dan gambarajah aktiviti. Kemudian, peraturan pemilihan agen dan peraturan meta agen digunakan untuk mengecam agen-agen relevan yang wujud di dalam domain. Hasil pengecaman pula dibentangkan oleh gambarajah kelas agen.

### 6.4.1.1 Gambarajah Kes Gunaan

Gambarajah kes gunaan adalah seperti pada Rajah 6.2. Pekerja Stor dan Pangkalan Data Inventori adalah dua pelaku utama yang wujud di dalam domain. Pekerja Stor bertanggungjawab menginput maklumat kepada sistem dan mengurus stor. Pangkalan Data Inventori pula bertanggungjawab menyimpan maklumat stok. Kes gunaan yang terlibat di dalam proses penjejakan inventori teragih adalah:

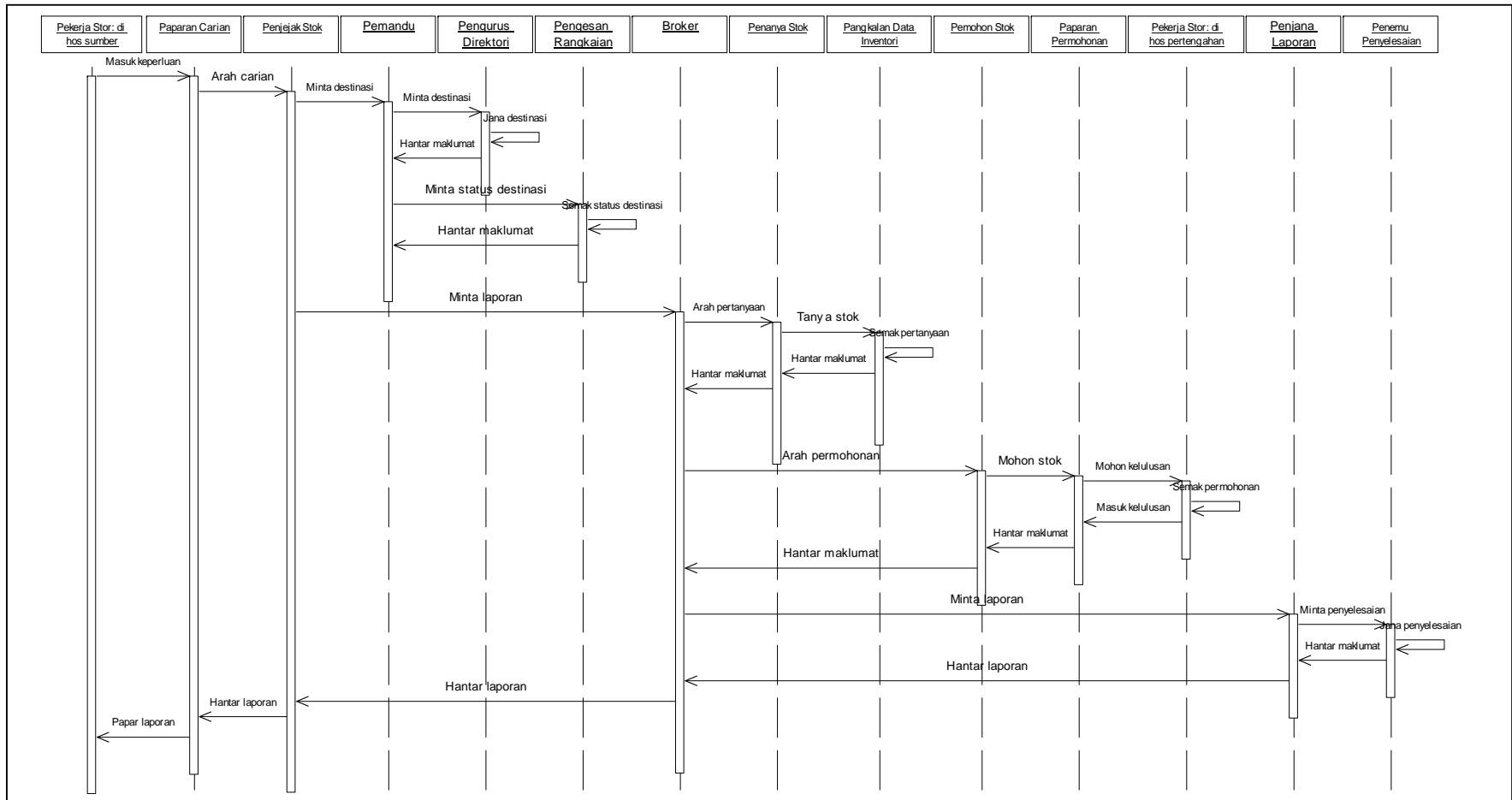
- (a) Kes gunaan Urus Carian Stok – mengurus komunikasi untuk mendapatkan input daripada pekerja stor.
- (b) Kes gunaan Laksana Carian – mendapatkan maklumat stok daripada pangkalan data inventori yang teragih.
- (c) Kes gunaan Mohon Stok – melaksanakan proses permohonan stok.
- (d) Kes gunaan Papar Laporan Carian – menyediakan laporan carian dan bekalan sokongan daripada stor-stor luaran.



**Rajah 6.2 : Gambarajah kes gunaan sistem pengurusan inventori teragih**

### 6.4.1.2 Gambarajah Jujukan

Sebanyak tiga belas kelas dikenal pasti daripada gambarajah jujukan seperti yang dipaparkan pada Rajah 6.3 untuk kes gunaan Urus Carian Stok, Laksana Carian, Mohon Stok dan Papar Laporan Carian. Kelas-kelas tersebut ialah kelas

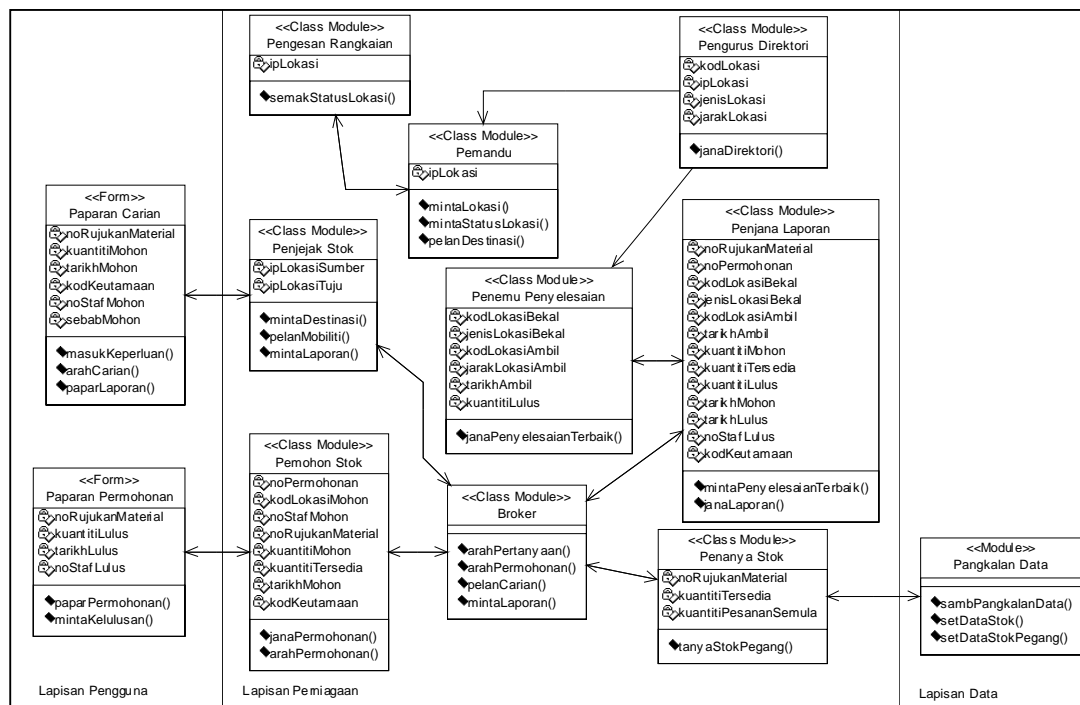


Rajah 6.3 : Gambarajah jujukan sistem pengurusan inventori teragih

Paparan Carian, Paparan Permohonan, Penjejak Stok, Pemandu, Broker, Pengurus Direktori, Pengesan Rangkaian, Penanya Stok, Pemohon Stok, Penjana Laporan, Penemu Penyelesaian, Pekerja Stor dan Pangkalan Data Inventori. Setiap kelas menghantar mesej kepada kelas yang lain. Sebagaimana kelas Pemandu menerima mesej MintaDestinasi() daripada kelas Penjejak Stok. Kelas Penjejak Stok kemudian melaksanakan beberapa proses bagi memenuhi permintaan tersebut dan menghantar hasilnya kepada kelas Penjejak Stok.

### 6.4.1.3 Gambarajah Kelas

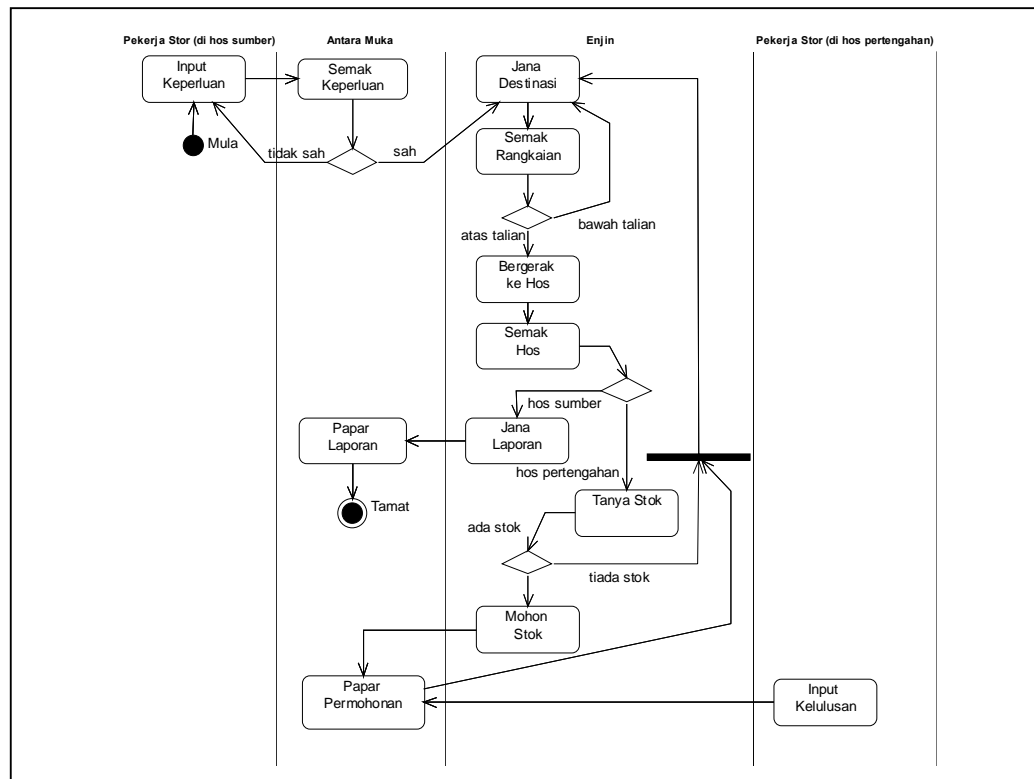
Gambarajah kelas dijana dengan menyaring objek-objek yang diidentifikasi daripada gambarajah kesgunaan dan gambarajah jujukan. Aspek statik sesuatu objek diwakili oleh atribut dan operasi bagi objek tersebut. Rajah 6.4 menunjukkan gambarajah kelas bagi proses penjejakan inventori teragih. Rangka kerja tiga lapisan digunakan untuk menghimpunkan kelas-kelas kepada lapisan pengguna (antara muka pengguna), lapisan perniagaan (objek perniagaan) dan lapisan data (pangkalan data).



Rajah 6.4 : Gambarajah kelas sistem pengurusan inventori teragih

#### 6.4.1.4 Gambarajah Aktiviti

Gambarajah aktiviti pada Rajah 6.5 mempamerkan aliran kerja untuk proses menjejaki stok. Aktiviti yang dilaksanakan oleh agen mobil pintar membolehkan ia menjana penyelesaian terbaik untuk membantu pengguna membuat keputusan, memanipulasi maklumat yang diterima dan memindahkan stok.



**Rajah 6.5 : Gambarajah aktiviti sistem pengurusan inventori teragih**

#### 6.4.1.5 Peraturan Pemilihan Agen dan Peraturan Meta Agen

Melalui penggunaan peraturan pemilihan agen dan peraturan meta agen, beberapa agen telah dikenal pasti untuk melaksanakan proses penjejakan inventori teragih. Agen-agen tersebut ialah agen Pengurus Domain, agen Pengurus Pelayan, agen Antara Muka, agen Penjejak Stok, agen Pemandu, agen Broker, agen Pengurus Direktori, agen Pengesan Rangkaian, agen Penanya Stok, agen Penjana Laporan dan



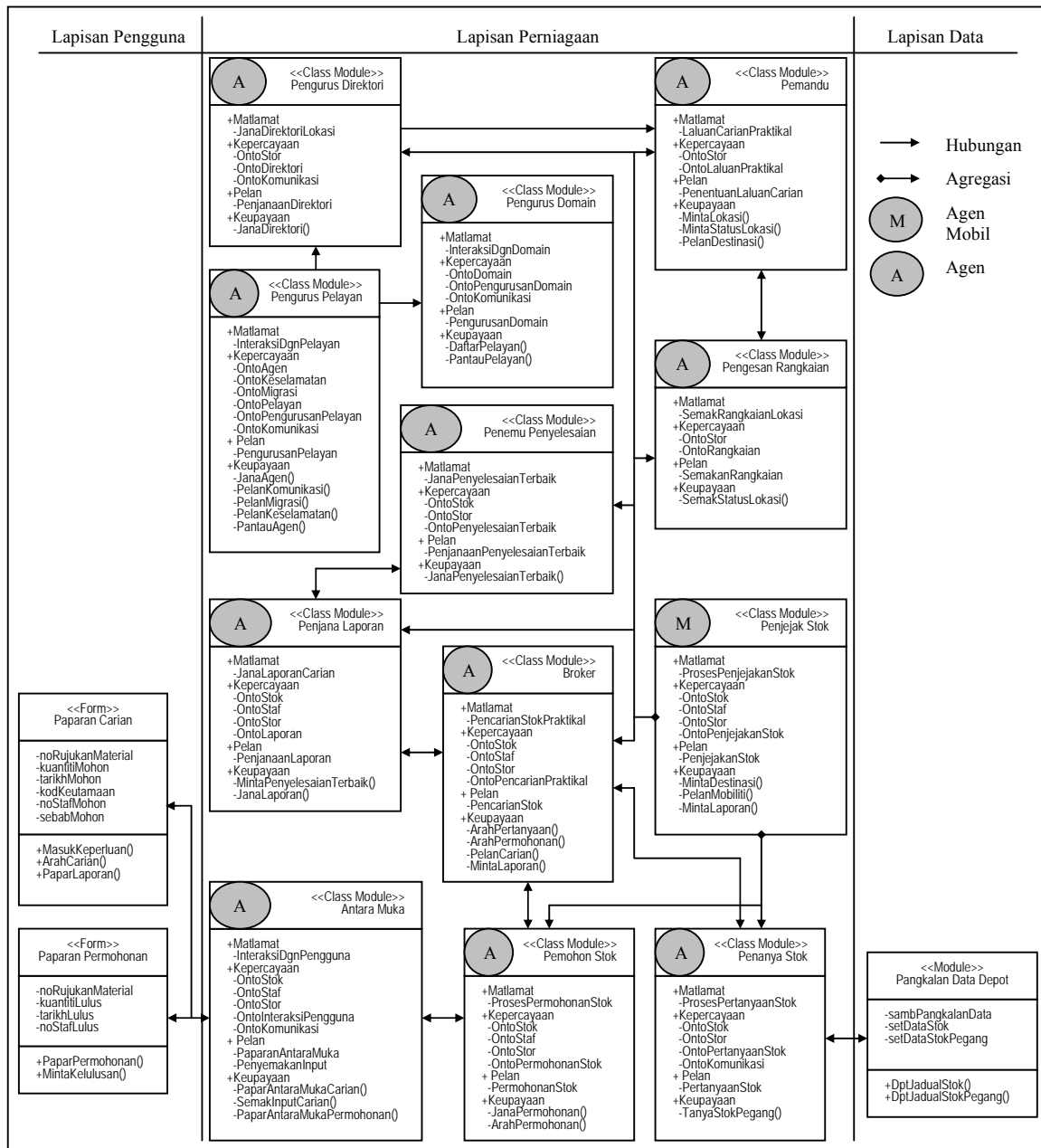
agen Penemu Penyelesaian. Jadual 6.3 adalah penerangan sebahagian daripada agen berkenaan.

**Jadual 6.3 : Contoh agen yang dikenal pasti menggunakan peraturan pemilihan agen**

1.	Agen Antara Muka. Berkomunikasi dengan pengguna untuk mendapatkan keperluan, memahami keinginan dan membekalkan maklumat kepada pengguna. Ianya merupakan agen antara muka dengan ciri autonomi dan penyesuaian (peraturan 1.1, 1.3 dan 2.3 digunakan).
2.	Agen Pemandu. Melalui maklumat yang dibekalkan oleh agen Pengurus Direktori dan agen Pengesan Rangkaian, agen Pemandu membuat keputusan menentukan destinasi laluan. Ianya merupakan agen pintar dengan ciri autonomi, penyesuaian dan kerjasama (peraturan 1.2, 1.3, 2.3 dan 3.1.1 digunakan).
3.	Agen Pemohon Stok. Melalui maklumat yang dibekalkan oleh agen Penanya Stok dan agen Broker, agen Pemohon Stok mendapatkan kelulusan bekalan daripada pekerja stor. Ianya merupakan agen pintar dengan ciri autonomi, penyesuaian dan kerjasama (peraturan 1.2, 1.3, 2.3 dan 3.1.1 digunakan).
4.	Agen Penemu Penyelesaian. Melalui maklumat yang dibekalkan oleh agen Penjana Laporan dan agen Pengurus Direktori, agen Penemu Penyelesaian menjana cadangan penyelesaian bekalan untuk membantu pekerja stor melaksanakan proses pemindahan stok. Ianya merupakan agen pintar dengan ciri autonomi, penyesuaian dan kerjasama (peraturan 1.2, 1.3, 2.3 dan 3.1.1 digunakan).
5.	Agen Penjejak Stok. Agen ini bergerak dari sebuah pelayan ke pelayan yang berikutnya untuk melaksanakan proses penjejakan stok. Ianya beroperasi secara pelbagai bebenang dan struktur dalamannya merupakan satu set agen-agen pintar yang saling bekerjasama. Ianya merupakan agen mobil pintar dengan ciri autonomi, penyesuaian, kerjasama dan mobiliti (peraturan 1.2, 2.3, 3.1.1, 3.1.2 dan 4.1 digunakan).

#### 6.4.1.6 Gambarajah Kelas Agen

Gambarajah kelas agen memaparkan interaksi dan hubungan di antara agen serta apakah kelas bukan agen yang digunakan oleh agen-agen tersebut untuk melaksanakan tugasnya. Rajah 6.6 menunjukkan gambarajah kelas agen. Setiap agen mempunyai tugas yang spesifik dengan matlamat, kepercayaan, pelan dan keupayaan untuk menyokong pelaksanaan tugas yang telah didelegasikan kepadanya.



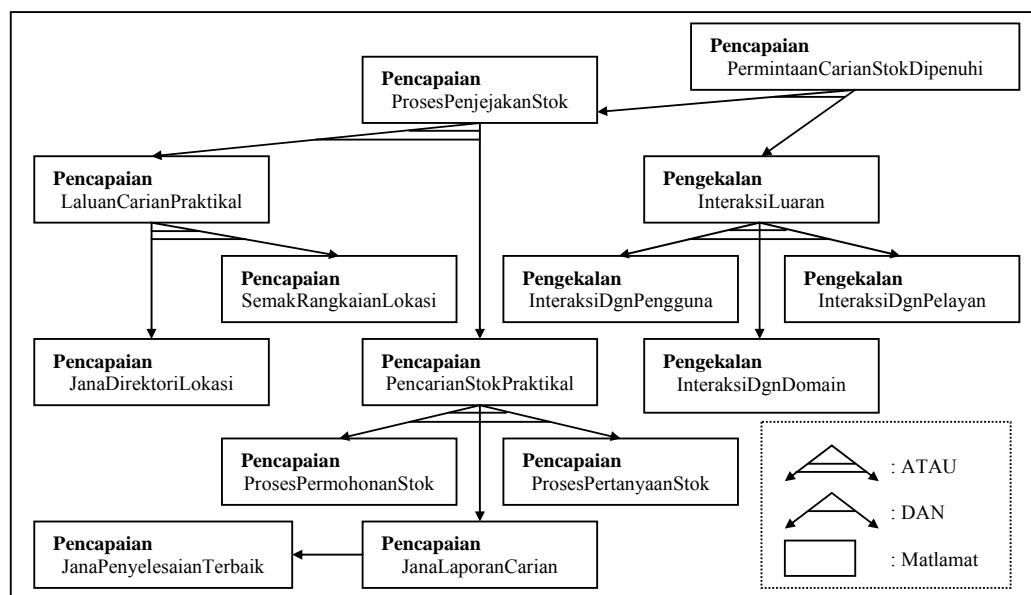
Rajah 6.6 : Gambarajah kelas agen sistem pengurusan inventori teragih

### 6.4.2 Model Intra Agen

Pandangan intra agen digambarkan oleh model intra agen. Proses pemodelan intra agen melibatkan penganalisaan dan perwakilan matlamat, kepercayaan, pelan dan keupayaan agen.

### 6.4.2.1 Gambarajah Hirarki Matlamat

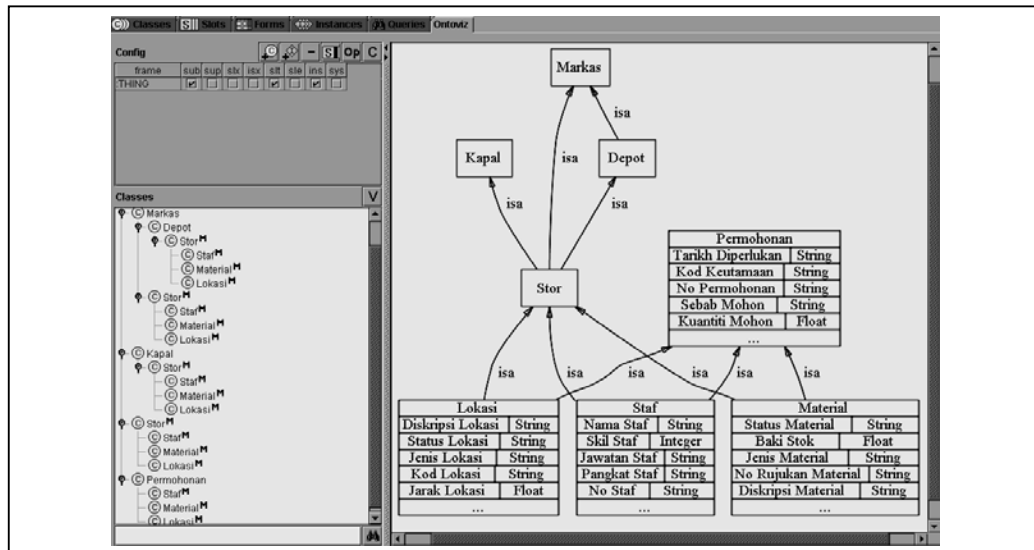
Setiap agen mempunyai matlamat tersendiri dan melaksanakan satu atau lebih tugas untuk mencapai matlamatnya. Rajah 6.7 menunjukkan gambarajah hirarki matlamat yang menjelaskan hubungan di antara matlamat dan submatlamat. Contohnya, matlamat agen Pemandu ialah mencapai LaluanCarianPraktikal. Submatlamat agen Pemandu pula ialah JanaDirektoriLokasi dan SemakRangkaianLokasi yang masing-masing merupakan matlamat kepada agen Pengurus Direktori dan agen Pengesan Rangkaian.



Rajah 6.7 : Gambarajah hirarki matlamat sistem pengurusan inventori teragih

### 6.4.2.2 Gambarajah Ontologi

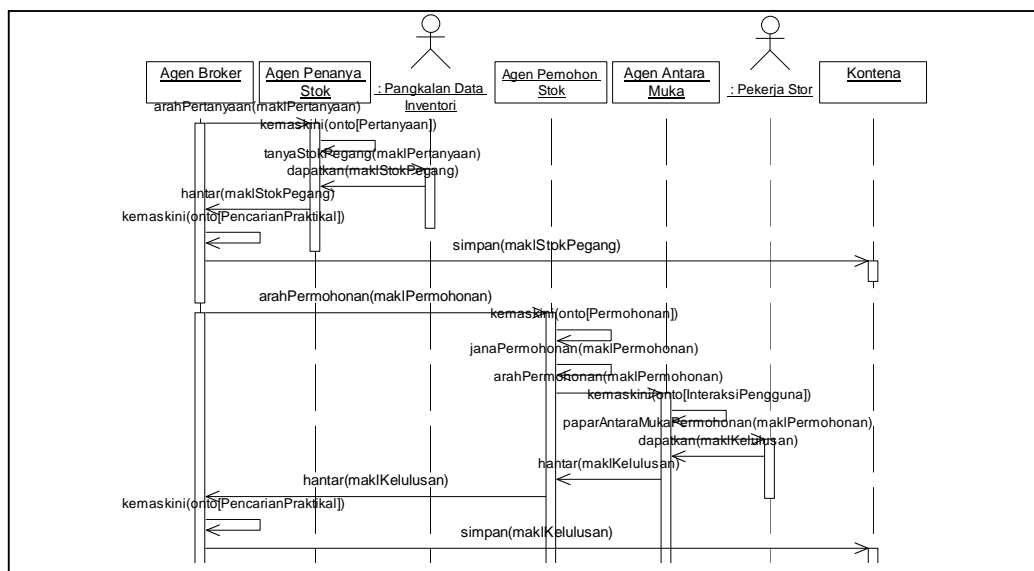
Kepercayaan agen adalah fakta atau pengetahuan yang dipegang oleh agen bagi mewakili keadaan semasa dunia nyata. Maklumat yang disimpan termasuklah keadaan pembolehubah, maklumat penderia, kesimpulan daripada inferens atau tolakan, model maklumat mengenai agen lain atau sebagainya. Ontologi (Gruber, 1993) digunakan sebagai pendekatan untuk memodelkan kepercayaan agen. Gambarajah ontologi adalah seperti pada Rajah 6.8.



Rajah 6.8 : Gambarajah ontologi sistem pengurusan inventori teragih

### 6.4.2.3 Gambarajah Jujukan Pelan

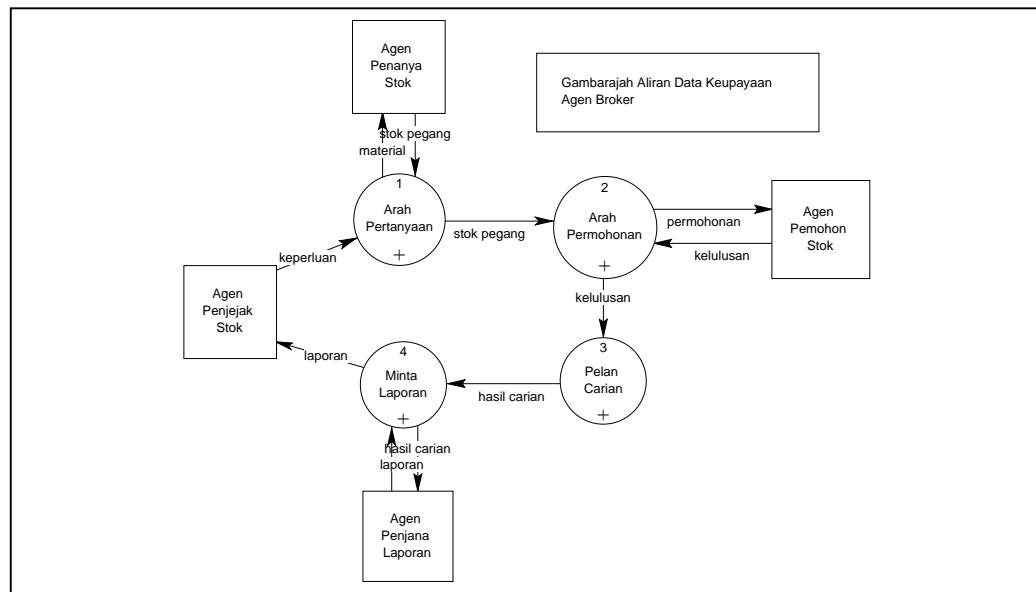
Pelan menghuraikan prosidur untuk mencapai sesuatu matlamat. Rajah 6.9 menunjukkan gambarajah jujukan pelan bagi agen Broker mencapai matlamat dan submatlamat PencarianStokPraktikal, ProsesPertanyaanStok dan ProsesPermohonanStok.



Rajah 6.9 : Gambarajah jujukan pelan sistem pengurusan inventori teragih

#### 6.4.2.4 Gambarajah Aliran Data Keupayaan

Keupayaan agen dimodelkan menggunakan gambarajah aliran data seperti pada Rajah 6.10. Agen Broker mempunyai empat keupayaan aras konteks iaitu ArahPertanyaan(), ArahPermohonan(), PelanCarian() dan MintaLaporan(). Keupayaan aras konteks pula boleh digerudi untuk membentuk beberapa subkeupayaan yang bergantung kepada kedalaman atau aras tertentu.



**Rajah 6.10 : Gambarajah aliran data keupayaan sistem pengurusan inventori teragih**

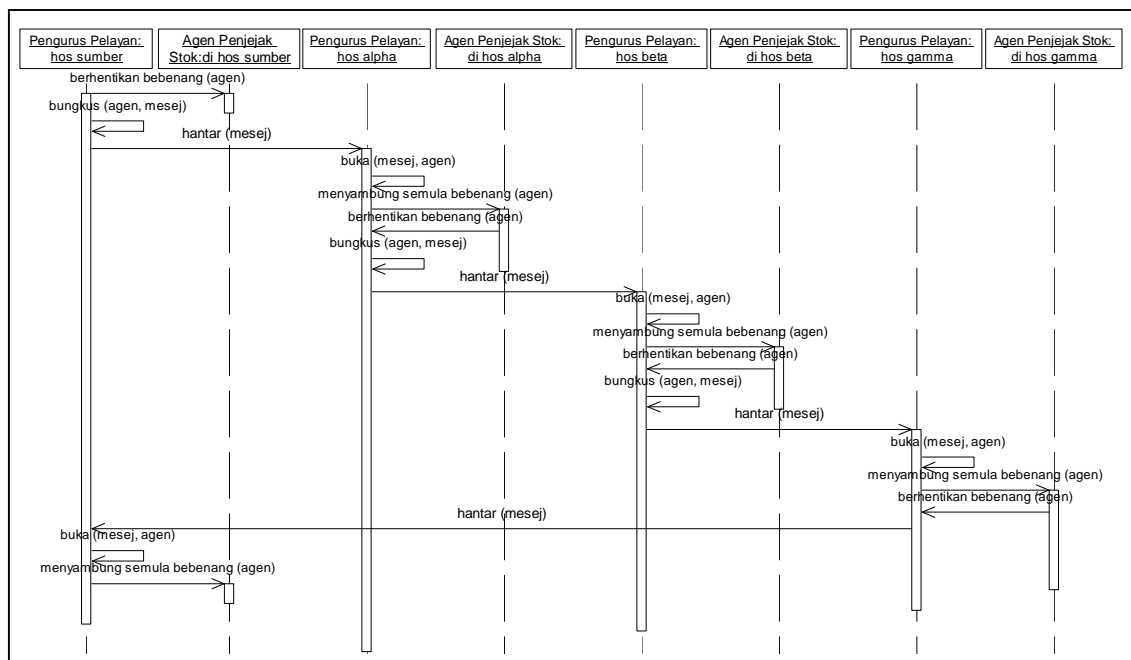
#### 6.4.3 Model Inter Agen

Model inter agen digunakan untuk mewakili pandangan inter agen. Dua proses utama di dalam model inter agen ialah pemodelan mobiliti agen dan pemodelan komunikasi agen. Pemodelan mobiliti agen bertujuan menerangkan strategi bagaimana agen mengubah kedudukan fizikalnya dalam satu set hos yang saling terangkai. Pemodelan komunikasi agen pula bertujuan menerangkan aspek dinamik pertukaran mesej di antara agen dalam hos yang sama atau hos yang berlainan.

### 6.4.3.1 Gambarajah Jujukan Migrasi

Gambarajah jujukan migrasi seperti pada Rajah 6.11 menjelaskan bagaimana agen dipindahkan daripada satu hos ke hos yang lain dalam persekitaran yang teragih. Rentetan mobiliti agen adalah seperti berikut:

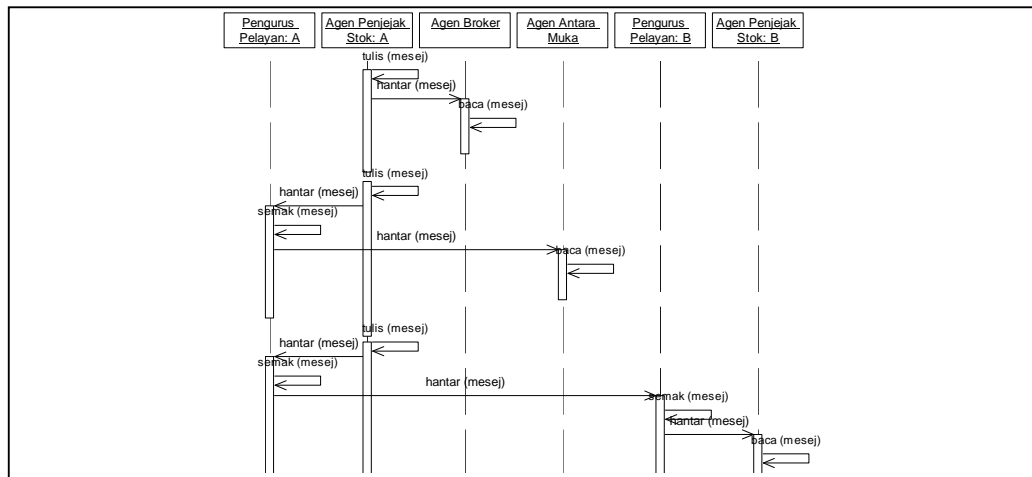
- (a) Pengurus pelayan hos penghantar memanggil keupayaan `SemasaBertolak()` agen dan memberhentikan bebenang agen.
- (b) Pengurus pelayan hos penghantar menghantar mesej yang merupakan keadaan dalaman agen kepada pengurus pelayan hos penerima.
- (c) Pengurus pelayan hos penerima menyemak sama ada ia mempunyai kebenaran mencapai kod bait kelas agen. Sekiranya tidak, mesej meminta kod bait kelas dihantar kepada pengurus pelayan hos penghantar.
- (d) Apabila pengurus pelayan hos penghantar menerima mesej meminta kod bait kelas, ia akan menghantar kod bait kelas yang diminta kepada pengurus pelayan hos penerima.
- (e) Pengurus pelayan hos penerima akan memulakan kelas tersebut dan memanggil keupayaan `SemasaSampai()` agen dan menyambung semula bebenang agen.



Rajah 6.11 : Gambarajah jujukan migrasi sistem pengurusan inventori teragih

### 6.4.3.2 Gambarajah Jujukan Komunikasi

Komunikasi agen dibahagikan kepada tiga kategori iaitu komunikasi dalam agen yang sama, komunikasi dalam hos yang sama dan komunikasi dalam hos yang berlainan. Hanya komunikasi di antara agen dalam agen yang sama sahaja tidak melibatkan Pengurus Pelayan. Pengurus Pelayan berperanan menyemak mesej untuk dihantar kepada agen yang sepatut menerimanya. Komunikasi agen dipamerkan oleh gambarajah jujukan komunikasi seperti pada Rajah 6.12.



**Rajah 6.12 : Gambarajah jujukan komunikasi sistem pengurusan inventori teragih**

## 6.5 Ringkasan

Di dalam bab ini, teknik pemodelan BDI-MU yang berasaskan UML telah diketengahkan dan digunakan ke atas sistem pengurusan inventori teragih. Teknik pemodelan ini melihat agen mobil pintar pada tiga sudut pandangan yang berbeza. Pertamanya dari sudut pandangan domain untuk melihat hubungan agen mobil pintar dengan objek bukan agen. Keduanya dari sudut pandangan intra agen untuk melihat struktur dalaman agen mobil pintar. Akhir sekali dari sudut pandangan inter agen untuk melihat interaksi di antara agen mobil pintar dan pergerakan agen mobil pintar untuk meningkatkan kemampuannya menyelesaikan masalah.

## **BAB VII**

### **RANGKA KERJA ORGANISASI HOLONIK DAN HIRARKI AMBIENTS UNTUK PEMBANGUNAN SENI BINA AGEN MOBIL PINTAR**

#### **7.1 Pengenalan**

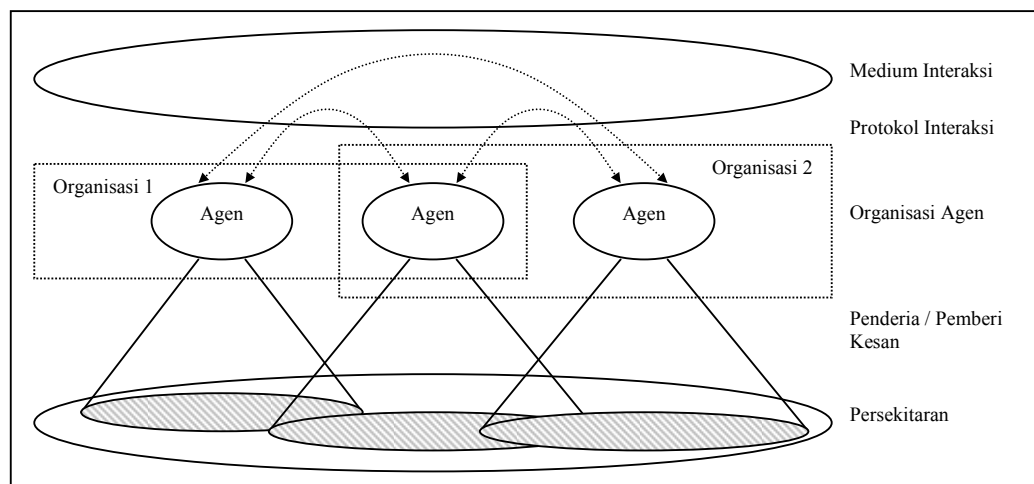
Inovasi seni bina agen sentiasa diperlukan dan dibangunkan untuk menyokong sistem agen. Ianya sejajar dengan implementasi sistem agen dalam persekitaran yang berskala besar, heterogen dan teragih. Tujuan seni bina agen adalah untuk menghasilkan sistem agen yang tegap, boleh dipercayai dan secocok dengan matlamat yang dikehendaki. Sebahagian contoh seni bina agen yang telah dibangunkan adalah seperti MARS (Cabri et al., 2000) untuk aplikasi internet, MACRON (Decker dan Sycara, 1997) untuk himpunan maklumat dan KAoS (Bradshaw et al., 1995) untuk aplikasi aeroangkasa. Oleh kerana itu, seni bina agen wajar dibangunkan berdasarkan prinsip-prinsip dan amalan kejuruteraan perisian. Bagaimanapun, majoriti seni bina agen dibangunkan secara ad hoc dan kurang menuruti asas-asas kejuruteraan perisian yang ketat.

Objektif bab ini adalah untuk mengetengahkan rangka kerja organisasi holonik dan hirarki ambients bagi pembangunan seni bina agen mobil pintar. Rangka kerja ini mengandungi tiga lapisan iaitu lapisan holon, lapisan objek dan lapisan kelas. Hirarki ambients pula mengimplementasi konsep dwihirarki dan kontena. Tujuannya adalah sebagai metodologi dan templat guna semula untuk memudahkan pembangunan seni bina agen mobil pintar. Di dalam bab ini juga ditunjukkan penggunaannya untuk membina seni bina sistem pengurusan inventori teragih.



## 7.2 Organisasi Agen

Organisasi agen ditakrifkan sebagai sistem yang dikomposisi oleh agen yang saling berinteraksi dan mempunyai hubungan di antara satu sama lain (Prietula dan Carley, 1994). Rajah 7.1 mempamerkan organisasi agen (Castelfranchi, 1998). Bagi memastikan peranan agen dalam organisasi dapat disempurnakan secara efektif, agen berinteraksi dengan protokol tertentu untuk bertukar pengetahuan dan bekerjasama dalam melaksanakan sesuatu aktiviti. Cabri dan Leonardi (2000) mengatakan bahawa konsep utama dalam organisasi agen adalah peraturan organisasi, struktur organisasi dan corak organisasi.

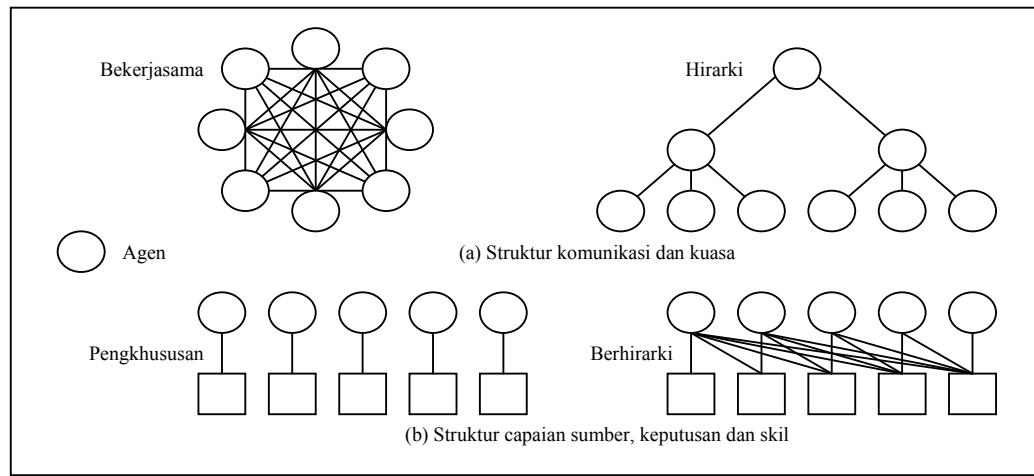


**Rajah 7.1 : Organisasi agen**

Peraturan organisasi menerangkan konvensyen sosial dan konvensyen organisasi. Konvensyen sosial mentakrifkan interaksi di antara ahli dalam organisasi (contohnya, kerani tidak boleh mengabaikan arahan pengurus). Konvensyen organisasi pula mentakrifkan bagaimana peranan yang berbeza memainkan tugas dalam organisasi (contohnya, pengurus tidak boleh menyerahkan tugas yang di luar bidang kerja kerani). Peraturan organisasi mengambil kira peranan individu dalam organisasi dan protokol interaksi di antara individu.

Struktur organisasi menerangkan topologi dan rejim kawalan aktiviti organisasi. Ia merupakan peringkasan seni bina sistem agen. Struktur organisasi menerangkan dan menentukan peranan dan tanggungjawab agen dalam komunikasi

dan kuasa serta capaian sumber, keputusan dan skil. Rajah 7.2 menggambarkan atribut struktur organisasi agen.



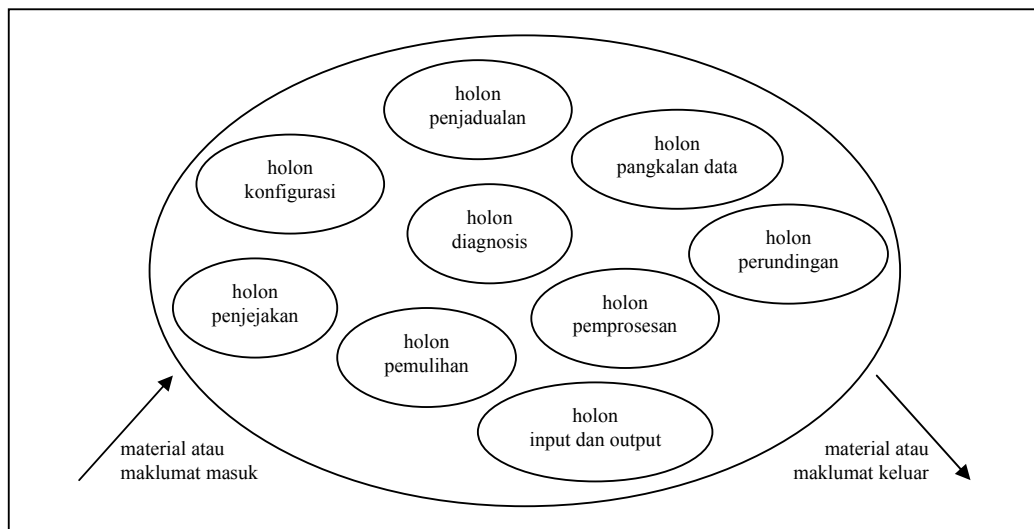
**Rajah 7.2 : Struktur organisasi agen**

Corak organisasi menerangkan strategi, struktur dan praktik organisasi. Corak berorientasikan objek dan struktur-dalam-5 (Do et al., 2002) adalah contoh corak organisasi. Corak berorientasikan objek mentakrifkan sekumpulan objek termasuk interaksi dan hubungan di antara objek. Corak struktur-dalam-5 pula mentakrifkan hubungan di antara *dependen-dependum-dependee*.

### 7.3 Holonik

Istilah holonik merupakan gabungan perkataan Greek iaitu *holos* yang bermaksud keseluruhan dan *on* yang bermaksud komponen atau bahagian. Koestler (1989) memperkenalkan holonik untuk menghuraikan susunan dan hirarki unit-unit asas organisasi dalam sistem sosial dan biologi. Koestler mendapati organisma hidup atau unit sosial dalam organisasi adalah saling bersandaran dan berinteraksi. Setiap unit sebagaimana sel-sel dalam mamalia mengandungi lebih dari satu plasma dan nukleus dengan hubungan ibu bapa dan adik-beradik. Sel-sel tersebut pada masa yang sama membentuk bahagian-bahagian organisasi yang lebih besar iaitu tisu-tisu otot.

Holon adalah bahagian sistem yang boleh dicam. Setiap holon mempunyai identiti yang unik. Gabungan holon akan membina bahagian-bahagian sistem yang lebih besar dan seterusnya membentuk sistem secara keseluruhan. Holonik telah digunakan dalam aplikasi pengeluaran seperti sistem pembuatan holonik (*holonic manufacturing system*, HMS) oleh Fischer (1999), aplikasi kawalan seperti sistem kawalan holonik (*holonic control system*, HCS) oleh McFarlane et al. (1995) dan robotik (Turgut et al., 1999). Rajah 7.2 menggambarkan struktur holon dalam HCS, di mana setiap holon mempunyai ciri-ciri berautonomi dan bekerjasama.



**Rajah 7.3 : Struktur holon dalam sistem kawalan holonik**

#### 7.4 Ambients

Ambients (Cardelli dan Gordon, 2000) merupakan entiti komputasi yang menyamai agen. Ambients ditakrifkan sebagai  $n[P]$ ,  $n$  dan  $P$  masing-masing merupakan nama ambients dan proses sedang jalan di dalam ambients. Proses terdiri daripada rencaman selari beberapa proses atau ambients:

$P, Q ::=$	proses
$0$	tidak aktif
$P \mid Q$	rencaman selari, untuk $P \rightarrow Q \Rightarrow P \mid R \rightarrow Q \mid R$
$n[P]$	ambients, untuk $P \rightarrow Q \Rightarrow n[P] \rightarrow n[Q]$

Setiap ambients dibenarkan bersarang di dalam ambients lain. Ianya membolehkan ambients di organisasi secara hirarki:

$$n[P_1 | \dots | P_p | m_1[\dots] | \dots | m_q[\dots]], \quad (P_i \neq n_i[\dots])$$

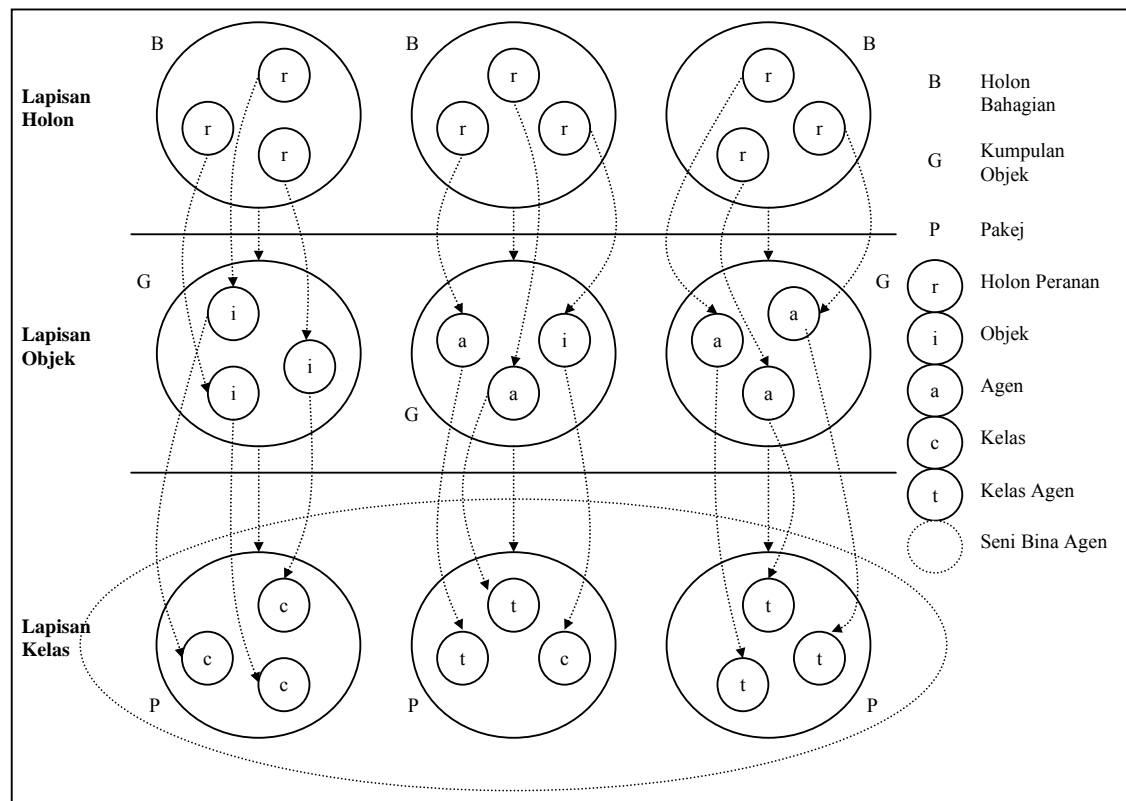
## 7.5 Rangka Kerja Organisasi Holonik

Ontologi, autonomi, rasional dan komputasi organisasi adalah konsep sosiologi yang telah digunakan ke dalam bidang agen. Kajian yang dijalankan oleh Gilbert dan Terna (2000), Prietula dan Carley (1999), Agre dan Horswill (1997) serta Parunak (1997) cuba menerokai fenomena sosiologi untuk meningkatkan potensi teknologi agen. Rangka kerja organisasi holonik diketengahkan untuk menangani masalah dalam mereka bentuk seni bina agen. Rangka kerja ini adalah berciri hibrid dengan menggabungkan konsep sosiologi iaitu organisasi dengan konsep biologi iaitu holonik. Secara ringkasnya, rangka kerja organisasi holonik melihat seni bina agen sebagai satu organisasi yang terbentuk di dalamnya holon-holon. Integrasi holon membentuk unit-unit yang lebih besar dan seterusnya membina sistem agen secara bersepadu.

Rangka kerja organisasi holonik mengandungi tiga lapisan bermula daripada reka bentuk konseptual organisasi holon, pemetaan holon kepada objek sehingga terjemahan objek kepada kelas. Holon boleh dipetakan kepada agen kerana terdapat holon yang mempunyai ciri setara dengan agen sebagai entiti berautonomi dengan aras ketidaksandaran tertentu dan saling bekerjasama serta berinteraksi dengan holon yang lain atau persekitaran (Bongaerts, 1998). Agen pula boleh diterjemahkan kepada kelas kerana ia adalah objek aktif yang mempunyai kawalan bebenang dan matlamat tersendiri (Odell dan Parunak, 2000). Ini adalah kerana agen mempunyai keupayaan “*go*” (boleh dilarikan tanpa dijana semula) dan “*no*” (boleh menentukan tindakan berdasarkan matlamat dan keadaan semasa). Lapisan-lapisan di dalam rangka kerja organisasi holonik adalah seperti pada Rajah 7.4, iaitu:

- (a) Lapisan holon – lapisan yang menerangkan bagaimana holon-holon membentuk satu pepaduan organisasi.
- (b) Lapisan objek – lapisan yang memetakan holon kepada objek.

(c) Lapisan kelas – lapisan yang menterjemahkan objek kepada kelas.

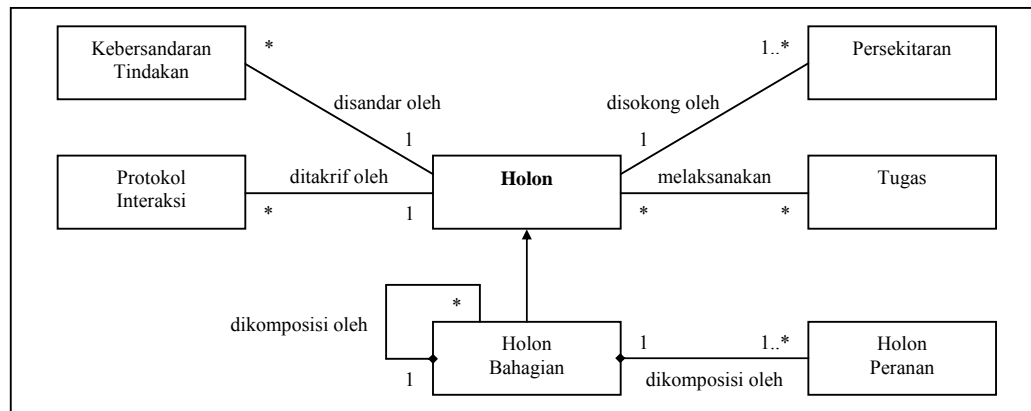


**Rajah 7.4 : Rangka kerja organisasi holonik**

### 7.5.1 Lapisan Holon

Gambaran keseluruhan lapisan holon adalah seperti pada Rajah 7.5. Lapisan holon melibatkan penyatuan dua holon utama, iaitu:

- Holon peranan – mewakili satu set peranan yang dipegang oleh bahagian (contohnya, kerani dan pengurus). Ianya turut mencakupi kuasa dan tanggungjawab. Setiap peranan diwakili oleh satu holon. Satu atau lebih holon peranan membentuk satu holon bahagian.
- Holon bahagian – mewakili satu set bahagian yang wujud dalam organisasi (contohnya, kewangan dan sumber manusia). Setiap bahagian diwakili oleh satu holon. Satu atau lebih holon bahagian membentuk organisasi.



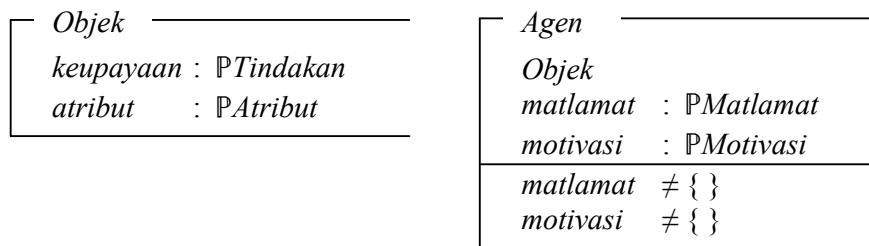
**Rajah 7.5 : Lapisan holon**

Lapisan holon juga merangkumi hubungan holon dengan elemen-elemen seperti berikut:

- (a) Persekitaran – holon sentiasa bergantung kepada persekitaran. Persekitaran berfungsi sebagai pemangkin tindakan holon dan medan untuk holon berinteraksi atau bertukar data.
- (b) Tugas – holon melaksanakan tugas-tugas tertentu untuk mencapai matlamat yang telah didelegasikan kepadanya.
- (c) Protokol interaksi – interaksi di antara holon yang ditakrif berdasarkan ciri-ciri kuasa, hubungan dan peranan holon.
- (d) Kebersandaran tindakan – memainkan fungsi yang penting di antara dua holon. Contohnya, jika holon A berperanan sebagai pelanggan, maka mesti terdapat holon B yang menjadi tempat untuk holon A bergantung bagi mendapatkan barangan atau perkhidmatan.

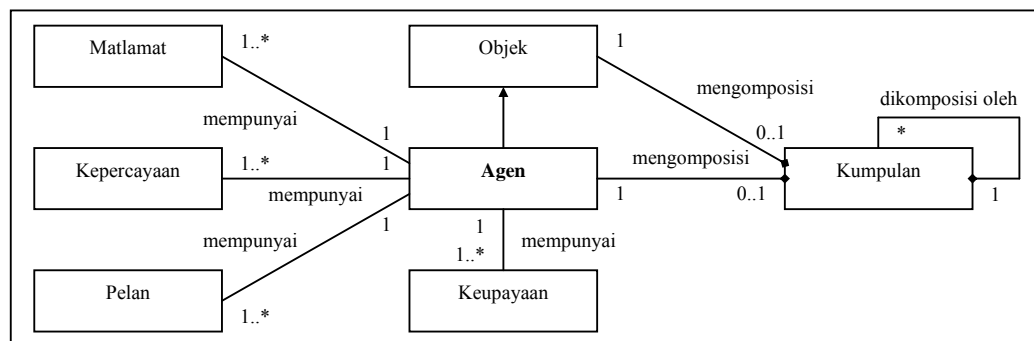
### 7.5.2 Lapisan Objek

Matlamat lapisan objek adalah untuk memetakan holon kepada objek atau agen. Setiap holon dianggap sebagai objek kerana holon mempunyai atribut holonik dan kelakuan holonik (Bongaerts, 1998). Bagaimanapun, holon aktif boleh dipetakan sebagai agen kerana agen adalah objek aktif yang mempunyai matlamat dan autonomi. Takrifan formal agen (Luck dan d'Inverno, 2001) adalah seperti berikut:



Kajian menggabungkan konsep holon dan agen yang dikenali sebagai agen holonik telah dijalankan oleh Gerber dan Klusch (2002), Burckert et al. (2000) dan Gerber et al. (1999). Tujuannya adalah untuk menangani sistem berskala besar yang melibatkan pemecahan tugas dan peruntukan sumber yang kompleks. Agen holonik berfungsi mewakili setiap unit, menyelaraskan dan mengawal aliran data serta bertindak secara berautonomi bagi pihak pengguna.

Peraturan pemilihan agen dan peraturan meta agen yang telah dibincangkan dalam Bab VI digunakan untuk mengecah holon aktif. Holon bahagian pula dipetakan kepada kumpulan. Setiap kumpulan dibentuk oleh objek atau agen. Rajah 7.6 menggambarkan lapisan objek. Agen dibentuk berasaskan seni bina BDI. Ia mempunyai atribut dan kelakuan tambahan iaitu matlamat, kepercayaan, pelan dan keupayaan.

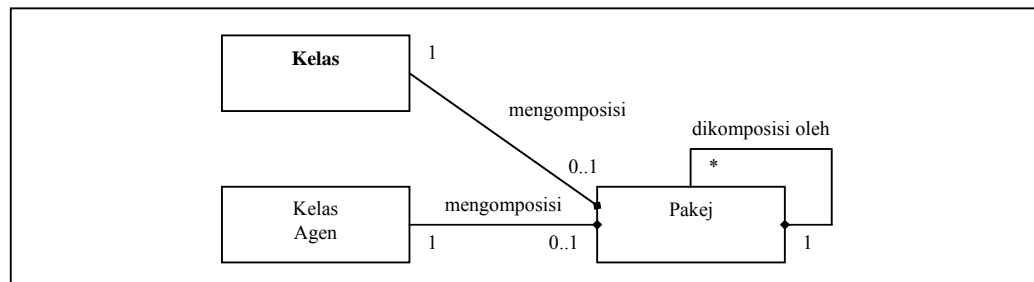


**Rajah 7.6 : Lapisan objek**

### 7.5.3 Lapisan Kelas

Lapisan kelas melibatkan terjemahan objek dan agen kepada kelas dan kelas agen yang boleh dikodkan dan dilarikan. Kumpulan pula diterjemahkan kepada

pakej. Kombinasi dan susunan pakej kemudian membentuk seni bina agen. Rajah 7.7 mempamerkan lapisan kelas.

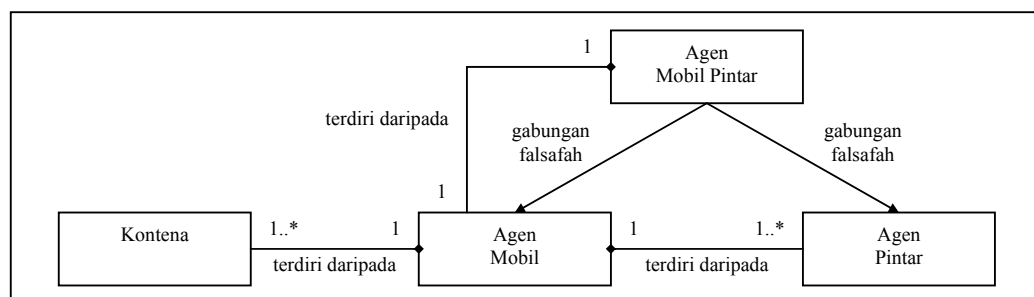


**Rajah 7.7 : Lapisan kelas**

## 7.6 Hirarki Ambients

Konsep hirarki di dalam ambients digunakan untuk membentuk susunan dan kerangka agen mobil pintar. Ciri-ciri agen mobil pintar berasaskan hirarki ambients yang diketengahkan adalah seperti pada Rajah 7.8, iaitu:

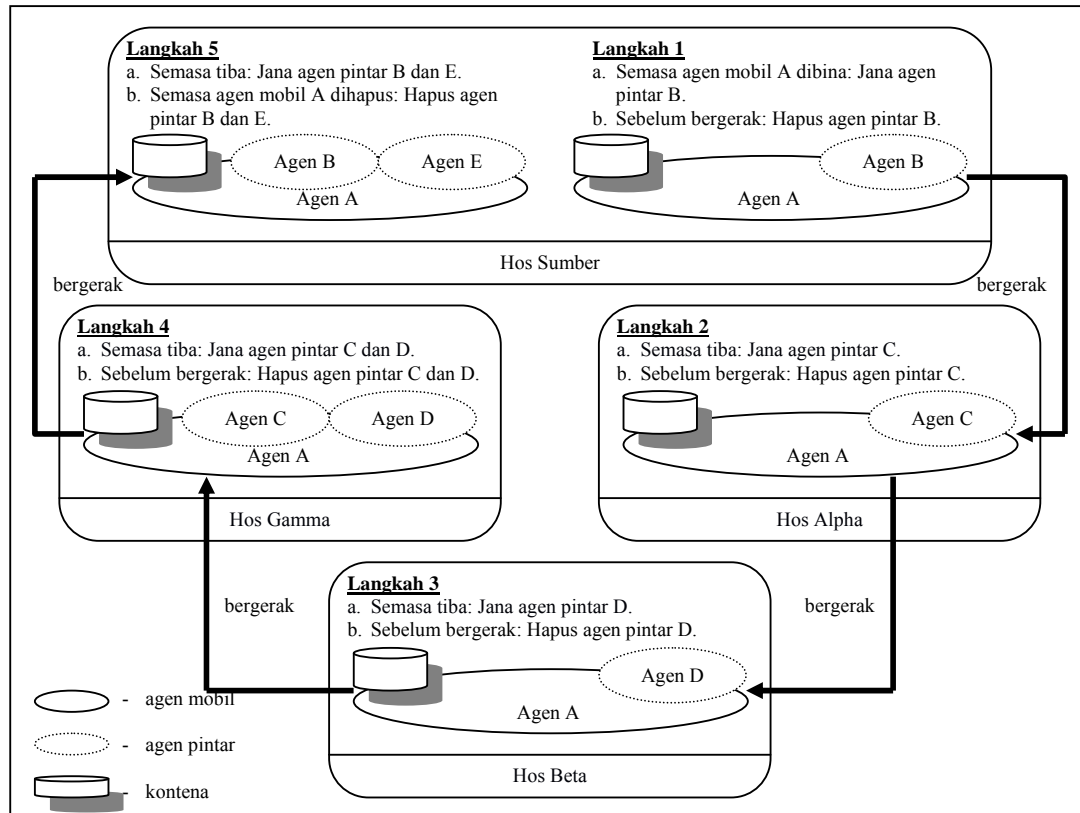
- (a) Dwihirarki – agen mobil pintar adalah gabungan falsafah agen mobil dan falsafah agen pintar yang membentuk agen mobil pintar dwihirarki. Hirarki pertama ialah agen mobil dan hirarki kedua ialah agen pintar. Satu agen mobil pintar terdiri daripada satu agen mobil. Agen mobil pula adalah ibu bapa kepada satu atau lebih agen pintar.
- (b) Kontena – agen mobil pintar mengandungi tempat simpanan yang boleh memuatkan maklumat untuk dibawa di sepanjang perjalanan. Satu atau lebih kontena dimuatkan dalam satu agen mobil.



**Rajah 7.8 : Hirarki ambients**



Hirarki ambients membenarkan agen mobil bergerak dari satu hos ke hos lain dengan menjana agen pintar untuk mencapai matlamat yang spesifik. Agen pintar akan dihapuskan sebaik sahaja matlamat terlaksana atau sebelum bermulanya proses migrasi. Situasi ini digambarkan pada Rajah 7.9.

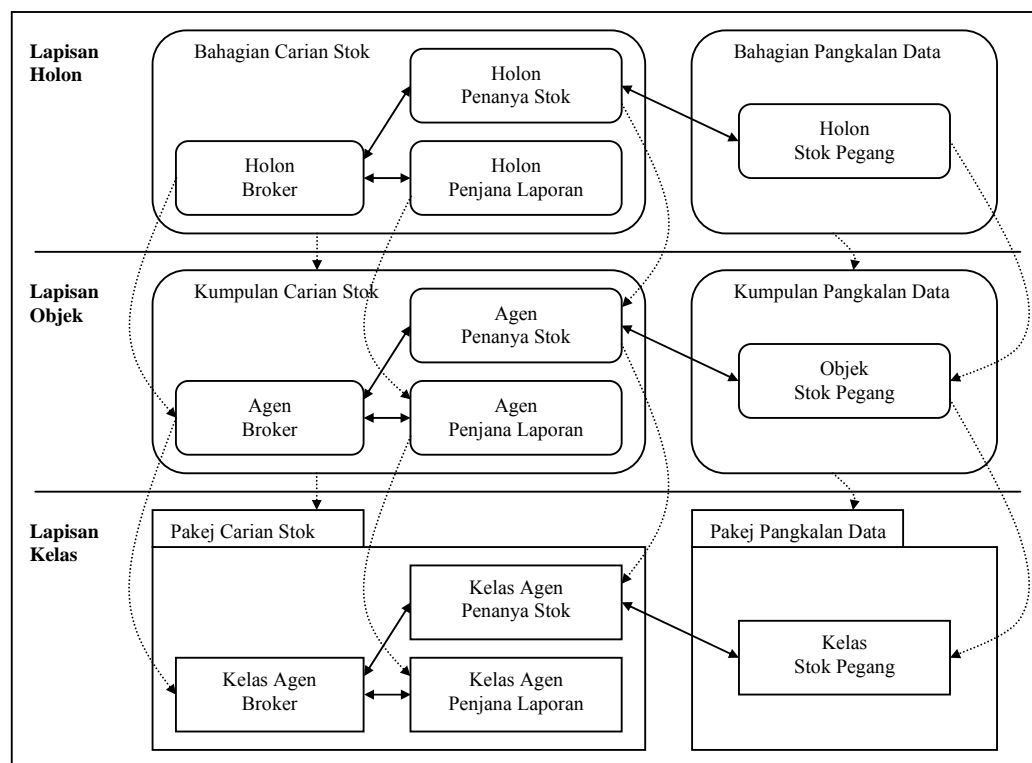


**Rajah 7.9 : Hirarki dan migrasi agen mobil pintar**

Dwihirarki digunakan untuk menghindarkan penjanaan agen mobil pintar yang monolitik dengan mengimplementasi teknologi pembangunan perisian berasaskan komponen (Broy et al., 1998). Ianya juga dapat mengurangkan bebanan bergerak oleh berbilang agen di dalam satu agen mobil pintar kerana hanya agen mobil sahaja dipindahkan. Agen pintar pula dihapuskan sebelum proses migrasi dilaksanakan. Kontena pula membolehkan agen mobil pintar membawa dan memanipulasi data, maklumat atau pengetahuan dari satu hos ke hos lain. Ianya juga dapat mengurangkan masa kependaman, lebar jalur dan transmisi data pertengahan ke atas hos semasa. Kontena adalah cadangan penyelesaian kepada masalah yang dikemukakan oleh Brewington dan Cybenko (2000) dalam kajian mereka ke atas penerokaan maklumat teragih menggunakan agen mobil.

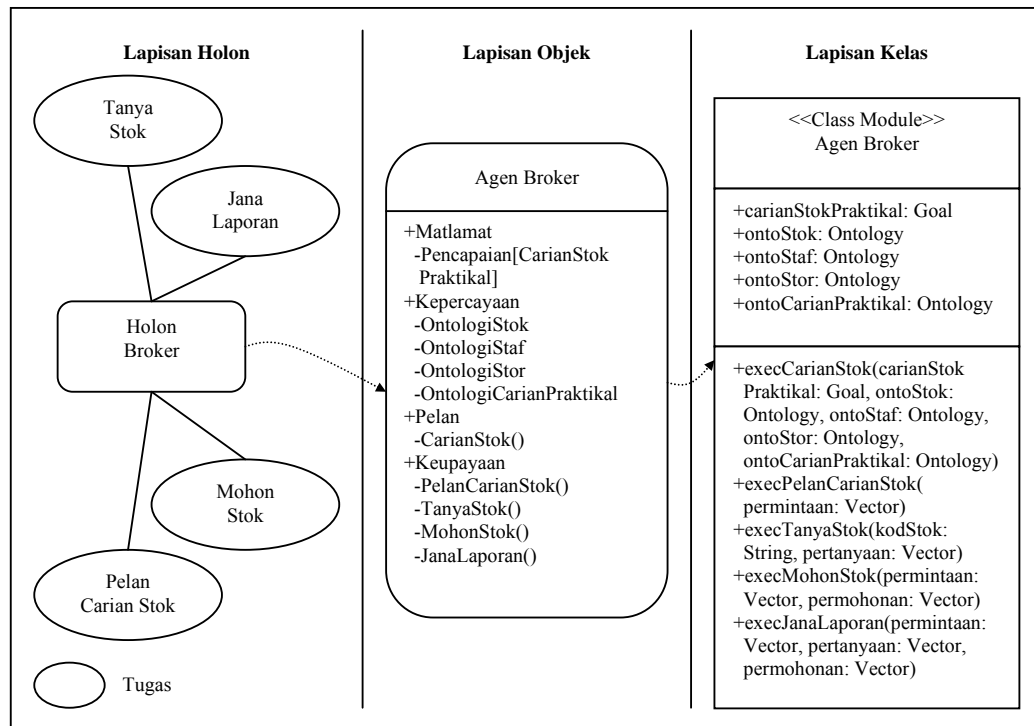
## 7.7 Implementasi Rangka Kerja Organisasi Holonik dan Hirarki Ambient dalam Pembangunan Seni Bina Sistem Pengurusan Inventori Teragih

Sampel implementasi rangka kerja organisasi holonik ke atas proses menghasilkan seni bina sistem pengurusan inventori teragih adalah seperti pada Rajah 7.10. Bahagian Carian Stok dibentuk oleh Holon Broker, Holon Penanya Stok, Holon Penjana Laporan dan lain-lainnya. Bahagian Carian Stok dipetakan kepada Kumpulan Carian Stok dan diterjemahkan kepada Pakej Carian Stok. Holon Broker pula dipetakan kepada Agen Broker dan diterjemahkan kepada Kelas Agen Broker. Keseluruhan seni bina sistem pengurusan inventori teragih dibincangkan dalam Bab VIII.



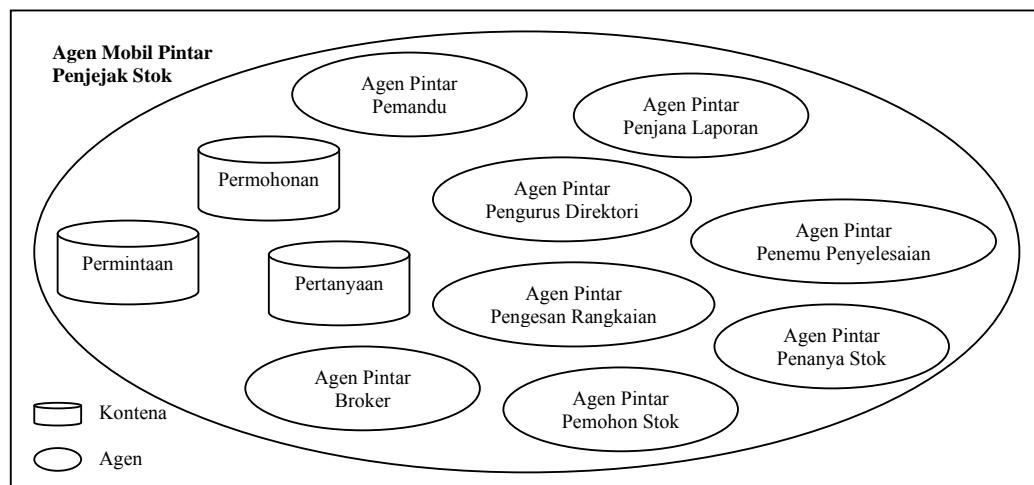
**Rajah 7.10 : Transformasi sistem pengurusan inventori teragih**

Contoh pemetaan dan terjemahan terperinci Holon Broker ditunjukkan pada Rajah 7.11. Holon Broker berperanan melaksanakan tugas `PelanCarianStok()`, `TanyaStok()`, `MohonStok()` dan `JanaLaporan()` yang telah didelegasikan kepadanya. Tugas holon kemudian diterjemahkan kepada operasi dalam lapisan kelas termasuk takrifan argumen yang dihantar masuk atau keluar dari operasi.



**Rajah 7.11 : Transformasi agen broker**

Contoh implementasi hirarki ambients yang digunakan untuk membolehkan agen mobil pintar disusun secara dwihirarki dan membawa informasi yang diperlukan pula adalah seperti pada Rajah 7.12. Agen Pintar Broker, Agen Pintar Pemandu, Kontena Permintaan, Kontena Permohonan dan lain-lain membentuk Agen Mobil Pintar Penjejak Stok yang mempunyai matlamat mengurus carian stok.



**Rajah 7.12 : Struktur agen mobil pintar penjejak stok**

## 7.8 Ringkasan

Rangka kerja organisasi holonik dan hirarki ambients yang diketengahkan ini telah digunakan dalam membangunkan seni bina sistem pengurusan inventori teragih seperti yang telah dibentangkan dalam Seksyen 7.7. Rangka kerja organisasi holonik adalah gabungan konsep organisasi agen dan holonik yang dibangunkan sebagai templat untuk menghasilkan seni bina agen. Rangka kerja hibrid ini melihat seni bina agen sebagai satu organisasi yang dibentuk oleh holon. Holon-holon akan membina bahagian-bahagian yang lebih besar dan seterusnya membentuk seni bina agen secara lengkap. Ia dapat menghasilkan seni bina agen yang mirip kepada organisasi sebenar dan berkomponen. Kelebihan seni bina agen berkomponen adalah komponen boleh dipalam masuk atau dipalam keluar tanpa menjejaskan seni bina agen sedia ada. Rangka kerja organisasi holonik ini juga menyokong sistem pelbagai agen. Hirarki ambients pula digunakan untuk menyokong pembinaan seni bina agen yang berupaya menghasilkan agen mobil pintar yang berkomponen, berhirarki dan tidak monolitik. Tujuannya adalah untuk memudahkan penyelarasan dan meningkatkan prestasi pergerakan agen mobil pintar di antara hos-hos yang teragih serta mendelegasikan tugas mengikut peranan yang telah ditetapkan.

## **BAB VIII**

### **SENI BINA BDI-MU: SENI BINA TERBUKA AGEN MOBIL PINTAR MENGUNAKAN CORAK SENI BINA HETEROGEN DAN MODEL PANDANGAN “4+1”**

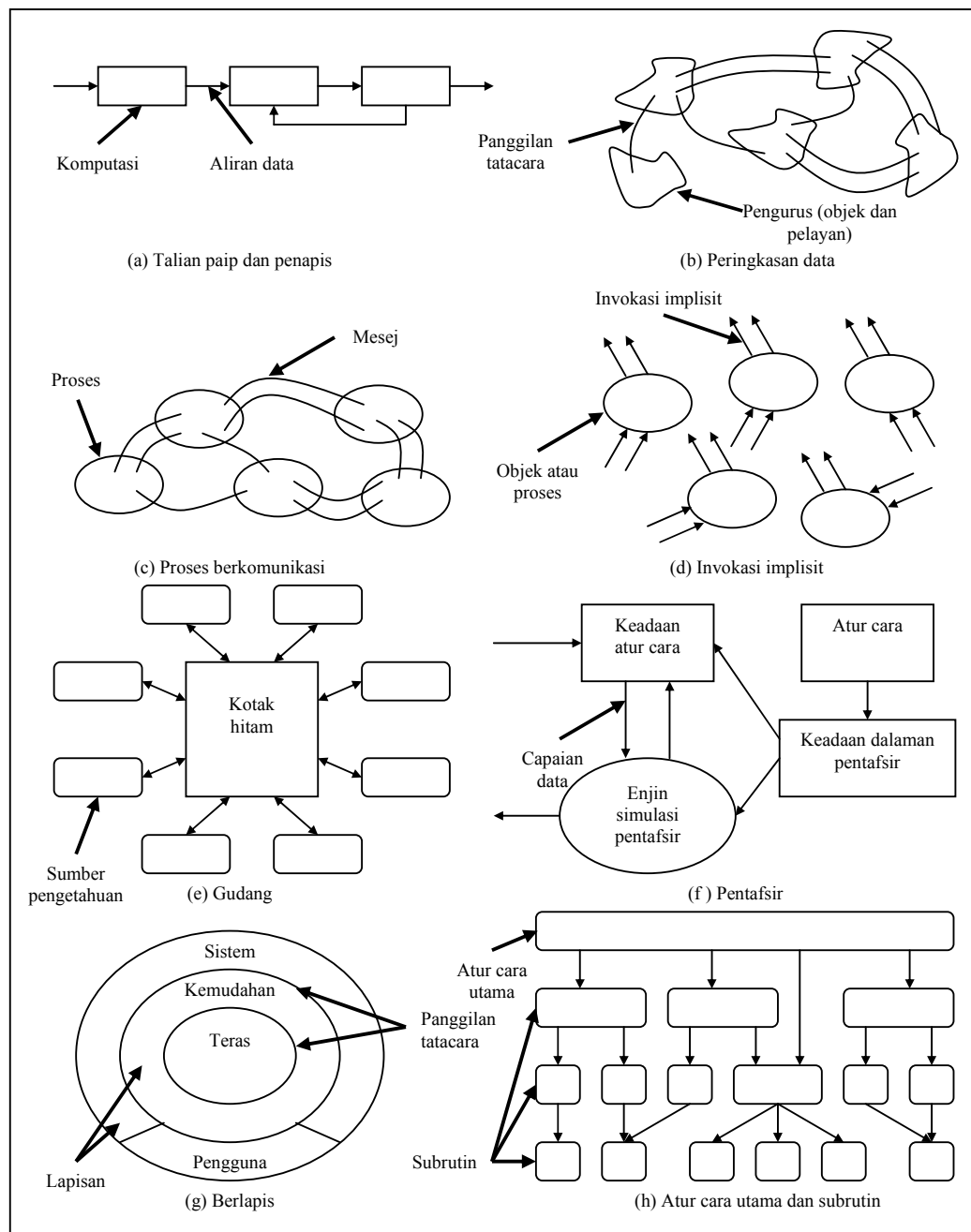
#### **8.1 Pengenalan**

Seni bina (Garlan, 1995) ditakrifkan sebagai struktur sesuatu atur cara atau sistem yang terdiri daripada komponen, atribut dan operasi komponen serta hubungan di antara komponen. Seni bina berfungsi menstabilkan keperluan melalui kemaskini berterusan dan menyediakan peringkasan kepada implementasi. Corak seni bina, pandangan seni bina, seni bina domain spesifik, seni bina baris produk, sintesis seni bina dan bahasa definisi seni bina adalah sebahagian dari elemen di dalam reka bentuk seni bina (Shaw et al., 1995). Setiap elemen memberi takrifan, perspektif dan servis yang berbeza tetapi mempunyai matlamat supaya untuk menggambarkan dan menghuraikan struktur statik atau dinamik komponen seni bina.

Objektif bab ini adalah untuk mengetengahkan seni bina BDI-MU yang dibentuk menggunakan corak seni bina heterogen dan model pandangan “4+1”. Tujuannya adalah untuk menghuraikan seni bina BDI-MU yang terbuka dengan ciri-ciri generik, fleksibel dan boleh disambung untuk pelbagai domain aplikasi teragih. Di dalam bab ini juga ditunjukkan bagaimana seni bina awalan sistem pengurusan inventori teragih yang telah dibina dalam Bab VII dipalam masuk ke dalam seni bina BDI-MU untuk membentuk seni bina agen mobil pintar yang lengkap. Ia bagi menjamin pengeluaran agen mobil pintar yang berautonomi, cerdas dan inter operasi.

## 8.2 Corak Seni Bina

Corak seni bina berfungsi menerangkan model masalah agen dan menjelaskan organisasi agen. Selain membantu membina dan menyenggara struktur agen serta menganalisa dan menyemak konsistensi agen. Sebahagian corak seni bina yang diidentifikasi oleh Shaw (1995) adalah seperti pada Rajah 8.1.



Rajah 8.1 : Corak seni bina

Corak seni bina tersebut telah digunakan ke atas sistem agen yang berlainan mengikut kesesuaian, iaitu:

- (a) Talian paip dan penapis – untuk sistem agen yang memerlukan aliran data secara bersiri di antara komponen seperti CIDRE (Bapst et al., 1996).
- (b) Peringkasan data – untuk sistem agen yang saling bersandaran dan berpusat seperti Web Store (Ardissono et al., 2002).
- (c) Proses berkomunikasi – untuk sistem agen dengan proses yang saling berkomunikasi dan tidak bersandaran seperti Tok (Bates, 1994).
- (d) Invokasi implisit – untuk sistem agen dengan proses reaktif yang tidak bersandaran seperti MobiDoc (Satoh, 2001).
- (e) Gudang – untuk sistem agen dengan data berpusat dan berkongsi data seperti OAA (Martin et al., 1999).
- (f) Pentafsir – untuk sistem agen yang memerlukan takrifan notasi untuk menjana penyelesaian seperti dMARS (Kinny dan Georgeff, 1997).
- (g) Berlapis – untuk sistem agen yang berhirarki dengan pelbagai lapisan servis yang berlainan seperti AHA (Bruderlin et al., 1997).
- (h) Atur cara utama dan subrutin – untuk sistem agen yang berhirarki secara subsistem atau modulariti seperti MACRON (Decker dan Sycara, 1997).

Corak seni bina heterogen pula merupakan gabungan dua atau lebih corak seni bina hibrid. Ianya telah diimplementasi oleh sistem agen seperti RobologKoblenz-00 (Chesnevar et al., 2002) dan AIS (Hayes-Roth, 1995).

### **8.3 Pandangan Seni Bina**

Pandangan seni bina bertujuan untuk memastikan penghuraian sistem kepada komponen selari dengan keperluan kefungsiian, keperluan bukan kefungsiian dan kekangan. Di antara pandangan seni bina adalah pandangan elemen (Perry dan Wolf, 1992), model pandangan “4+1” (Kruchten, 1995) dan empat pandangan seni bina (Nord et al., 2001).

Model pandangan “4+1” menghuraikan seni bina dari lima pandangan yang berbeza, iaitu:

- (a) Pandangan lojikal – menerangkan komponen seni bina yang terdiri daripada pakej, subsistem dan kelas. Ia merangkumi peranan, hubungan dan operasi pakej, subsistem dan kelas.
- (b) Pandangan proses – menerangkan komponen serempak seni bina yang sedang berjalan seperti proses atau bebenang. Ia termasuk kawalan bebenang, hebahan mesej dan sampukan.
- (c) Pandangan implementasi – menerangkan organisasi komponen statik seni bina seperti kod sumber, pangkalan data dan fail boleh laku.
- (d) Pandangan penempatan – menerangkan pemetaan komponen boleh laku dan komponen serempak kepada nod-nod fizikal seperti komputer dan unit pemprosesan pusat (*central processing unit*, CPU). Selain inter sambungan di antara nod fizikal seperti bas, titik ke titik dan rangkaian kawasan setempat (*local area network*, LAN).
- (e) Pandangan kes gunaan – menerangkan peranan dan hubungan di antara pelaku dan kes gunaan serta keperluan kefungisian.

Pandangan-pandangan tersebut digambarkan oleh model reka bentuk, model proses, model implementasi, model penempatan dan model kes gunaan. Model digunakan sebagai mekanisme untuk mewakili dan menggambarkan setiap pandangan melalui notasi-notasi tertentu seperti notasi UML, OMT, Booch dan Coad-Yourdon.

Empat pandangan seni bina pula mengfokus kepada pandangan konseptual, pandangan modul, pandangan perlakuan dan pandangan kod. Pandangan elemen pula memberi penekanan ke atas pandangan proses, pandangan data dan pandangan sambungan.

#### **8.4 Seni Bina BDI-MU**

Corak seni bina heterogen dibentuk melalui pelbagai cara. Antaranya ialah secara hirarki iaitu struktur dalaman komponen seni bina di organisasi menggunakan pelbagai corak seni bina hibrid (Monroe et al., 1993). Paras konteks seni bina BDI-



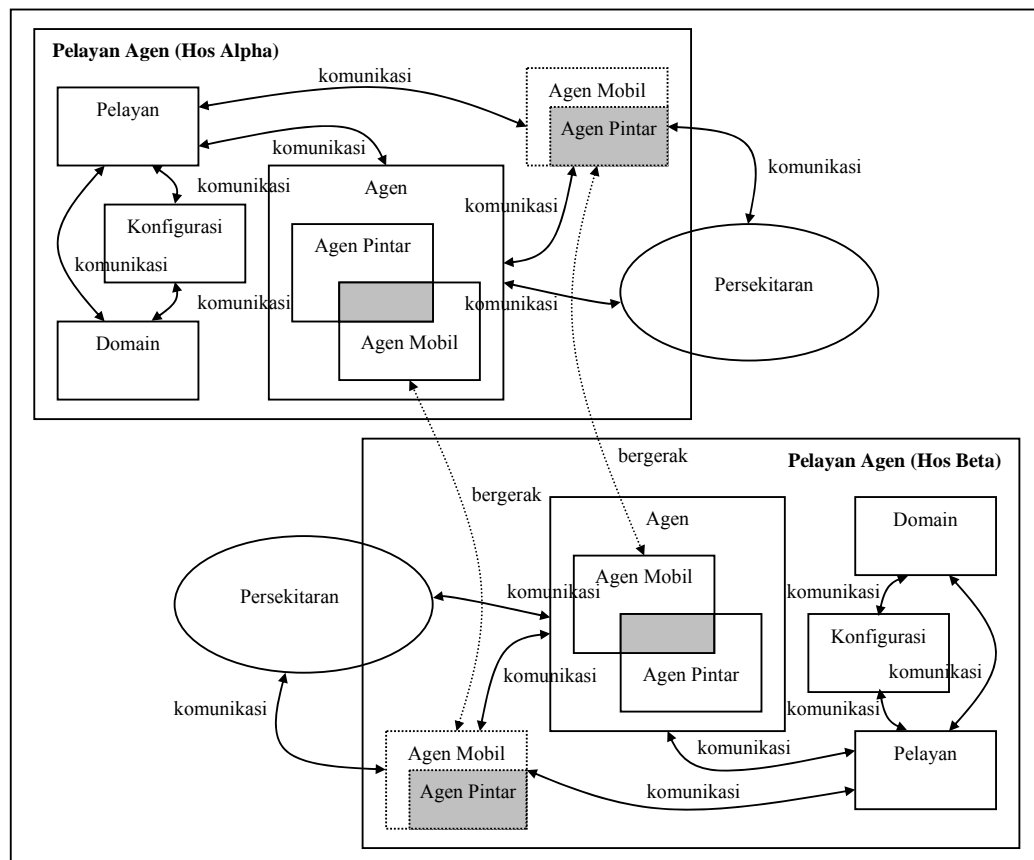
MU dibentuk oleh corak seni bina proses berkomunikasi. Domain, pelayan, konfigurasi, agen dan persekitaran adalah proses yang saling berinteraksi menggunakan mesej. Interaksi bagi setiap proses tersebut adalah seperti pada Jadual 8.1.

**Jadual 8.1 : Interaksi di antara proses dalam seni bina BDI-MU**

<b>Penghantar</b>	<b>Penerima</b>	<b>Huraian Interaksi</b>
Domain	Pelayan	Mesej dihantar apabila terdapat pergerakan agen di antara hos atau pelayan berstatus dalam talian atau luar talian.
	Konfigurasi	Mesej dihantar apabila terdapat pertanyaan parameter rangkaian atau keselamatan domain oleh agen.
Pelayan	Domain	Mesej dihantar apabila terdapat agen dijana atau dihapuskan atau pelayan berstatus dalam talian atau luar talian.
	Konfigurasi	Mesej dihantar apabila terdapat pertanyaan parameter rangkaian atau keselamatan pelayan oleh agen.
	Agen	Mesej dihantar apabila terdapat komunikasi di antara agen pada hos yang berlainan.
Konfigurasi	Domain	Mesej dihantar apabila domain hendak dilarikan.
	Pelayan	Mesej dihantar apabila pelayan hendak dilarikan.
Agen	Pelayan	Mesej dihantar apabila agen ingin berkomunikasi dengan agen yang lain pada hos yang berlainan.
	Agen	Mesej dihantar agen ingin berkomunikasi dengan agen yang lain pada hos yang sama.
	Persekitaran	Mesej dihantar apabila agen ingin berkomunikasi dengan pengguna, aplikasi atau pangkalan data.
Persekitaran	Agen	Mesej dihantar apabila pengguna, aplikasi atau pangkalan data ingin menghantar data kepada agen.

Setiap hos boleh menjana agen masing-masing dengan identiti unik untuk tujuan pengecaman dan identifikasi. Agen mencakupi agen mobil, agen pintar dan agen mobil pintar. Agen mobil pintar dibentuk oleh kombinasi agen mobil dan agen

pintar. Agen dengan keupayaan mobiliti boleh bergerak di antara hos melalui rangkaian. Sempadan komunikasi dan mobiliti agen adalah terhad kepada skop yang telah ditetapkan oleh domain. Oleh kerana itu, setiap pelayan dalam hos akan didaftarkan kepada sesuatu domain untuk membolehkan agen berkomunikasi atau bergerak di antara hos-hos yang mendaftar kepada domain tersebut. Seni bina BDI-MU adalah seperti yang ditunjukkan pada Rajah 8.2.



**Rajah 8.2 : Seni bina BDI-MU**

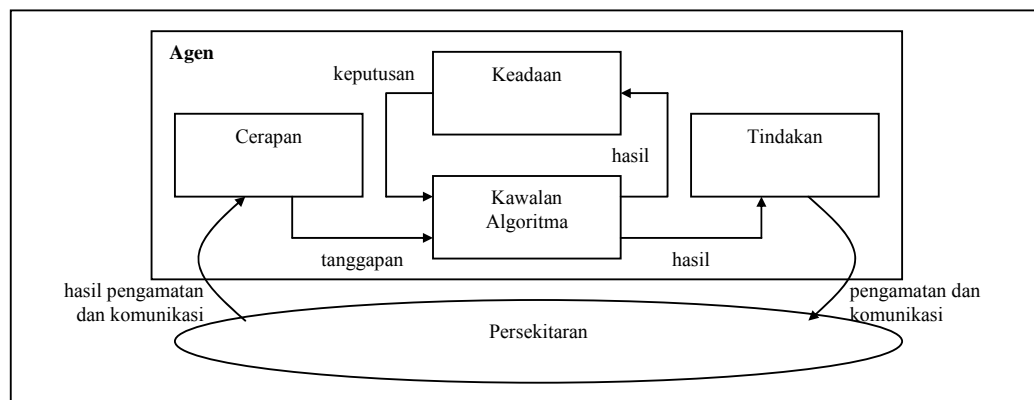
#### 8.4.1 Corak Seni Bina Heterogen

##### 8.4.1.1 Agen, Agen Pintar, Agen Mobil dan Agen Mobil Pintar

Corak seni bina kawalan gelung (Lozano-Perez, 1985) dihibridkan dengan corak seni bina peringkasan data untuk membentuk struktur agen seperti pada Rajah

8.3. Agen terdiri daripada komponen cerapan, keadaan, kawalan algoritma dan tindakan yang mempunyai fungsi seperti berikut:

- (a) Cerapan – membolehkan agen membuat pengamatan dan penganalisan mudah ke atas persekitarannya.
- (b) Keadaan – membolehkan agen membuat keputusan berdasarkan pengaruh maklumat peribadi yang dimiliki oleh agen.
- (c) Kawalan algoritma – membolehkan agen melaksanakan operasi untuk mencapai matlamatnya berdasarkan aras autonomi yang telah ditakrifkan.
- (d) Tindakan – membolehkan agen melaksanakan tindakan ke atas persekitarannya.

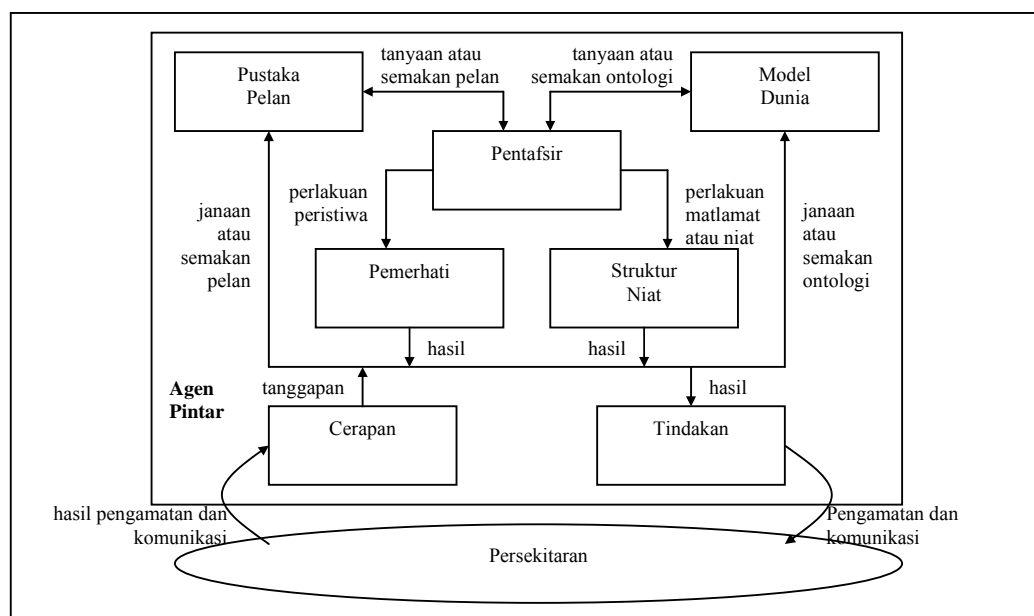


**Rajah 8.3 : Struktur agen**

Agen pintar dibentuk oleh corak seni bina kawalan gelung dan pentafsir seperti pada Rajah 8.4. Setiap input daripada persekitaran akan ditafsir dan diproses oleh agen pintar secara taakulan praktikal bagi menjana tindakan yang akan disuap balik kepada persekitaran. Taakulan praktikal adalah proses memutuskan apakah yang perlu dibuat bersamaan dengan taakulan yang dilakukan oleh manusia dalam kehidupan harian untuk mencapai sesuatu matlamat. Taakulan praktikal ini adalah berasaskan seni bina BDI. Ia dibuat melalui olahan komponen berikut:

- (a) Model dunia – pangkalan data yang menyimpan kepercayaan yang dimiliki oleh agen.
- (b) Pustaka pelan – koleksi tatacara yang boleh digunakan oleh agen untuk mencapai matlamatnya.

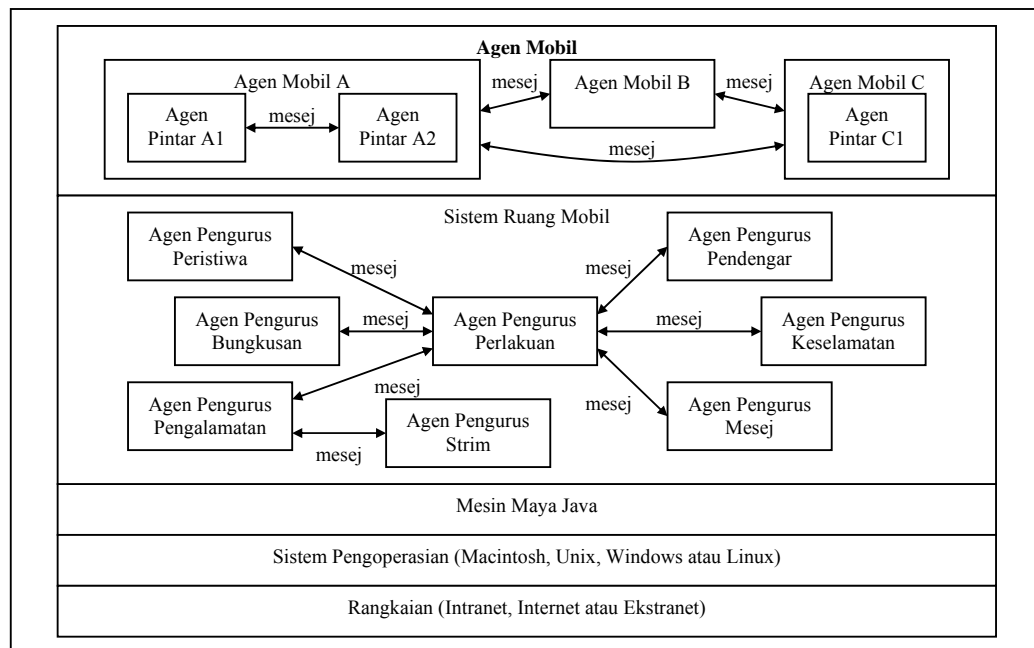
- (c) Pentafsir – otak agen yang akan menaakul mengenai apakah yang perlu buat, bila dan bagaimana ia dilakukan.
- (d) Struktur niat – model dalaman mengenai matlamat semasa agen. Ia berfungsi mengawal dan mengendalikan komitmen dan kemajuan yang dicapai oleh agen sepanjang menyempurnakan sesuatu matlamat.
- (e) Pemerhati – tatacara yang digunakan untuk melaksanakan fungsi yang diluar skop taakulan berasaskan matlamat atau pelan (contohnya, menimbal mesej masuk).



**Rajah 8.4 : Struktur agen pintar**

Agan mobil adalah inter platform kerana ia boleh beroperasi dalam pelbagai sistem pengoperasian dengan sokongan JVM. Agan pengurus perlakuan, agan pendengar, agan pengurus peristiwa, agan pengurus keselamatan, agan pengurus bungkusan, agan pengurus pengalamatan dan agan pengurus strim yang membentuk sistem ruang mobil diperlukan untuk mengangkut dan menyediakan kemudahan bagi membolehkan agan mobil bergerak di sepanjang rangkaian. Agan mobil berupaya berpindah dalam rangkaian internet, intranet atau ekstranet melalui protokol kawalan transmisi/protokol internet (*transmission control protocol/internet protocol*, TCP/IP). Agan mobil seperti pada Rajah 8.5 dibentuk oleh gabungan corak seni bina berlapis dan proses berkomunikasi. Corak seni bina hibrid ini umumnya digunakan untuk

membentuk seni bina mikro kernel sistem pengoperasian (Michel, 1990) atau seni bina agen mobil seperti MAF (Eng et al., 2002) dan DIM Agents (Dale, 1997).

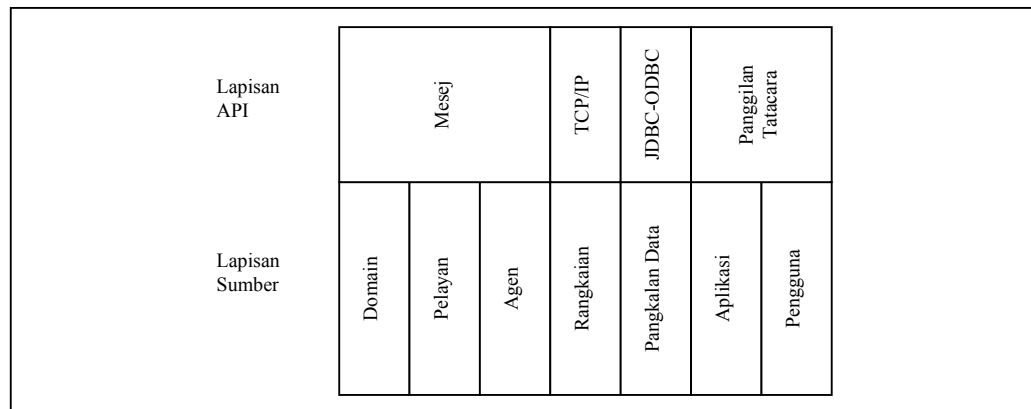


**Rajah 8.5 : Struktur agen mobil**

Gabungan struktur agen mobil dan agen pintar digunakan untuk membentuk agen mobil pintar. Sebagaimana yang ditunjukkan pada Rajah 8.4 dan 8.5, agen mobil pintar berupaya membuat keputusan secara taakulan praktikal dan bergerak di sepanjang rangkaian untuk meningkatkan keupayaannya membuat kata putus.

#### 8.4.1.2 Persekitaran

Corak seni bina berlapis horizontal dan vertikal (Muller et al., 1994) diadun untuk membentuk struktur persekitaran seperti pada Rajah 8.6. Interaksi dengan lapisan sumber yang menyediakan data, maklumat atau pengetahuan adalah melalui medium antara muka pengaturcaraan aplikasi (*application programming interface*, API). Lapisan API terdiri daripada mesej, TCP/IP, penyambungan pangkalan data Java-penyambungan pangkalan data terbuka (*Java database connectivity-open database connectivity*, JDBC-ODBC) atau panggilan tatacara.



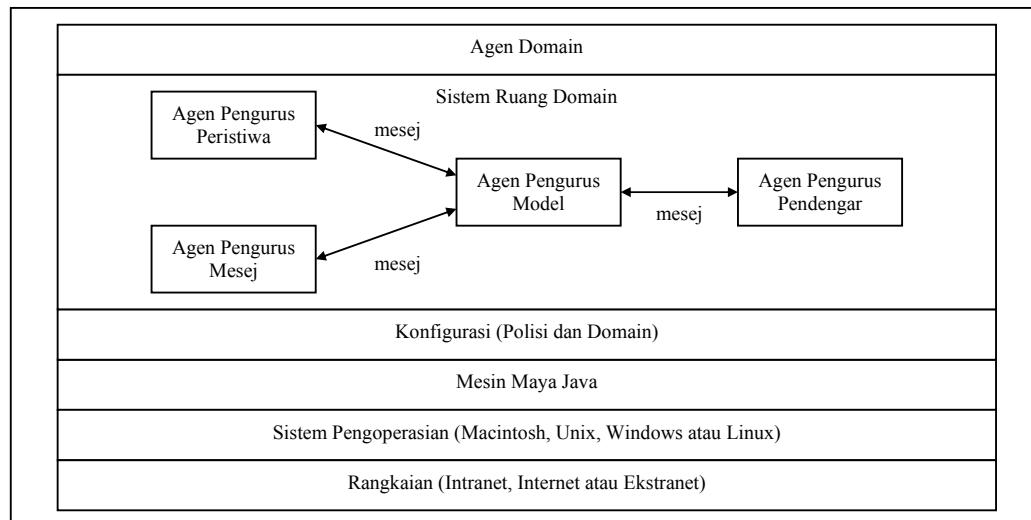
**Rajah 8.6 : Struktur persekitaran**

### 8.4.1.3 Konfigurasi, Domain dan Pelayan

Konfigurasi mengandungi tiga komponen utama iaitu polisi, konfigurasi domain dan konfigurasi pelayan dengan peranan seperti berikut:

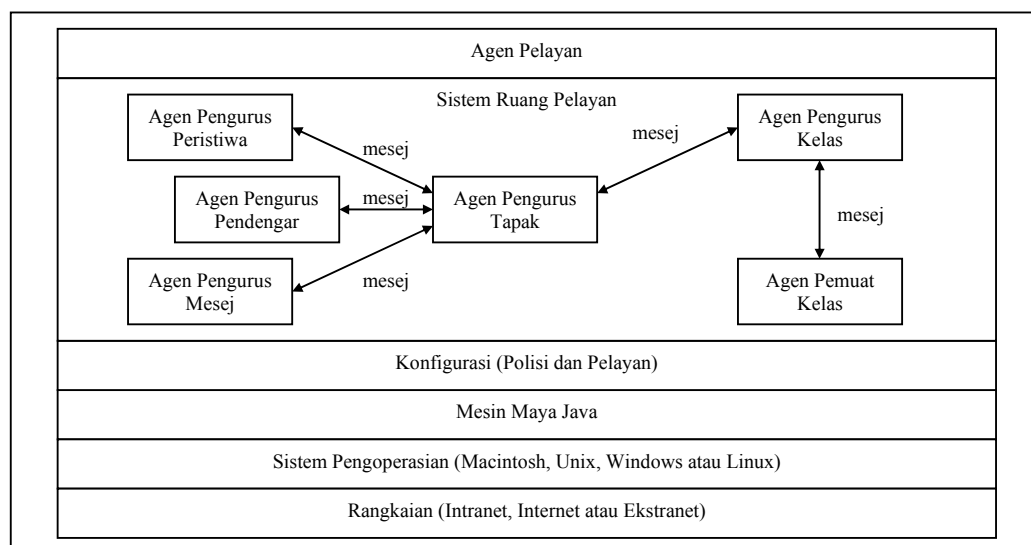
- (a) Polisi – menyetkan parameter kebenaran dan keselamatan seperti soket kebenaran dan port pendengar.
- (b) Konfigurasi domain – menyetkan parameter domain seperti alamat IP domain dan port domain.
- (c) Konfigurasi pelayan – menyetkan parameter pelayan seperti alamat IP pelayan, port pelayan dan direktori lalai.

Domain berfungsi mengurus satu set pelayan. Setiap pelayan perlu mendaftar kepada sesuatu domain. Ini bagi membolehkan agen yang dijana oleh sesuatu pelayan bergerak dalam skop dan lingkungan pelayan-pelayan yang berdaftar kepada domain tersebut. Domain juga memerlukan JVM untuk menyokong lariannya dan ia boleh beroperasi dalam pelbagai sistem pengoperasian seperti Macintosh, Unix, Windows dan Linux. Selain memerlukan rangkaian untuk berkomunikasi dengan satu set pelayan yang saling terangkai dan berinteraksi di antara satu sama lain. Sistem ruang domain pula berperanan menyokong operasi domain. Sistem ruang domain dibentuk oleh agen pengurus model, agen pengurus peristiwa, agen pengurus pendengar dan agen pengurus mesej. Struktur domain adalah seperti pada Rajah 8.7.



**Rajah 8.7 : Struktur domain**

Pelayan bertanggungjawab menyediakan infrastruktur dan mengurus agen yang terdapat di dalam hosnya. JVM diperlukan untuk membolehkan pelayan beroperasi dalam pelbagai sistem pengoperasian. Rangkaian pula digunakan untuk memindahkan agen dari satu hos ke hos yang lain dan berkomunikasi dengan domain atau pelayan-pelayan lain. Agen pengurus tapak, agen pengurus peristiwa, agen pengurus mesej, agen pengurus pendengar, agen pengurus kelas dan agen pemuat kelas yang membentuk sistem ruang pelayan berfungsi mengendalikan pelayan. Struktur pelayan adalah seperti pada Rajah 8.8.

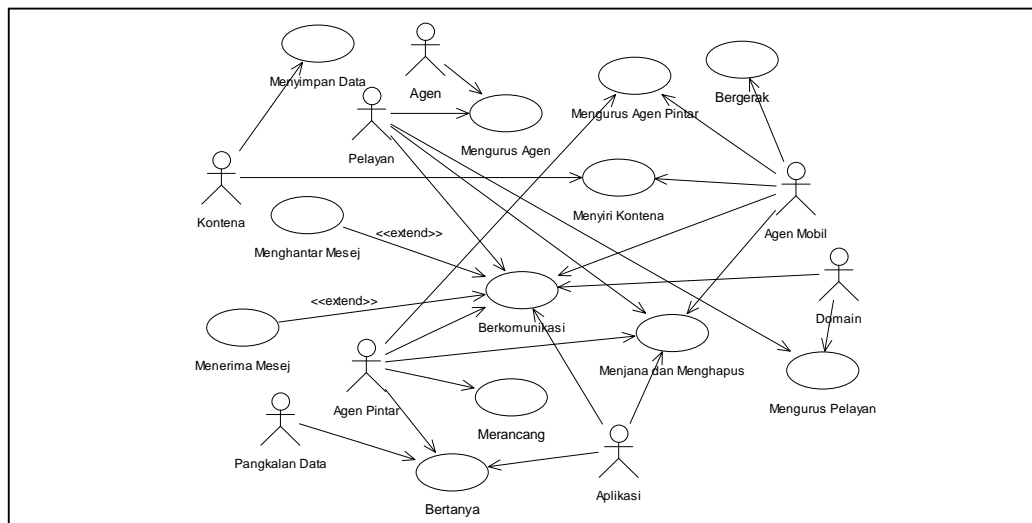


**Rajah 8.8 : Struktur pelayan**

## 8.4.2 Model Pandangan “4+1”

### 8.4.2.1 Pandangan Kes Gunaan

Pandangan kes gunaam menerangkan senario yang berlaku dalam seni bina BDI-MU. Rajah 8.9 menunjukkan pelaku dan kes gunaam serta hubungan di antaranya menggunakan gambarajah kes gunaam. Pelaku yang terlibat adalah Pelayan, Agen Mobil, Pangkalan Data dan sebagainya. Kes gunaam yang terlibat pula ialah Bergerak, Berkomunikasi, Menghantar Mesej dan sebagainya. Seni bina BDI-MU dibentuk oleh satu set pelaku yang membina satu organisasi dengan aras kawalan, komunikasi dan peranan yang berbeza. Agen Mobil berupaya untuk bergerak di sepanjang rangkaian, Agen Pintar berupaya menaakul untuk mencapai sesuatu matlamat dan Kontena berupaya untuk menyimpan data yang boleh dibawa di sepanjang perpindahan.



**Rajah 8.9 : Gambarajah kes gunaam seni bina BDI-MU**

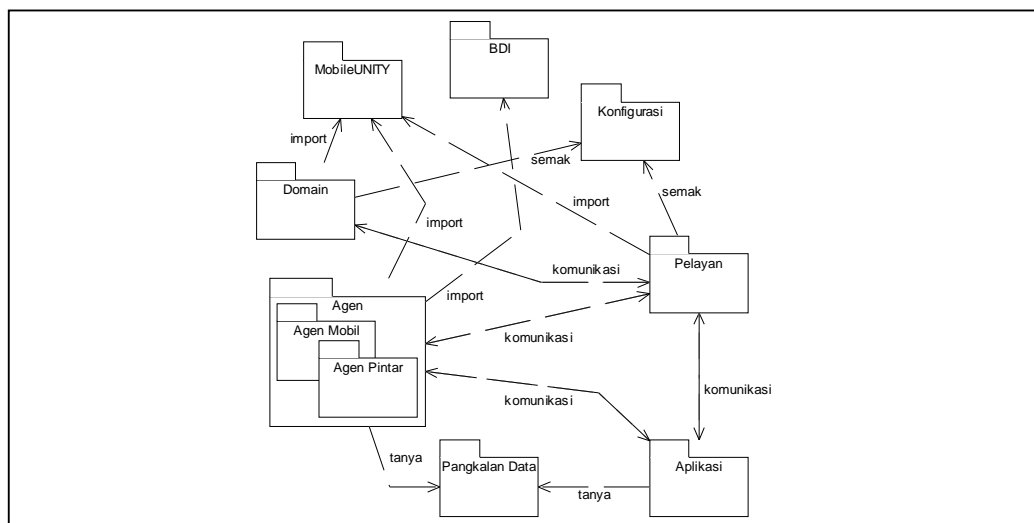
### 8.4.2.2 Pandangan Lojikal

Seni bina BDI-MU dibentuk oleh pakej takrifan pengguna dan pakej sistem. Pakej takrifan pengguna adalah pakej yang dipalam masuk oleh pembangun sistem.



Ia dibina mengikut objektif dan keperluan sistem agen yang hendak dibangunkan. Pakej takrifan pengguna terdiri daripada pakej Agen, Aplikasi dan Pangkalan Data. Pakej sistem pula adalah pakej tersedia ada yang boleh disambung atau diimport. Pakej sistem terdiri daripada pakej Domain, Pelayan, Konfigurasi, MobileUNITY dan BDI. Gambarajah pakej digunakan untuk menggambarkan pakej dan hubungan di antara pakej seperti pada Rajah 8.10, iaitu:

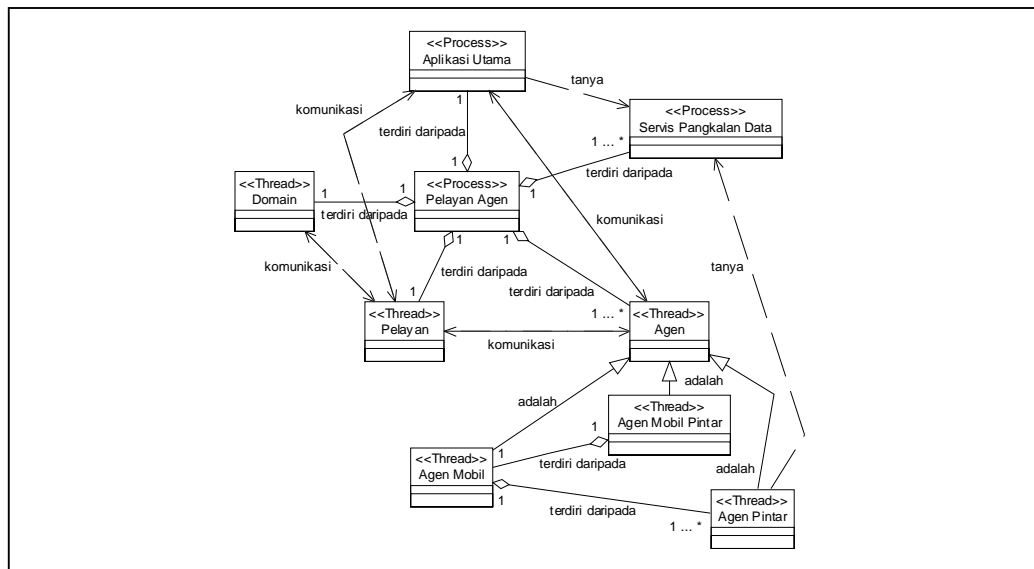
- (a) Domain – pakej untuk membentuk domain agen yang berfungsi mengurus satu set pelayan agen.
- (b) Pelayan – pakej untuk membentuk pelayan agen yang berfungsi mengurus satu set agen.
- (c) Konfigurasi – pakej untuk mengkonfigurasi pelayan dan domain agen.
- (d) Agen – pakej yang mengandungi agen termasuk agen mobil, agen pintar dan agen mobil pintar.
- (e) MobileUNITY – pakej yang mengandungi kelas-kelas untuk membentuk struktur inter agen.
- (f) BDI – pakej yang mengandungi kelas-kelas untuk membentuk struktur intra agen.
- (g) Aplikasi – pakej yang membentuk sistem agen (seperti sistem pengurusan pustaka teragih dan sistem pengurusan hartanah teragih).
- (h) Pangkalan Data – pakej yang mengandungi pangkalan data sistem agen.



**Rajah 8.10 : Gambarajah pakej seni bina BDI-MU**

### 8.4.2.3 Pandangan Proses

Proses dan benang yang sedang berjalan termasuk interaksinya dapat ditunjukkan menggunakan gambarajah kelas proses. Ianya seperti yang ditunjukkan pada Rajah 8.11. Perlakuan seni bina BDI-MU melibatkan larian tidak segerak benang domain, pelayan dan agen. Selain itu, ia turut merangkumi pelaksanaan proses aplikasi dan pangkalan data.

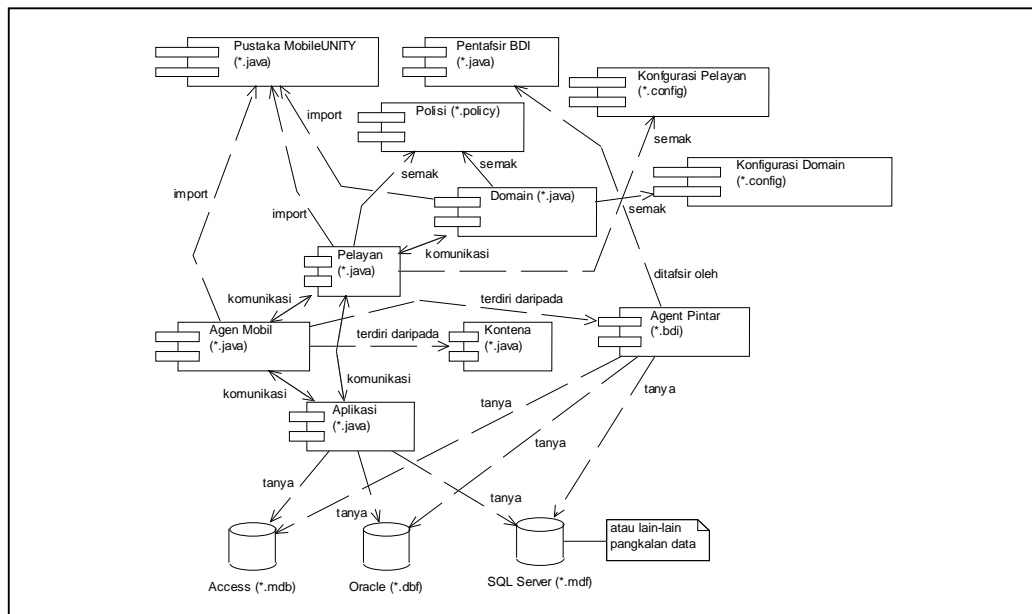


**Rajah 8.11 : Gambarajah kelas proses seni bina BDI-MU**

### 8.4.2.4 Pandangan Implementasi

Hampir keseluruhan komponen seni bina BDI-MU dibangunkan menggunakan bahasa Java seperti yang ditunjukkan oleh gambarajah komponen pada Rajah 8.12. Bagaimanapun, Agen Pintar dibina berasaskan Pentafsir BDI dengan fail yang mempunyai sambungan *bdi* (*\*.bdi*). Petafsir BDI turut dibangunkan menggunakan bahasa Java. Oracle, SQL Server, Sybase, MySQL, Microsoft Access dan lain-lain boleh digunakan untuk membentuk Pangkalan Data. Fail dengan sambungan *config* (*\*.config*) dan *policy* (*\*.policy*) pula digunakan untuk mengkonfigurasi parameter rangkaian dan keselamatan bagi Domain dan Pelayan.

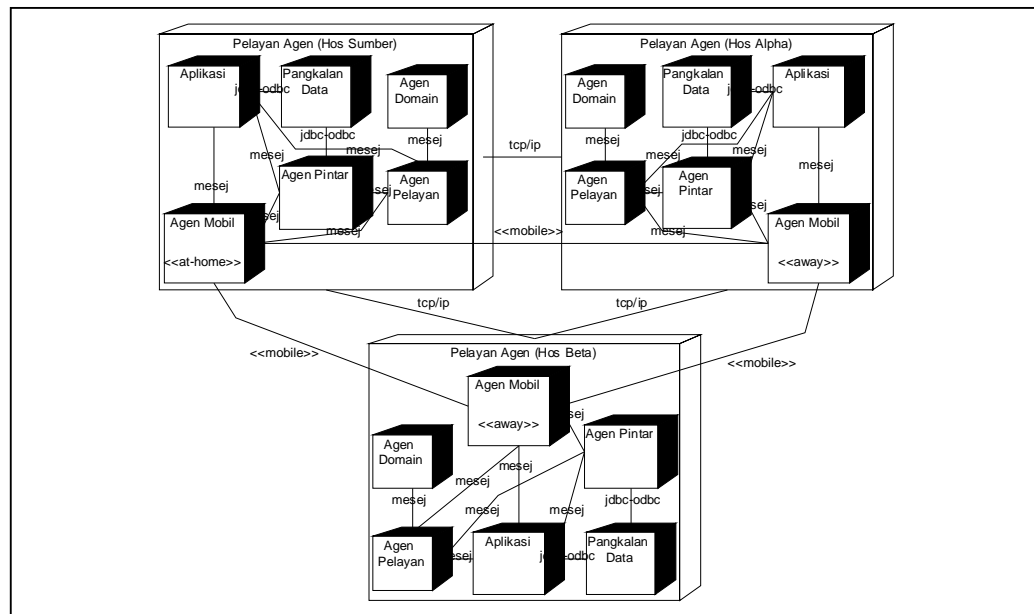
Pentafsir BDI dan Pustaka MobileUNITY merupakan komponen teras dalam pembentukan agen. Ini adalah kerana agen yang dihasilkan perlu mengimport, menyambung atau ditafsir oleh kelas-kelas yang terdapat di dalam komponen ini. Ia bagi membolehkan agen berkenaan menjadi autonomi, reaktif, proaktif, keupayaan bersosial dan menukar lokasinya secara dinamik.



**Rajah 8.12 : Gambarajah komponen seni bina BDI-MU**

#### 8.4.2.5 Pandangan Penempatan

Gambarajah penempatan seni bina BDI-MU adalah seperti pada Rajah 8.13. Tujuannya ialah untuk menggambarkan pandangan penempatan yang menerangkan pemetaan perisian kepada perkakasan dan kesannya ke atas ciri-ciri teragih sistem agen. Agen dengan keupayaan mobiliti akan bergerak di antara hos dan berkomunikasi dengan pangkalan data, aplikasi atau agen lain. Notasi <<mobile>> diserap daripada AUML (Bauer et al., 2001). Matlamatnya ialah untuk menggambarkan kelakuan agen mobil atau agen mobil pintar yang berupaya memindahkan keadaan data dan keadaan kawalannya di antara hos yang saling berhubungan melalui rangkaian.



**Rajah 8.13 : Gambarajah penempatan seni bina BDI-MU**

## 8.5 Implementasi Seni Bina BDI-MU dalam Pembangunan Seni Bina Sistem Pengurusan Inventori Teragih

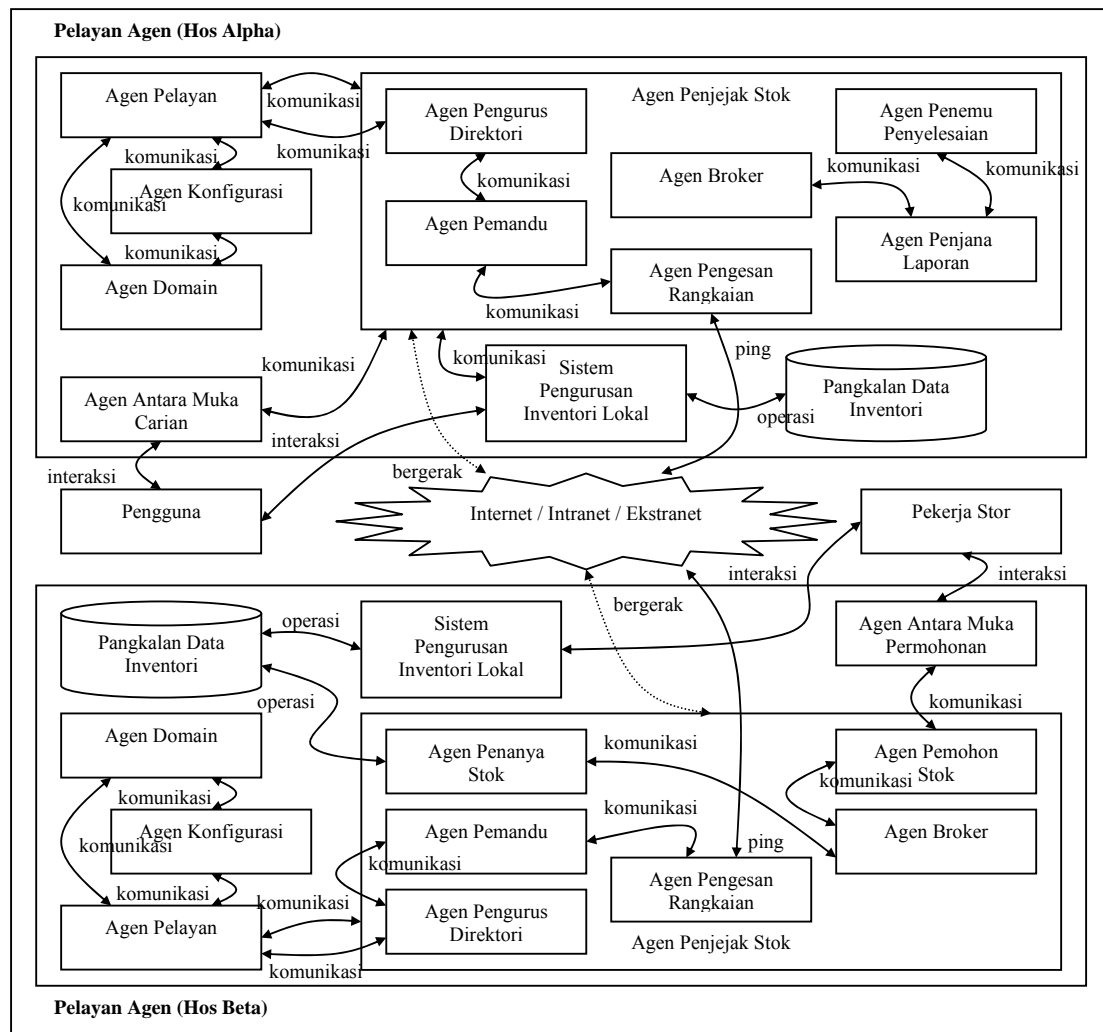
Seni bina yang telah di reka bentuk pada Bab VII adalah seni bina awalan yang tidak dapat berfungsi dengan sempurna kerana ia hanya menyediakan pakej takrifan pengguna. Oleh kerana itu, ia perlu dipalam masuk ke dalam seni bina BDI-MU untuk membolehkannya beroperasi di dalam persekitaran teragih. Tujuannya adalah untuk membolehkan utiliti dan komponen di dalam pakej takrifan sistem yang disediakan oleh seni bina BDI-MU seperti pelayan, domain, konfigurasi, BDI dan MobileUNITY diimport atau disambung untuk menyediakan platform operasi.

Seni bina sistem pengurusan inventori teragih dibentuk oleh pelbagai agen dengan peranan tersendiri seperti berikut:

- Agan Penjejak Stok – menjejaki stok pada satu set pelayan agen.
- Agan Pemandu – menentukan destinasi laluan.
- Agan Pengurus Direktori – menjana direktori laluan.
- Agan Pengesan Rangkaian – menyemak kesegaran rangkaian.

- (e) Agen Broker – merancang carian stok.
- (f) Agen Penanya Stok – melaksanakan operasi ke atas pangkalan data.
- (g) Agen Pemohon Stok – membuat permohonan stok.
- (h) Agen Penjana Laporan – menjana laporan carian.
- (i) Agen Penemu Penyelesaian – membuat keputusan carian.

Keseluruhan seni bina sistem pengurusan inventori teragih setelah ia dipalam masuk ke dalam seni bina BDI-MU adalah seperti pada Rajah 8.14. Ia disokong oleh Agen Domain, Agen Pelayan dan Agen Konfigurasi yang berperanan menyediakan infrastruktur kepada agen. Agen Antara Muka Carian dan Agen Antara Muka Permohonan pula berperanan untuk berinteraksi dengan pengguna. Selain terdapat sistem pengurusan inventori lokal dan pangkalan data inventori.



Rajah 8.14 : Seni bina sistem pengurusan inventori teragih

## 8.6 Ringkasan

Seni bina BDI-MU mempunyai struktur dalaman yang tersusun secara hirarki dengan senario, proses dan implementasi yang berlainan serta disokong oleh komponen luaran seperti domain, pelayan dan persekitaran. Bagi menyelesaikan variasi ini, ia perlu dihurai dari pelbagai pandangan yang berbeza dan dibentuk oleh pelbagai corak yang berbeza. Di dalam bab ini, model pandangan “4+1” telah digunakan untuk menghuraikan seni bina BDI-MU dari aspek lojikal, proses, fizikal, pembangunan dan senario. Corak seni bina heterogen yang menggabungkan pelbagai corak seni bina hibrid pula telah digunakan untuk menggambarkan keseluruhan seni bina BDI-MU dan komponennya. Hasilnya, satu seni bina agen mobil pintar yang terbuka telah diketengahkan. Ianya mempunyai ciri-ciri generik, fleksibel dan boleh diguna semula. Huraian terperinci seni bina BDI-MU turut dibincangkan dalam Bab IX sehingga Bab XII seperti berikut:

- (a) Bab IX – menghuraikan operasi dan hubungan komponen-komponen di dalam pakej BDI yang digunakan untuk membentuk struktur intra agen BDI-MU menggunakan seni bina BDI dan autonomi.
- (b) Bab X – menghuraikan operasi dan hubungan komponen-komponen di dalam pakej MobileUNITY yang digunakan untuk membentuk struktur inter agen BDI-MU menggunakan mesej dan konsep bahasa Mobile UNITY.
- (c) Bab XI – membentangkan spesifikasi formal operasi dan hubungan komponen-komponen yang membentuk struktur intra dan inter seni bina BDI-MU menggunakan Z.
- (d) Bab XII – membentangkan pembentukan komponen-komponen seni bina BDI-MU dalam alatan pembangunan BDI-MU kit menggunakan Java.

Seni bina BDI-MU yang dihasilkan ini telah diimplementasi ke atas sistem pengurusan inventori teragih seperti yang telah dihurai pada Seksyen 8.5. Selain itu, ia juga telah digunakan untuk membangunkan pelbagai jenis sistem agen seperti yang disenaraikan pada Bab XII.

## **BAB IX**

### **STRUKTUR INTRA AGEN BDI-MU MENGGUNAKAN SENI BINA KEPERCAYAAN-KEINGINAN-NIAT DAN AUTONOMI**

#### **9.1 Pengenalan**

Percambahan seni bina berasaskan taakulan praktikal untuk agen telah berkembang bermula pertengahan 1980-an (Wooldridge dan Jennings, 1994). Contoh seni bina agen berasaskan taakulan praktikal adalah GRATE (Jennings, 1993), BDI (Rao dan Georgeff, 1998), HOMER (Vere dan Bickmore, 1990) dan IRMA (Bratman et al., 1988). Seni bina BDI adalah seni bina berasaskan taakulan praktikal yang paling berjaya. Ia telah dikodkan dalam pelbagai jenis bahasa seperti LISP, C dan C++. Konsep BDI mula-mula sekali telah dibangunkan oleh Bratman (1987). Tinjauan tentang seni bina BDI telah dikaji oleh Rao dan Georgeff (1995). Autonomi adalah salah satu ciri penting kepada agen kerana ia memastikan agen bermotivasi dan tidak bersandaran. Kombinasi taakulan praktikal dan keupayaan menaakul dengan paras autonomi tertentu merupakan adunan ideal yang digunakan untuk membentuk struktur intra agen BDI-MU. Ia bagi memastikan agen BDI-MU berkemampuan membuat keputusan dan melaksanakan tindakan secara maksima.

Objektif bab ini adalah untuk menerangkan operasi dan hubungan komponen-komponen yang digunakan untuk membentuk struktur intra agen BDI-MU menggunakan seni bina BDI dan autonomi. Di dalam bab ini juga dibentangkan keputusan masa larian pentafsir BDI dan saringan autonomi. Pentafsir BDI adalah komponen utama yang akan mengawal proses taakulan agen BDI-MU. Manakala,

saringan autonomi dibuat ke atas kepercayaan, matlamat, niat dan pelan agen BDI-MU untuk meningkatkan aras autonomi.

## 9.2 Seni Bina Kepercayaan-Keinginan-Niat

Seni bina BDI telah diimplementasi oleh sistem agen di dalam pelbagai domain seperti ARCHON (Jennings et al., 1996) untuk pengurusan penghantaran elektrik dan kawalan pemecut komponen, UM-PRS (Huber dan Hadley, 1997) untuk pengurusan rangkaian pengangkutan, dMARS (Kinny dan Georgeff, 1997) untuk simulasi pertempuran udara dan penerokaan sumber serta BDPAgent (Egges et al., 2001) untuk perniagaan elektronik.

Seni bina BDI seperti pada Rajah 9.1 adalah berasaskan taakulan praktikal. Ia dibentuk oleh struktur data kepercayaan,  $Bel$  ( $B \subseteq Bel$ ), keinginan,  $Des$  ( $D \subseteq Des$ ) dan niat,  $Int$  ( $I \subseteq Int$ ). Proses taakulan praktikal dilakukan oleh fungsi-fungsi berikut:

- (a) Fungsi semakan kepercayaan,  $brf: \rho(Bel) \times P \rightarrow \rho(Bel)$  – fungsi yang menentukan satu set kepercayaan yang baru daripada kepercayaan semasa dan tanggapan semasa,  $P$ .
- (b) Fungsi penjanaan pilihan,  $options: \rho(Bel) \times \rho(Int) \rightarrow \rho(Des)$  – fungsi yang memetakan satu set keinginan daripada satu set kepercayaan dan satu set niat. Fungsi tapisan,  $filter: \rho(Bel) \times \rho(Des) \times \rho(Int) \rightarrow \rho(Int)$  – fungsi yang memutuskan apakah yang perlu dilaksanakan oleh agen daripada niat sebelumnya yang masih dipegang serta kepercayaan semasa dan keinginan semasa.
- (d) Fungsi perlakuan,  $execute: \rho(Int) \rightarrow A$  – fungsi yang memulangkan mana-mana niat yang akan dilaksanakan oleh agen iaitu niat yang berpadanan dengan tindakan,  $A$  yang akan dilarikan.
- (e) Fungsi tindakan,  $action: P \rightarrow A$  – fungsi yang membuat keputusan menentukan tindakan yang akan dilaksanakan oleh agen. Ia ditakrifkan oleh kod pseudo seperti di bawah:

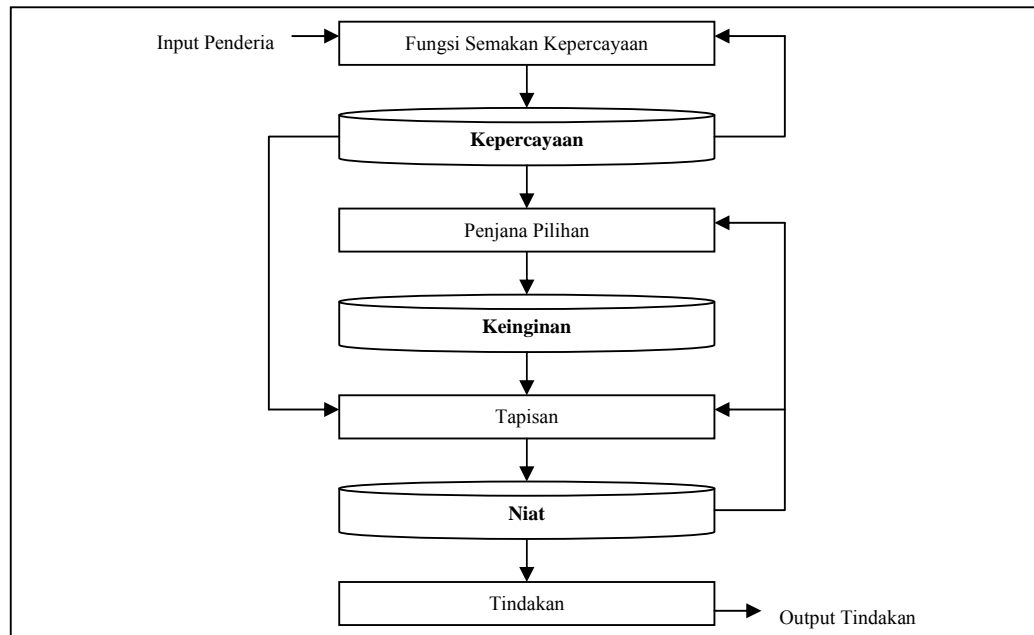
*function action (p) returns tindakan inputs: p, tanggapan  
local variables: B, kepercayaan*



```

D, keinginan
I, niat
B ← brf(B, p)
D ← options(D, I)
I ← filter(B, D, I)
return execute(I)
end

```



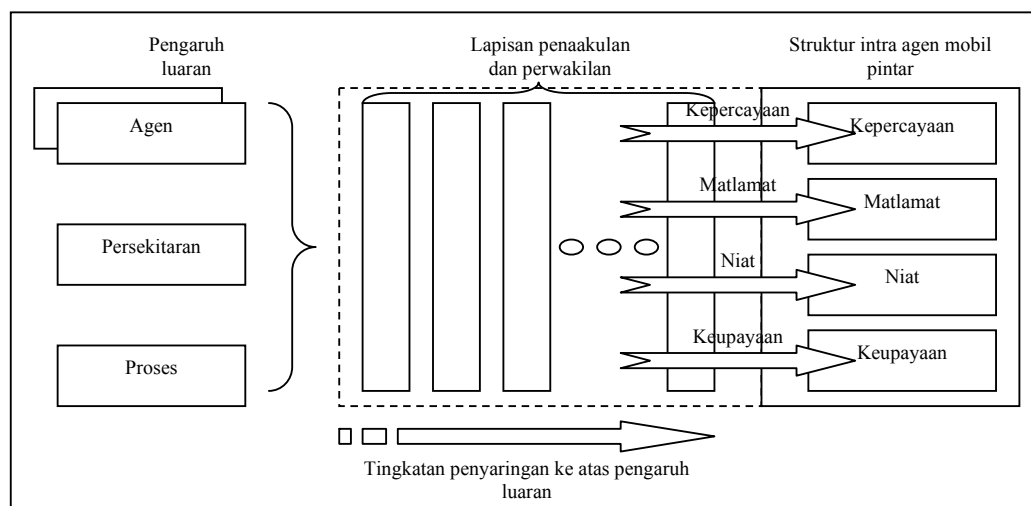
**Rajah 9.1 : Seni bina kepercayaan-keinginan-niat**

### 9.3 Autonomi

Luck dan d’Inverno (1995) mengaitkan autonomi dengan kelakuan terarah matlamat iaitu ciri yang sangat penting kepada agen berautonomi. Matlamat bersangkut-paut dengan terbitan motivasi. Motivasi ditakrifkan sebagai hasrat atau kecenderungan yang akan memandu kepada penjanaan matlamat. Falcone dan Castelfranchi (2000) mentakrifkan autonomi sebagai sejumlah pengasingan di antara pengaruh luaran, kepercayaan dan matlamat agen. Beliau mengesyorkan tapisan berlapis ke atas autonomi matlamat, di mana pengaruh luaran mesti dituras menerusi kepercayaan sebelum matlamat agen diubah suai. Oleh kerana itu, autonomi mesti

ditentukan berdasarkan pengaruh luaran agen dan tidak boleh ditakrif secara mudah sebagai atribut agen.

Dari perspektif seni bina BDI, autonomi bersabit dengan kepercayaan, matlamat, niat dan keupayaan agen. Autonomi kepercayaan, matlamat, niat dan keupayaan ditakrifkan sebagai sejumlah lapisan penaakulan dan perwakilan di antara pengaruh luaran dengan struktur intra agen. Setiap lapisan penaakulan dan perwakilan membantu agen memeriksa maklumat, menerima atau menolak maklumat dan melaksanakan penukaran ke atas maklumat tersebut. Contohnya, agen memproses imej yang diterima, membuat sedikit pemotongan, mencantumkannya menjadi bentuk yang baru dan melabelkannya. Saringan lapisan penaakulan dan perwakilan yang membentuk autonomi agen serta hubungannya dengan pengaruh luaran dan struktur intra agen adalah seperti pada Rajah 9.2.



**Rajah 9.2 : Saringan autonomi agen mobil pintar**

#### 9.4 Struktur Intra Agen BDI-MU

Struktur intra agen BDI-MU dibina berdasarkan seni bina BDI yang digunakan oleh PRS (Georgeff dan Lansky, 1986; Georgeff et al., 1985). Ia turut ditokok

tambah dengan konsep semantik litar berstruktur (*structured circuit semantics*) oleh Lee (1996) dan formalisasi pelan perbuatan (*act plan formalism*) oleh Wilkins dan Myers (1995). Komponen-komponen yang digunakan untuk membentuk struktur intra agen BDI-MU adalah model dunia atau kepercayaan, matlamat, pelan, struktur niat, pentafsir dan pemerhati sepertimana yang dispesifikasikan dalam Seksyen 9.4.1 sehingga Seksyen 9.4.6. Perisytiharan simbol-simbol yang digunakan adalah seperti pada Lampiran A. Manakala, sintaks, anatomi dan bentuk Backus-Naur (*Backus-Naur form*, BNF) komponen-komponen tersebut dibentangkan dalam Seksyen 9.4.7.

### 9.4.1 Model Dunia

Kepercayaan (*bel*) merupakan pernyataan (*proposition*) mudah yang dibentuk oleh label rentetan (*label*) dan senarai argumen (*argList*).

$$\begin{aligned}
 \text{const} & ::= \{ \chi \mid \chi = \text{number} \vee \chi = \text{string} \} \\
 \text{var} & ::= \{ v \mid \text{value}(v) = \text{const} \} \\
 \text{varList} & ::= (\text{var})^+, & + \text{ ialah satu atau lebih ulangan tem} \\
 \text{arg} & ::= \text{const} \mid \text{var} \\
 \text{argList} & ::= (\text{arg})^*, & * \text{ ialah sifar atau lebih ulangan tem} \\
 \text{proposition} & ::= \text{label} \bullet \text{argList} \\
 \text{bel} & ::= \text{proposition}
 \end{aligned}$$

Model dunia (*worldModel*) mengandungi kepercayaan sifar atau lebih. Keadaan awalan model dunia (*worldModel<sub>initial</sub>*) dispesifikasikan dalam fail teks (*file<sub>worldModel</sub>*) yang akan dihurai oleh agen semasa dijana.

$$\begin{aligned}
 \text{worldModel} & ::= (\text{bel})^* \\
 \text{worldModel}_{\text{initial}} & ::= \text{file}_{\text{worldModel}}
 \end{aligned}$$

### 9.4.2 Matlamat

Matlamat (*goal*) diwakili oleh jenis matlamat (*ACHIEVE*, *PERFORM* atau *MAINTAIN*), pernyataan dan nilai opsional utiliti (*utility*).

$$goal ::= (ACHIEVE \mid PERFORM \mid MAINTAIN) \bullet \\ proposition \bullet [ : utility \chi ], \quad \chi \neq string$$

Matlamat *ACHIEVE* menghuraikan keinginan agen untuk mencapai keadaan matlamat. Apabila matlamat telah dicapai agen akan berterusan memantau pencapaian matlamat tersebut. Matlamat *PERFORM* menghuraikan keinginan agen untuk melaksanakan sebahagian kelakuannya dan tidak berminat untuk mencapai matlamat tersebut. Matlamat *MAINTAIN* ialah matlamat homeostatik yang memberi indikasi bahawa matlamat mesti dicapai semula sekiranya pencapaian matlamat tidak memenuhi kehendak. Hanya matlamat *ACHIEVE* dan *MAINTAIN* sahaja melibatkan pencapaian matlamat dengan penerapan kepercayaan.

Agen tidak mempunyai senarai matlamat (*goalList*) yang berasingan daripada matlamat-matlamat yang terdapat di dalam struktur niat. Keadaan awalan senarai matlamat (*goalList<sub>initial</sub>*) dispesifikasikan dalam fail teks (*file<sub>goalList</sub>*) yang akan dihurai oleh agen semasa dijana.

$$goalList ::= \{ goal \mid goal \in intStructure, intention-of(goal) = null \} \\ goalList_{initial} ::= file_{goalList}$$

### 9.4.3 Pelan

Syarat (*condition*) ialah fungsi yang memulangkan nilai boolean apabila ditaksir. Ianya terdiri daripada fungsi hubungan (*relation*) dan senarai argumen.

$$condition ::= relation \bullet argList, \quad value(eval(condition)) = True \mid False$$

Pelan adalah satu komposisi komponen yang kompleks. Pelan terdiri daripada matlamat atau kepercayaan yang dijana. Prasyarat (*precondition*) dan syarat masa larian (*runtimecondition*) berfungsi menapis kesesuaian penggunaan pelan. Tatacara pelan (*procedure*) dibina daripada atur cara arbitari sama ada tindakan mudah (perlakuan primitif takrifan pengguna) atau struktur kompleks (perlakuan jika-sebaliknya atau gelung). Komponen badan pelan (*body*) yang merupakan tatacara primer boleh terdiri daripada tindakan submatlamat. Komponen efek (*effects*) dilarikan apabila komponen badan pelan berjaya dilaksanakan. Komponen kegagalan (*failure*) pula dilarikan apabila pelan gagal semasa perlakuan. Komponen kegagalan

tidak boleh mengandungi tindakan submatlamat. Keadaan awalan pustaka pelan ( $planLibrary_{initial}$ ) dispesifikasikan dalam fail teks ( $file_{planLibrary}$ ) yang akan dihurai oleh agen semasa dijana.

$precondition$	$::= (condition)^*$
$runtimecondition$	$::= (condition)^*$
$body$	$::= procedure$
$effects$	$::= procedure$
$failure$	$::= procedure$
$attributes$	$::= (attribute, value)^*$
$plan$	$::= (goal \mid bel) \bullet precondition \bullet runtimecondition \bullet procedure \bullet effects \bullet attributes$
$planLibrary$	$::= (plan)^*$
$planLibrary_{initial}$	$::= file_{planLibrary}$

#### 9.4.4 Struktur Niat

Niat ( $int$ ) merupakan tika pelan yang pembolehubahnya terikat kepada nilai. Pengikatan pembolehubah ( $binding$ ) menyediakan jadual carian yang memetakan pembolehubah kepada nilai yang dipegang oleh pembolehubah tersebut. Tindakan niat ( $intStack$ ) ialah matlamat yang belum mempunyai niat atau jujukan matlamat yang berpasangan dengan niat. Matlamat pertama di dalam tindakan niat adalah matlamat aras tinggi. Matlamat kedua dan seterusnya adalah submatlamat kepada matlamat aras tinggi. Struktur niat ( $intStructure$ ) pula ialah koleksi tindakan niat. Ia mewakili semua tugas layak yang sentiasa diberi pertimbangan oleh agen pada setiap masa untuk dilaksanakan.

$binding$	$::= \{ (v, \delta) \mid v \in varList, \delta \in const, value(v) = \delta \}$
$int$	$::= goal \bullet plan \bullet bel$
$intStack$	$::= goal \mid (goal \bullet int)^+, \quad goal = goal-of(int)$
$intStructure$	$::= (intStack)^+$

#### 9.4.5 Pentafsir

Pentafsir membuat proses penaakulan ( $genAPL$ ) ke atas matlamat yang belum dihasratkan, pelan yang sedia digunakan dan satu set kepercayaan untuk menjana

pelan boleh guna (*apl*). Pelan boleh guna terdiri daripada satu senarai niat yang boleh digunakan. Niat dengan utiliti tertinggi dipilih (*selectAPLElement*) dan dicadangkan (*intend*) kepada struktur niat agen. Pentafsir akan menyisih niat (*sortInt*) berdasarkan nilai utiliti niat. Nilai utiliti ini berubah pada setiap kitaran. Ia disebabkan oleh perubahan pada model dunia.

$$\begin{aligned}
 apl & ::= (int)^* \\
 genAPL & : worldModel \times goalList \times planLibrary \rightarrow apl \\
 selectAPLElement & : apl \rightarrow int, \quad int = highest-utility(apl) \\
 intend & : intStructure \times int \rightarrow intStructure \\
 sortInt & : worldModel \times intStructure \rightarrow intStructure
 \end{aligned}$$

Seni bina BDI tidak mengekang maklumat berasaskan komunikasi (*comm*) dan tanggapan (*percepts*). Maklumat relevan melalui komunikasi dan tanggapan adalah paras perwakilan kepercayaan, matlamat, pelan dan niat individu. Pelan membenarkan agen menaakul terhadap maklumat komunikasi dan tanggapan serta kepercayaan dan matlamat agen untuk mengemaskini kepercayaan dan matlamat sedia ada. Perlakuan pelan boleh memberi kesan ke atas penghuraian dan penggabungan pelan ke dalam pustaka pelan. Taakulan pelan aras meta (*execMLP*) dilakukan jika terdapat lebih daripada satu hasil dijana oleh *apl*. Proses penaakulan dilaksanakan secara lelaran untuk memilih elemen *apl* yang akan dicadangkan kepada struktur niat. Ia berakhir jika tiada lagi elemen alternatif dijana oleh *apl* dan ia mestilah bukan tindanan kosong.

$$\begin{aligned}
 execPlan & : comm \times percepts \times worldModel \times goalList \rightarrow \\
 & \quad worldModel \times goalList \times planLibrary \\
 execMLP & : comm \times percepts \times worldModel \times goalList \times apl \rightarrow \\
 & \quad worldModel \times goalList \times intStructure \times planLibrary
 \end{aligned}$$

#### 9.4.6 Pemerhati

Perlakuan pemerhati (*execObserver*) dilaksanakan setiap kali gelung pentafsir dilarikan. Ia merupakan prosidur yang mengamati peristiwa tidak segerak. Pemerhati

hanya terdiri daripada komponen badan pelan sahaja. Ia berpunca daripada hasil komunikasi dan tanggapan. Pemerhati boleh mencapai model dunia agen dan senarai pelan boleh guna agen. Operasi ke atas model dunia dan penjanaan matlamat aras tinggi boleh dilaksanakan oleh pemerhati. Sebagaimana pelan aras meta, pemerhati boleh mencapai senarai pelan boleh guna di dalam model dunia dan mencadangkannya kepada struktur niat.

$$\text{execObserver} : \text{comm} \times \text{percepts} \times \text{worldModel} \times \text{apl} \rightarrow \text{worldModel} \times \text{goalList} \times \text{intStructure} \times \text{planLibrary}$$

#### 9.4.7 Sintaks, Anatomi dan Bentuk Backus-Naur

Sintaks dan anatomi untuk mentakrif matlamat, pelan, model dunia dan pemerhati agen yang akan dihurai oleh pentafsir adalah seperti berikut:

```

GOAL: jenis_matlamat nama_matlamat parameter1, ..., parameterN
      < : UTILITY pernyataan >;
PLAN: {
      GOAL: [spesifikasi matlamat]
      |
      CONCLUDE: [hubungan model dunia]
      NAME: [rentetan]
      BODY: [prosidur]
      < DOCUMENTATION: [rentetan] >
      < PRECONDITION: [pernyataan] >
      < CONTEXT: [pernyataan] >
      < UTILITY: [pernyataan numerik] >
      < FAILURE: [prosidur tanpa submatlamat] >
      < EFFECTS: [prosidur tanpa submatlamat] >
      < ATTRIBUTES: [rentetan] > }
FACTS: nama_hubungan argumen1 argumen2 ... argumenN;
OBSERVER: {
      [prosidur tanpa submatlamat] }

```

Manakala, nahu BNF yang digunakan untuk menerangkan sintaks bahasa pengaturcaraan yang telah dibangunkan untuk mentakrif model dunia, matlamat, pelan, struktur niat, pentafsir dan pemerhati adalah seperti pada Lampiran B.

## 9.5 Autonomi di dalam Struktur Intra Agen BDI-MU

Aras autonomi ( $\lambda$ ) untuk niat ( $\lambda_{int}$ ), keupayaan ( $\lambda_{cap}$ ), kepercayaan ( $\lambda_{bel}$ ) dan matlamat ( $\lambda_{goal}$ ) dikira berdasarkan sejumlah penukaran berprosudur dan berperwakilan di antara pengaruh luaran dan struktur intra agen BDI-MU. Bilangan penukaran terendah diambil sebagai nilai  $\lambda$  menggunakan fungsi minimum ( $min$ ). Sekiranya nilai  $\lambda$  adalah sifar, maka pengaruh luaran mempunyai kawalan sepenuhnya ke atas struktur intra agen BDI-MU. Sebaliknya, ia menandakan aras autonomi agen yang semakin tinggi.

### 9.5.1 Autonomi Niat, Keupayaan, Kepercayaan dan Matlamat

Struktur niat boleh diubah suai semasa penaakulan lalai pentafsir, semasa perlakuan pelan aras meta atau semasa perlakuan pemerhati.

Bagi kes pertama:

$$\lambda_{int} = 5$$

$$[worldModel \times goalList \times planLibrary \rightarrow genAPL() \rightarrow apl \rightarrow selectAPLElement() \rightarrow int \rightarrow intend() \rightarrow intStructure]$$

Bagi kes kedua dan ketiga:

$$\lambda_{int} = 3$$

$$[worldModel \times goalList \times planLibrary \rightarrow genAPL() \rightarrow apl \rightarrow execX() \rightarrow intStructure, \text{ di mana } execX() \text{ mewakili } execMLP() \text{ atau } execObserver()]$$

$$\therefore, \lambda_{int} = min(5, 3, 3)$$

$$= 3$$

Keupayaan agen datang daripada fail yang dispesifikasikan semasa penjanaan agen (ia tidak diambil kira kerana dijana secara dalaman), semasa perlakuan pelan, semasa perlakuan pelan aras meta atau semasa perlakuan pemerhati.

Bagi kes kedua dan ketiga:

$$\lambda_{cap} = 9$$

$$[worldModel \times goalList \times planLibrary \rightarrow genAPL() \rightarrow apl \rightarrow selectAPLElement() \rightarrow int \rightarrow intend() \rightarrow intStructure \rightarrow sortInt() \rightarrow intStructure \rightarrow execX() \rightarrow planLibrary, \text{ di mana } execX() \text{ mewakili } execPlan() \text{ atau } execMLP()]$$

Manakala, penjanaan pemerhati hanya menghasilkan satu aras autonomi,  $\lambda_{cap} = 1$ .



$$\begin{aligned}\therefore, \lambda_{cap} &= \min(9, 9, 1) \\ &= 1\end{aligned}$$

Model dunia dan matlamat agen terbit daripada senario yang sama dengan keupayaan agen.

$$\begin{aligned}\therefore, \lambda_{bel} &= \min(9, 9, 1) \\ &= 1 \\ \lambda_{goal} &= \min(9, 9, 1) \\ &= 1\end{aligned}$$

### 9.5.2 Autonomi Fleksibel

Aras autonomi agen boleh ditingkatkan melalui penstrukturan, reka bentuk dan pembangunan prosidur pemerhati secara teliti. Ini adalah kerana pemerhati memberikan  $\lambda_{cap} = \lambda_{bel} = \lambda_{goal} = 1$ .  $\lambda_{cap}$  boleh ditingkatkan melalui pemeriksaan pelan bagi memastikan pelan tidak mengandungi tindakan merbahaya sebelum ditindan ke dalam pustaka pelan.  $\lambda_{bel}$  boleh ditingkatkan dengan menambah taakulan untuk menyemak konsistensi kepercayaan dan kredibiliti sumber kepercayaan sebelum menambahnya ke dalam model dunia.  $\lambda_{goal}$  pula boleh ditingkatkan dengan menambah lapisan taakulan bagi membolehkan agen menyemak pertalian matlamat.

## 9.6 Keputusan Komputasi

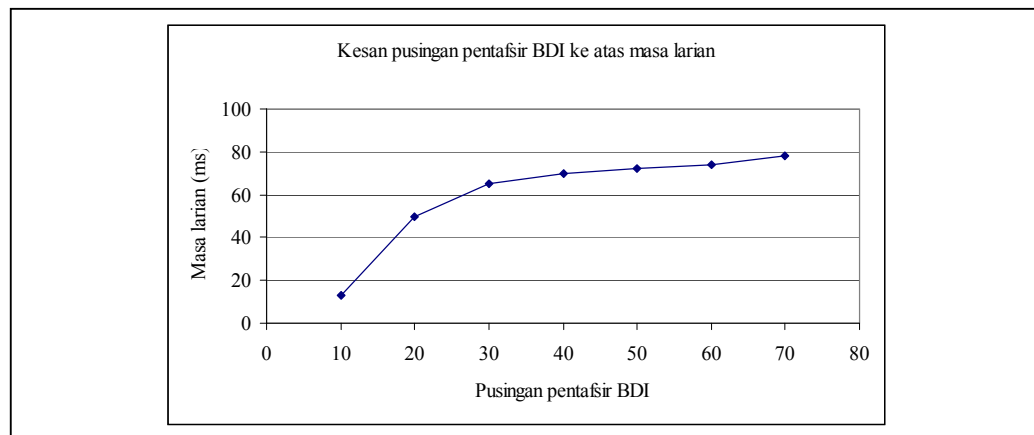
Komponen-komponen yang dibangunkan di atas telah diuji lariannya menggunakan data-data sistem pengurusan inventori teragih. Maklumat bagi data-data sistem pengurusan inventori teragih adalah seperti pada Jadual 9.1. Dua kes eksperimen telah digunakan untuk menilai kebolehlaksanaan struktur intra agen BDI-MU di dalam persekitaran teragih dan heterogen, iaitu:

- (a) Masa larian bagi pusingan pentafsir BDI (Rajah 9.3).
- (b) Masa larian bagi saringan autonomi (Rajah 9.4).

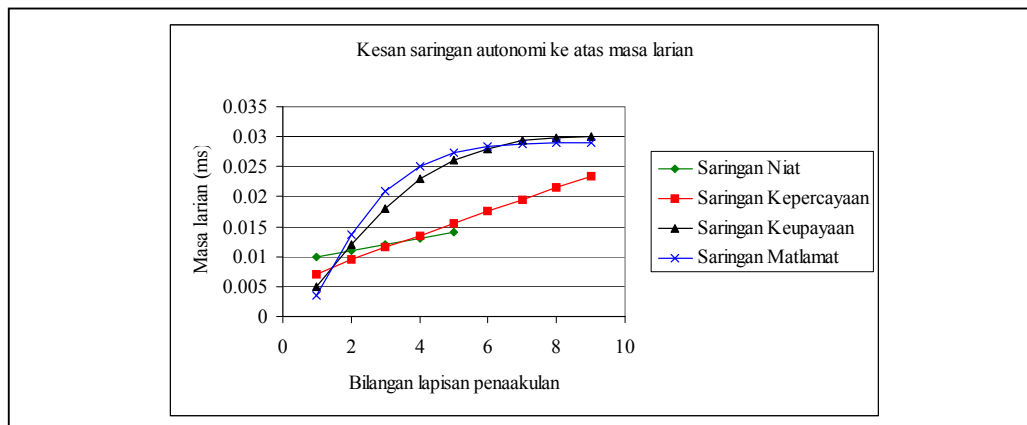
Pengukuran dijalankan pada persekitaran pengkomputeran dua buah komputer yang terangkai melalui intranet menggunakan protokol rangkaian TCP/IP. Spesifikasi komputer adalah Pentium III 500 MHz, 256 MB RAM, 10/100 MBps NIC, JDK 1.3.1 dengan sebuah komputer dipasang Windows XP dan sebuah lagi dipasang Red Hat Linux.

**Jadual 9.1 : Maklumat bagi data-data sistem pengurusan inventori teragih**

No	Item	Data
1.	Bilangan kepercayaan	7,750
2.	Bilangan pelan	90
3.	Bilangan matlamat	20
4.	Bilangan penjanaian niat ( <i>apl</i> )	90
5.	Bilangan matlamat tercapai	20
6.	Bilangan pusingan pentafsir	70
7.	Bilangan lapisan penaakulan matlamat	9
8.	Bilangan lapisan penaakulan kepercayaan	9
9.	Bilangan lapisan penaakulan keupayaan	9
10.	Bilangan lapisan penaakulan niat	5



**Rajah 9.3 : Graf kesan pusingan pentafsir BDI ke atas masa larian**



**Rajah 9.4 : Graf kesan saringan autonomi ke atas masa larian**

Sebagaimana yang ditunjukkan pada Rajah 9.3 dan Rajah 9.4, keputusan ini menunjukkan walaupun bilangan pusingan pentafsir BDI dan lapisan penaakulan matlamat, kepercayaan, keupayaan dan niat meningkat tetapi kenaikan masa larian agen BDI-MU adalah munasabah. Ini menandakan bahawa penambahan bilangan pusingan pentafsir BDI untuk membolehkan agen BDI-MU membuat taakulan praktikal dan penambahan lapisan taakulan untuk meningkatkan autonomi agen BDI-MU tidak menimbulkan kesan luar jangka yang akan mengurangkan prestasi larian agen BDI-MU.

## 9.7 Ringkasan

Seni bina BDI dan autonomi telah digabungkan untuk membentuk komponen-komponen model dunia, matlamat, pelan, struktur niat, pentafsir dan pemerhati yang digunakan untuk membangunkan struktur intra agen BDI-MU. Tujuannya adalah untuk membolehkan agen BDI-MU membuat taakulan praktikal secara berautonomi. Proses taakulan dibuat melalui tafsiran struktur data kepercayaan, keinginan dan niat yang dilakukan oleh pentafsir BDI menerusi beberapa lapisan taakulan yang bertujuan untuk meningkatkan aras autonomi. Spesifikasi operasi dan hubungan di antara komponen-komponen tersebut telah dihuraikan dalam Seksyen 9.4. Manakala, pengukuran autonomi dan kaedah untuk

meningkatkan fleksibiliti autonomi bagi menambahkan kebolehpayaan agen BDI-MU dihuraikan dalam Seksyen 9.5. Hasil eksperimen yang dibentangkan dalam Seksyen 9.6 pula menunjukkan perlaksanaan pentafsir BDI dan saringan autonomi yang munasabah apabila agen BDI-MU diimplementasi dalam sistem dunia nyata. Spesifikasi formal operasi dan hubungan di antara komponen-komponen model dunia, matlamat, pelan, struktur niat, pentafsir dan pemerhati dibentangkan dalam Bab XI. Di dalam seni bina BDI-MU, komponen-komponen tersebut dikelompokkan dalam pakej BDI. Huraian kelas-kelas yang mewakili komponen-komponen tersebut termasuk penggunaannya ke atas sistem pengurusan inventori teragih dibentangkan dalam Bab XII.

## **BAB X**

### **STRUKTUR INTER AGEN BDI-MU MENGGUNAKAN MESEJ DAN KONSEP BAHASA MOBILE UNITY**

#### **10.1 Pengenalan**

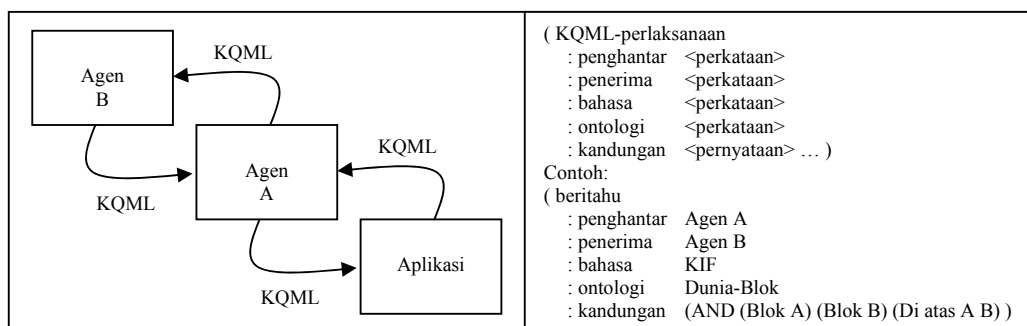
Agen mobil pintar telah diimplementasi secara meluas dalam aplikasi teragih dan sistem pelbagai agen seperti Bee-gent (Kawamura et al., 2000), DIA (Hattori et al., 1999) dan WSS (Kim et al., 1999). Oleh kerana itu, agen mobil pintar perlu berkomunikasi dengan entiti lain dan membuat penjelajahan di antara mesin untuk meningkatkan kemampuannya mencapai matlamat secara sempurna. Komunikasi membolehkan agen mobil pintar membuat tanggapan dan melaksanakan tindakan. Ia berlaku melalui mesej yang diterima dan mesej yang dihantar oleh agen mobil pintar. Penjelajahan pula membolehkan agen mobil pintar mencapai dan mengumpul sumber teragih. Komunikasi dan mobiliti adalah komponen yang membentuk struktur inter agen mobil pintar. Komponen ini diperlukan untuk membantu agen mobil pintar membentuk organisasi yang saling bekerjasama dan berkongsi sumber secara kolektif.

Objektif bab ini adalah untuk menerangkan operasi dan hubungan komponen-komponen yang digunakan untuk membentuk struktur inter agen BDI-MU menggunakan mesej dan konsep bahasa Mobile UNITY. Mesej digunakan untuk menyelesaikan masalah komunikasi teragih. Konsep bahasa Mobile UNITY pula digunakan untuk menyelesaikan masalah simulasi teragih. Di dalam bab ini juga dibentangkan keputusan masa larian komunikasi dan mobiliti agen BDI-MU.

## 10.2 Mesej

Mesej (Huhns dan Singh, 1997) adalah struktur data yang merupakan unit asas komunikasi. Mesej juga adalah perwakilan kelakuan agen yang dihantar kepada agen lain. Mesej telah digunakan dalam pelbagai domain yang melibatkan komunikasi dan perkongsian (perkongsian pengetahuan atau perkongsian tugas). Contohnya, KIF (Singh et al., 1995), KQML (Finin et al., 1994), CORBA (Siegel, 1998) dan HL7 (Dolin et al., 2000).

Terdapat dua jenis mesej iaitu permintaan dan pertanyaan. Setiap agen sama ada pasif atau aktif mempunyai keupayaan untuk membuat permintaan dan pertanyaan melalui penerimaan atau penghantaran mesej. Mesej diterima atau dihantar berdasarkan protokol komunikasi yang tertentu. Protokol komunikasi adalah peraturan yang menetapkan bagaimana mesej dihebah kepada agen lain melalui persekitaran. Terdapat tiga aras protokol komunikasi iaitu protokol aras rendah, protokol aras pertengahan dan protokol aras tinggi. Protokol aras rendah menghuraikan kaedah untuk penyambungan. Protokol aras pertengahan menghuraikan format, sintaks dan maklumat yang dipindahkan. Protokol aras tinggi pula menghuraikan makna dan semantik maklumat. Terdapat dua jenis protokol komunikasi iaitu binari dan n-ari. Protokol binari melibatkan satu penghantar dan satu penerima. Protokol n-ari pula melibatkan satu penghantar dan berbilang penerima. Protokol komunikasi diuraikan oleh struktur data yang mengandungi lima medan iaitu penghantar, penerima, bahasa protokol komunikasi, fungsi menyahkod dan mengekod serta tindakan yang akan dilaksanakan oleh penerima. Contoh protokol komunikasi KQML adalah seperti pada Rajah 10.1.



**Rajah 10.1 : Protokol komunikasi KQML**

### 10.3 Mobile UNITY

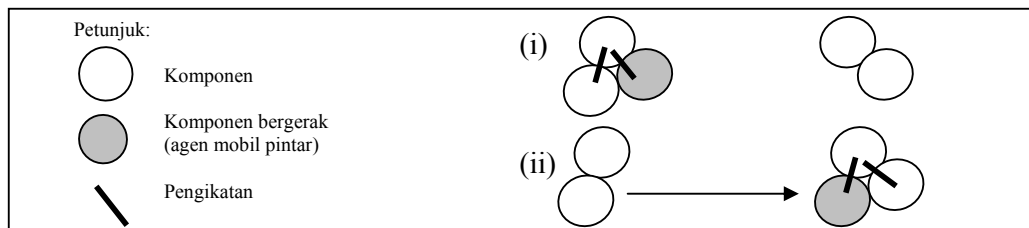
Mobile UNITY (McCann dan Roman, 1998) merupakan bahasa berasaskan keadaan yang dikembangkan daripada bahasa UNITY (Chandy dan Misra, 1986). Struktur bahasa Mobile UNITY menyokong teknologi agen mobil pintar melalui kaedah seperti berikut:

- (a) Atur cara – merupakan unit yang berupaya bergerak dengan takrifan perlakuan dan atribut yang tertentu. Berdasarkan kemampuannya untuk bergerak, setiap atur cara mempunyai pembolehubah yang menyimpan nilai lokasi. Pengguna akan menulis atur cara yang boleh digerakkan di sepanjang rangkaian. Setiap kali berlaku pergerakan, nilai pembolehubah lokasi akan diumpukkan semula.
- (b) Seksyen komponen – mentakrifkan tika atur cara yang akan wujud di dalam sistem. Pembinaan komponen baru secara dinamik tidak dibenarkan.
- (c) Seksyen interaksi – mengandungi pernyataan yang mentakrifkan interaksi dan pergerakan komponen.

Bagi membolehkan agen mobil pintar di hos pelanggan mencapai sebahagian sumber yang terdapat di hos pelayan, Mobile UNITY membenarkan hos pelanggan menghantar keseluruhan komponen agen mobil pintar yang sedang dilarikan ke hos pelayan. Ia termasuklah keadaan data dan keadaan kawalan agen mobil pintar. Apabila agen mobil pintar sampai di hos pelayan, ia akan menyambung semula perlakuannya. Situasi ini membolehkan agen mobil pintar melawati beberapa hos yang lain bagi pihak hos sumbernya tanpa berinteraksi dengan hos sumber. Rajah 10.2 menggambarkan pergerakan agen mobil pintar. Setiap agen mobil pintar mengandungi kod dan keadaan masa larian. Kod masa larian merupakan spesifikasi yang menentukan kelakuan masa larian agen mobil pintar. Keadaan masa larian pula terdiri daripada:

- (a) Keadaan data – data tempatan atau persendirian yang mewakili ciri-ciri intrinsik agen mobil pintar.
- (b) Keadaan kawalan – semasa agen mobil pintar dilarikan, keadaan kawalan memegang maklumat berkaitan dengan status perlakuannya.

- (c) Pengikatan – nama yang menuding kepada komponen lain yang sama ada terletak di hos yang sama atau berlainan yang disimpan oleh agen mobil pintar.



**Rajah 10.2 : Pergerakan agen mobil pintar berasaskan konsep bahasa Mobile UNITY**

Contoh sistem agen yang dibangunkan berasaskan konsep bahasa Mobile UNITY adalah seperti VirtualCart (Roman et al., 1997) untuk pengurusan muatan dan punggahan kargo, Mobisaic (Voelker dan Bershad, 1998) untuk penyemakan pengimbasan maklumat internet dan LIME (Murphy et al., 2002) untuk pengurusan halaaan IP bagi hos mobil atau tanpa wayar. Model formal bahasa Mobile UNITY pula telah dibangunkan oleh Mascolo (2001).

## 10.4 Huraian Masalah Struktur Inter Agen BDI-MU

### 10.4.1 Masalah Komunikasi Teragih

Masalah komunikasi teragih adalah masalah yang berkaitan dengan penghantaran mesej. Penghantaran mesej adalah paradigma komunikasi di mana agen berinteraksi dengan menghantar mesej eksplisit kepada agen lain (Milojicic et al., 2002). Semasa penghantaran mesej, agen BDI-MU tidak bergerak dari posisinya. Hanya mesej yang dihantar atau diterima daripada agen lain. Muatan rangkaian,  $B_{mesej}$  bagi penghantaran mesej daripada lokasi  $L_1$  ke lokasi  $L_2$  adalah saiz permintaan,  $B_{minta}$  dicampur dengan saiz jawapan,  $B_{jawab}$ :

$$B_{mesej}(L_1, L_2, B_{minta}, B_{jawab}) = \begin{cases} 0 & \text{jika } L_1 = L_2, \\ B_{minta} + B_{jawab} & \text{jika } L_1 \neq L_2. \end{cases}$$

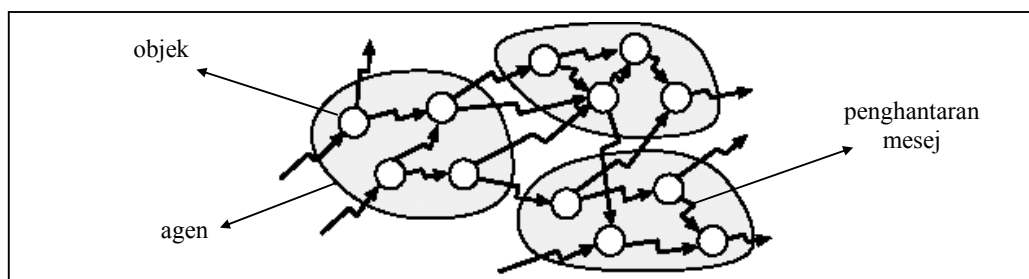


Masa pemindahan,  $T_{mesej}$  mengandungi masa untuk menghantar permintaan dan menerima jawapan:

$$T_{mesej}(L_1, L_2, B_{minta}, B_{jawab}) = 2\delta(L_1, L_2) + \left[ \frac{1}{\tau(L_1, L_2)} \right] B_{mesej}(L_1, L_2, B_{minta}, B_{jawab})$$

di mana  $\delta(L_1, L_2)$  adalah lengah rangkaian dan  $\tau(L_1, L_2)$  adalah trupert di antara lokasi  $L_1$  dengan  $L_2$ .

Satu lagi masalah komunikasi teragih adalah orientasi objek teragih (Fujita dan Akagi, 1999). Objek yang mewakili entiti dan atribut boleh disembunyikan dan teragih melalui agen seperti pada Rajah 10.3.



**Rajah 10.3 : Orientasi objek teragih**

#### 10.4.2 Masalah Simulasi Teragih

Masalah simulasi teragih (Chandy dan Misra, 1979) adalah contoh masalah teragih berasaskan aspek temporal. Idea asasnya adalah untuk menyimulasikan kelakuan sistem fizikal seperti litar elektronik dalam nod-nod komputasi rangkaian yang berkomunikasi secara tidak segerak dan ketiadaan ingatan perkongsian global. Masalah simulasi teragih boleh diformalisasikan seperti berikut:

- Sistem dimodelkan oleh satu set  $N$  proses,  $0, 1, \dots, (N - 1)$ .
- Setiap proses  $P(i)$  mempunyai sekutuan dengan pemasa tempatan,  $t(i)$ .
- Masa maya global (*global virtual time*, GVT),  $T$  ditakrifkan sebagai pemasa tempatan terkecil daripada semua pemasa tempatan.

$$T = \langle \min i : 0 \leq i \leq N - 1 :: t(i) \rangle^1, \quad \langle \dots \rangle^1 \text{ ialah satu nilai pemasa tempatan terkecil}$$

- (d) Kriteria penjadualan yang digunakan oleh proses untuk mengemaskini pemasanya dimuatkan dalam takrifan fungsi  $f$  dan  $g$  yang diindeks oleh pengecam proses yang spesifik:

$$f_i(t(i), T, a) = \begin{cases} t(i) & \text{jika } t(i) > T, \\ g_i(t(i), a) & \text{jika } t(i) = T, \\ \perp & \text{jika } t(i) < T. \end{cases}$$

$$g_i(t(i), a) > t(i)$$

di mana  $a$  mengidentifikasikan mod simulasi. Contohnya, parameter bagi komponen fizikal.

Takrifan di atas memberikan masalah simulasi teragih keperluan seperti berikut:

- (a) Pemas tempatan tidak berubah jika ia berada di hadapan GVT.
- (b) Pemas tempatan, jika dibenarkan, hanya boleh bertambah, iaitu tindakan sentiasa dijadualkan pada masa hadapan.
- (c) Nilai pemas tempatan tidak boleh berada di belakang GVT. Bagi kes tersebut fungsi  $f$  didefinisikan.

## 10.5 Penyelesaian Masalah Komunikasi Teragih Menggunakan Mesej

Komunikasi agen dicapai menggunakan mesej diskret berasaskan rentetan yang dihantar melalui rangkaian menggunakan protokol TCP/IP. Mesej diwakili oleh objek. Manipulasi mudah objek seperti membina atau menghapus tika boleh dilakukan ke atas mesej. Mesej dikodkan sebagai objek seperti berikut:

```
public class Message {
    private AgentAddress sender;
    private AgentAddress receiver;
    private String kind;
    private String content;
    private byte[] binary;
}
```

Mesej mengandungi maklumat mengenai penghantar, penerima, jenis, kandungan dan perduaan mesej seperti pada Jadual 10.1. Kandungan mesej boleh terdiri daripada agen, objek, kod sumber kelas atau fail.

**Jadual 10.1 : Maklumat bagi struktur mesej**

<b>Medan</b>	<b>Keterangan</b>
Penghantar	Agen yang menghantar mesej.
Penerima	Agen yang menerima mesej.
Jenis	Jenis mesej yang mengandungi nilai: <ul style="list-style-type: none"> <li>(a) <i>AGENT</i>: mesej mengandungi agen. Kandungan menyimpan nama agen. Perduaan menyimpan keadaan data dan keadaan kawalan agen.</li> <li>(b) <i>GET</i>: mendapatkan agen dari pelayan lain. Kandungan menyimpan nama agen. Perduaan menyimpan kod bait agen.</li> <li>(c) <i>GET_CLASS</i>: mendapatkan kod bait kelas dari pelayan lain. Kandungan menyimpan nama kelas. Perduaan menyimpan kod bait kelas.</li> <li>(d) <i>CLASS</i>: mesej mengandungi kod bait kelas. Kandungan menyimpan nama kelas. Perduaan menyimpan kod bait kelas.</li> <li>(e) <i>FILE</i>: mesej mengandungi fail. Kandungan menyimpan nama fail. Perduaan menyimpan kod perduaan fail.</li> <li>(f) <i>MESSAGE</i>: menghantar mesej arbitrari. Kandungan dan perduaan boleh digunakan secara bebas.</li> </ul>
Kandungan	Rentetan yang menyimpan nama kandungan.
Perduaan	Menyimpan kod perduaan kandungan.

## **10.6 Penyelesaian Masalah Simulasi Teragih Menggunakan Konsep Bahasa Mobile UNITY**

Agen BDI-MU digunakan untuk membantu hos penghantar mencapai sumber yang terletak pada satu set hos sasaran. Hos penghantar memindahkan agen BDI-MU yang sedang dilarikan di hosnya ke hos sasaran bersama dengan keadaan data dan keadaan kawalan agen BDI-MU. Pengikatan ke atas sumber di hos penghantar dilepaskan dan digantikan dengan pengikatan baru ke atas sumber di hos sasaran. Agen BDI-MU apabila sampai di hos sasaran akan menyambung semula perlakuannya. Langkah ini dilaksanakan berulang kali bagi membolehkan agen BDI-MU melawat beberapa hos sasaran bagi pihak hos penghantar tanpa memerlukan interaksi dengan hos penghantar. Penyelesaian masalah simulasi teragih menggunakan konsep bahasa Mobile UNITY adalah seperti pada Rajah 10.4. Perisytiharan simbol-simbol bahasa Mobile UNITY yang digunakan adalah seperti pada Lampiran C.

```

System STAgenBDI-MU
  Program P(i) at  $\lambda$ 
    declare
       $x, t, a : \text{integer} \sqcap T : \text{integer} \cup \{\perp\}$ 
    initially
       $x, t, a = 0, 0, \bar{a} \sqcap T = \perp \sqcap \lambda = \text{Location}(i)$ 
    assign
       $x := t$ 
       $\sqcap t, T := f_i(t, T, a), \perp$                                 if  $\text{def}(T)$ 
    end
  Program Agen at  $\lambda$ 
    declare
       $t, T : \text{integer} \cup \{\perp\} \sqcap \tau : \text{array of integer}$ 
       $\sqcap \text{pos} : \text{targetAddress}$ 
    initially
       $t, T = \perp, \perp \sqcap \langle \parallel j :: \tau(j) = 0 \rangle$ 
       $\sqcap \lambda = \text{Location}(\text{pos})$ 
    assign
       $\tau(\text{pos}), t := t, \perp$                                 if  $\text{def}(t)$ 
       $\sqcap T, \text{pos}, \lambda := \langle \min k :: \tau(k) \rangle, \text{pos} + 1 \bmod N,$ 
       $\text{Location}(\text{pos} + 1 \bmod N)$                                 if  $\neg \text{def}(t)$ 
    end
  Components
     $\langle \sqcap i :: P(i) \rangle \sqcap \text{Agen}$ 
  Interactions
     $P(i).T \leftarrow \text{Agen}.T$                                 when  $P(i).\lambda = \text{Agen}.\lambda$ 
    disengage  $P(i).T, \perp$ 
     $\sqcap \text{Agen}.t \leftarrow P(i).t$                                 when  $P(i).\lambda = \text{Agen}.\lambda$ 
    engage  $P(i).t$ 
  end
end

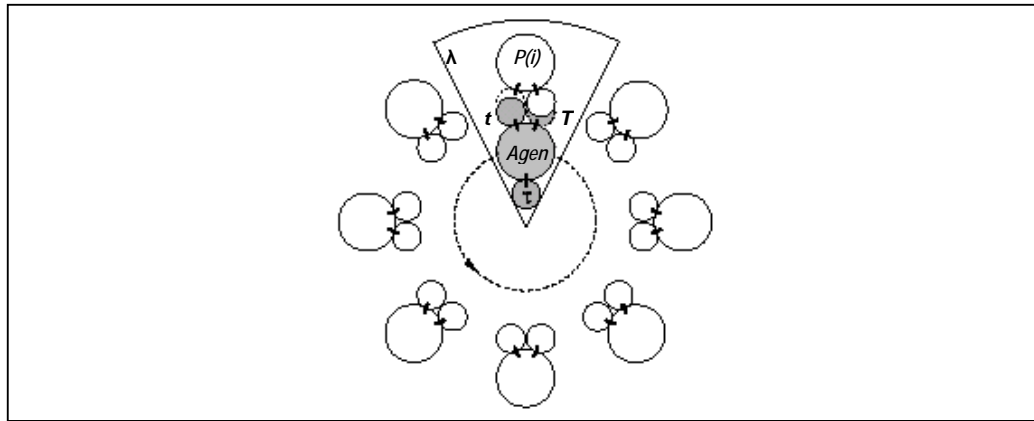
```

**Rajah 10.4 : Penyelesaian masalah simulasi teragih menggunakan konsep bahasa Mobile UNITY**

$P(i)$  berfungsi sebagai kod yang dilarikan oleh setiap hos sasaran dengan parameter  $i$  yang mewakili nombor hos sasaran. *Agen* pula berfungsi sebagai agen BDI-MU yang sedang dilarikan oleh hos penghantar. Seksyen komponen menjana satu hos sasaran bagi setiap nilai  $i, i = 1, 2, \dots, n$  dan satu tika *Agen*. *Program*  $\langle \text{name} \rangle$  at  $\lambda$  melambangkan bahawa setiap atur cara wujud di lokasi fizikal atau logikal spesifik yang ditandakan oleh pembolehubah  $\lambda$ .

Setiap atur cara mengandungi seksyen isytihar, permulaan dan umpukkan. Seksyen isytihar berfungsi untuk mengumumkan pembolehubah yang digunakan dalam atur cara. Seksyen permulaan memberikan posisi awalan kepada komponen.

Seksyen umpukkan pula berfungsi untuk membaca atau mengubah posisi komponen. Lokasi permulaan bagi setiap hos sasaran adalah alamat hos sasaran yang dipulangkan oleh fungsi  $Location()$ , di mana alamat tersebut tidak berubah. Sebaliknya, lokasi permulaan bagi *Agen* adalah lokasi arbitrari yang dipilih daripada hos sasaran sedia ada. Lokasi *Agen* berubah secara eksplisit semasa perlakuan. Tujuannya untuk membenarkan hos sasaran dijelajahi oleh *Agen* secara pusingan robin seperti pada Rajah 10.5.



**Rajah 10.5 : *Agen* berupaya mengubah lokasi bagi membolehkannya melawati setiap hos sasaran secara pusingan robin**

*Agen* membawa bersamanya keadaan global semua pemasa tempatan. Keadaan global dikemaskini dengan nilai pemasa bagi hos sasaran  $P(i)$  ketika kedudukan kedua-dua komponen ditentukan semula. *Agen* juga bertanggungjawab mengira anggaran GVT yang baru. Pengemaskinian keadaan global dan komputasi GVT berlaku dalam sebarang aturan. Ini adalah kerana *Agen* tidak boleh bertolak sehingga kedua-dua nilai ini dikemaskini. Tindakan tersebut diselaraskan oleh perkongsian pembolehkan sementara yang ditakrifkan oleh seksyen interaksi. Anggaran GVT dan pemasa tempatan yang dipunyai oleh hos sasaran  $P(i)$  dikongsi bersama dengan analoginya dalam *Agen* selagi mana kedudukan kedua-dua komponen saling ditentukan semula.

Pernyataan terakhir dalam *Agen* akan mengubah secara eksplisit lokasi  $\lambda$  bagi komponen. Ia secara langsung melarikan klausa *engage* bagi interaksi yang kedua. Semasa bertolak, klausa *disengage* menjamin bahawa hos sasaran memegang

anggaran GVT yang baru dan *Agen* pula mensetkan nilai GVT semasanya. Semasa sampai di lokasi berikutnya, klausa *engage* memastikan nilai yang dikomputasi oleh *Agen* dalam hos sasaran sebelumnya disampaikan kepada  $P(i)$  tatkala kedudukan kedua-dua komponen ditentukan semula. Secara umumnya, nilai GVT berubah berdasarkan komputasi semula pemasa terkecil. Seandainya pemasa ini tidak berubah sejak dari lawatan terakhir, nilai dalam  $\tau$  kekal tidak berubah dan GVT yang dikomputasi pada hos sasaran sebelumnya masih sah serta tidak perlu dikomputasi semula. Dalam kes sebegini, pernyataan ketiga dalam *Agen* dibolehkan bagi membenarkan *Agen* bertolak tanpa menunggu perlakuan dua pernyataan yang lainnya. Perlu diberi perhatian bahawa perlepasan nilai, pergerakan dan pengikatan menghasilkan tindakan atomik. Ini bererti *Agen* tidak boleh melaksanakan sebarang tindakan ketika sedang menjelajah di sepanjang sambungan komunikasi.

## 10.7 Keputusan Komputasi

Struktur inter agen BDI-MU dibentuk oleh komponen komunikasi dan komponen mobiliti. Komponen komunikasi dibina menggunakan mesej untuk menangani masalah komunikasi teragih. Komponen mobiliti pula dibina menggunakan konsep bahasa Mobile UNITY untuk menangani masalah simulasi teragih. Kedua-dua komponen tersebut telah diuji lariannya menggunakan data-data sistem pengurusan inventori teragih. Maklumat bagi data-data sistem pengurusan inventori teragih adalah seperti pada Jadual 10.2. Kes eksperimen yang telah digunakan untuk menilai kebolehlaksanaan struktur inter agen BDI-MU di dalam persekitaran teragih dan heterogen adalah seperti berikut:

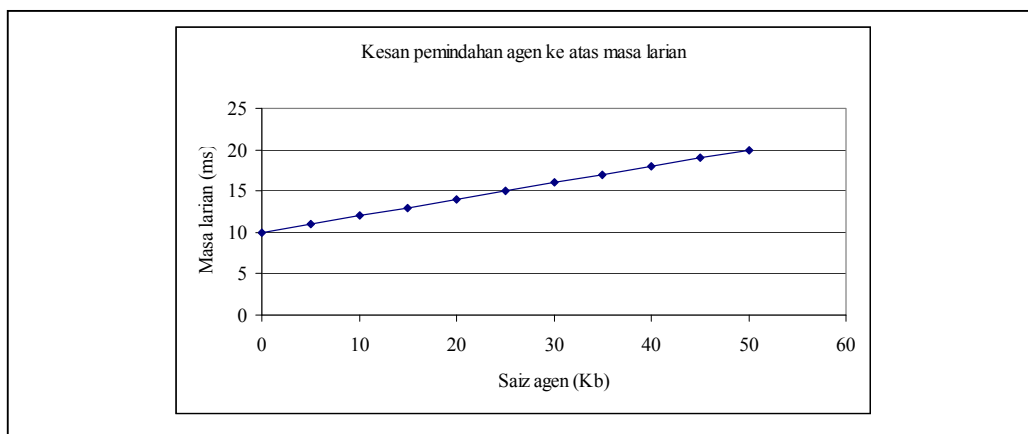
- (c) Masa larian bagi pemindahan agen BDI-MU di antara dua buah komputer (Rajah 10.6).
- (d) Masa larian bagi hebahan mesej di antara dua agen BDI-MU (Rajah 10.7).

Pengukuran dijalankan pada persekitaran pengkomputeran dua buah komputer yang terangkai melalui intranet menggunakan protokol rangkaian TCP/IP. Spesifikasi komputer adalah Pentium III 500 MHz, 256 MB RAM, 10/100 MBps NIC, JDK

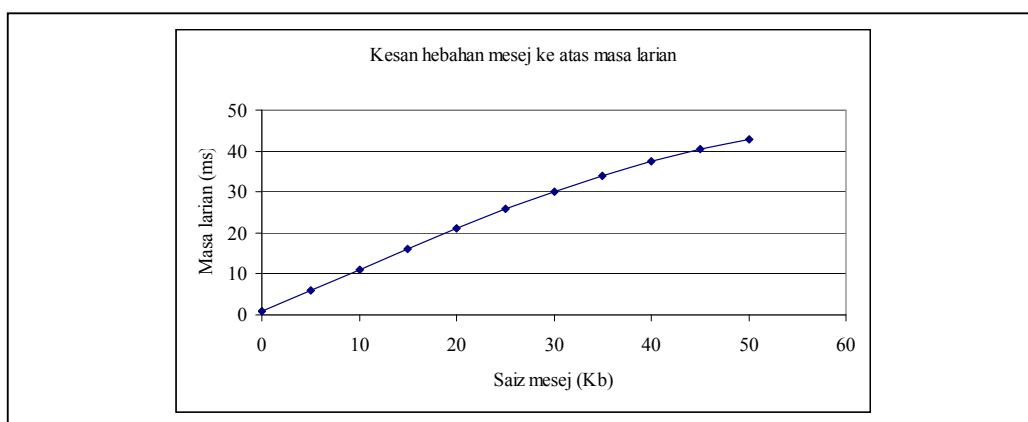
1.3.1 dengan sebuah komputer dipasang Windows XP dan sebuah lagi dipasang Red Hat Linux.

**Jadual 10.2 : Maklumat bagi data-data sistem pengurusan inventori teragih**

No	Item	Data
1.	Bilangan agen sedang jalan	30
2.	Saiz pangkalan data	65.5 MB
3.	Penggunaan maksima CPU	14 %
4.	Penggunaan maksima ingatan	166 MB
5.	Kelajuan rangkaian	10 MBps
6.	Kelajuan maksima penerimaan dan penghantaran data	550 KB/s
7.	Saiz maksimum agen	50 KB
8.	Saiz maksimum kontena	20.6 KB
9.	Saiz maksimum mesej	50 KB



**Rajah 10.6 : Graf kesan pemindahan agen BDI-MU ke atas masa larian**



**Rajah 10.7 : Graf kesan hebahan mesej ke atas masa larian**

Sebagaimana yang ditunjukkan pada Rajah 10.6 dan Rajah 10.7, keputusan ini menunjukkan walaupun saiz agen BDI-MU dan saiz mesej meningkat tetapi kenaikan masa larian adalah munasabah. Ini menandakan bahawa penambahan saiz agen BDI-MU akibat daripada peningkatan saiz kontenanya dan penambahan saiz mesej yang dihebahkan di antara agen BDI-MU untuk meningkatkan kerjasama dan penyelarasan di antara agen BDI-MU di dalam persekitaran pengkomputeran teragih dan heterogen tidak menimbulkan kesan luar jangka yang akan mengurangkan prestasi larian agen BDI-MU.

## **10.8 Ringkasan**

Mesej dan konsep bahasa Mobile UNITY telah dihibridkan untuk membina komponen komunikasi dan komponen mobiliti yang digunakan untuk membentuk struktur inter agen BDI-MU. Mesej berfungsi untuk menyelesaikan masalah komunikasi teragih. Struktur data mesej berupaya memindahkan keadaan data dan keadaan kawalan agen BDI-MU. Selain mampu menghantar fail, objek atau kod bait kelas yang diperlukan oleh agen BDI-MU. Konsep bahasa Mobile UNITY pula berfungsi untuk menyelesaikan masalah simulasi teragih. Konsep bahasa Mobile UNITY membolehkan agen BDI-MU melawati satu set hos teragih yang saling berhubungan melalui rangkaian tanpa campur tangan hos penghantar. Komponen komunikasi dan komponen mobiliti diperlukan untuk meningkatkan keupayaan menyelaraskan dan mencapai sumber teragih secara lebih maksima bagi meningkatkan keupayaan agen BDI-MU membuat keputusan. Hasil eksperimen yang dibentangkan dalam Seksyen 10.7 menunjukkan keupayaan agen BDI-MU berkomunikasi dan bergerak dalam persekitaran pengkomputeran yang teragih dan heterogen dengan masa larian yang munasabah. Di dalam seni bina BDI-MU, komponen komunikasi dan komponen mobiliti dikelompokkan dalam pakej MobileUNITY. Spesifikasi formal operasi dan hubungan komponen-komponen yang membentuk struktur inter agen BDI-MU dibentangkan dalam Bab XI. Huraian kelas-kelas yang mewakili komponen-komponen tersebut termasuk implementasinya dalam sistem pengurusan inventori teragih pula dibentangkan dalam Bab XII.



## **BAB XI**

### **SPESIFIKASI FORMAL STRUKTUR INTRA DAN INTER AGEN BDI-MU MENGGUNAKAN Z**

#### **11.1 Pengenalan**

Salah satu kaedah untuk meningkatkan kualiti perisian adalah dengan mengubah cara bagaimana perisian didokumentasikan semasa paras reka bentuk, implementasi dan selepas ianya diterbitkan. Menurut Cavalcanti dan Woodcock (1998), pembangunan spesifikasi formal melalui penggunaan notasi Z yang berasaskan elemen matematik boleh digunakan untuk menghasilkan dokumentasi yang tepat dan tidak kabur. Melalui Z, maklumat perisian distrukturkan dan dipersembahkan dalam satu jujukan spesifikasi. Bagi setiap jujukan spesifikasi, proses penghalusan dilakukan supaya ia lebih terperinci daripada sebelumnya. Proses ini secara tidak langsung akan bergerak ke arah atur cara yang dapat memenuhi spesifikasi sistem. Aturan ini telah berjaya dilaksanakan dalam pembangunan agen dan sistem agen seperti DESIRE (Brazier et al., 1997), Concurrent METATEM (Fisher, 1996), Bayou (Terry et al., 1995) dan PRS (Georgeff dan Lansky, 1986). Tujuannya adalah untuk memastikan kebenaran reka bentuk sistem agen khususnya ketepatan seni bina agen.

Objektif bab ini adalah untuk membangunkan spesifikasi formal struktur intra dan inter agen BDI-MU menggunakan Z. Pembangunan spesifikasi formal merangkumi huraian atribut, kelakuan dan hubungan komponen-komponen yang digunakan untuk membentuk struktur intra dan inter agen BDI-MU.

## 11.2 Bahasa Formal Z

Z (Spivey, 1996) adalah bahasa spesifikasi berasaskan teori set ( $\cap$ ,  $\cup$ ,  $\in$ ,  $\notin$ ) dan logik matematik ( $\neg$ ,  $\wedge$ ,  $\vee$ ,  $\Rightarrow$ ,  $\forall$ ,  $\exists$ ) yang dibangunkan oleh Programming Research Group di Universiti Oxford semenjak 1978. Dalam Z, objek matematik termasuk atribut dan operasinya digabungkan dalam skima. Skima mengandungi dua bahagian iaitu perisytiharan pembolehubah dan predikat yang mengekang nilai pembolehubah. Teks dan contoh skima adalah seperti berikut:

<i>Nama-skima</i>	<i>BukuKelahiran</i>
<i>Perisytiharan</i>	<i>persona</i> : $\mathbb{P}NAMA$ <i>tarikhLahir</i> : $NAMA \rightarrow TARIKH$
<i>Predikat</i>	<i>persona</i> = dom <i>tarikhLahir</i>

Skima digunakan untuk menghuraikan keadaan dan ciri-ciri sistem dari aspek yang statik dan dinamik. Dalam skima, jenis data matematik digunakan untuk memodelkan data sistem. Notasi logik predikat pula digunakan untuk menghuraikan operasi sistem. Jenis data matematik dan notasi logik predikat seterusnya digunakan untuk menulis spesifikasi formal bagi reka bentuk sistem. Spesifikasi formal ini kemudiannya boleh dibuktikan dan diuji bagi memastikan reka bentuk sistem adalah betul, memenuhi spesifikasi dan konsisten. Tujuannya untuk meningkatkan kualiti, pelaksanaan dan kebolehpercayaan sistem. Pembuktian spesifikasi formal boleh dilakukan secara lelaran. Ia bagi memastikan penghalusan reka bentuk sistem dapat dilaksanakan sehingga ia menghampiri kod boleh laku.

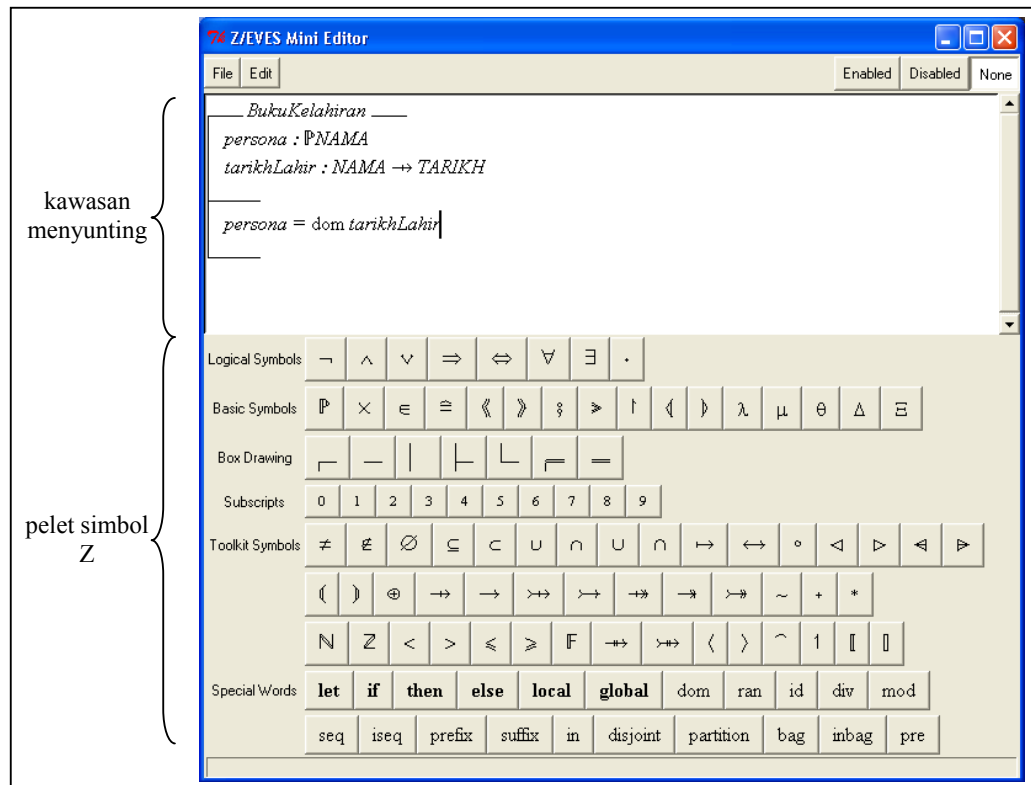
Menurut Luck dan d'Inverno (1995), ciri-ciri utama Z yang membolehkannya digunakan untuk menghasilkan spesifikasi formal agen dan sistem agen adalah:

- (a) Z menyediakan modulariti dan peringkasan bagi perwakilan atribut dan operasi agen yang konsisten, tergabung dan berstruktur.
- (b) Membantu pembangunan model formal untuk menerangkan konsep sesuatu model atau reka bentuk agen yang diketengahkan.

Z juga telah diterima pakai di kalangan komuniti agen sebagai piawaian dan penyebaran teknologi agen sebagaimana kerja-kerja yang telah dijalankan oleh d'Inverno dan Luck (1998), Kinny dan Georgeff (1997), Goodwin (1995) dan Craig (1991).

### 11.3 Alatan Z-Eves

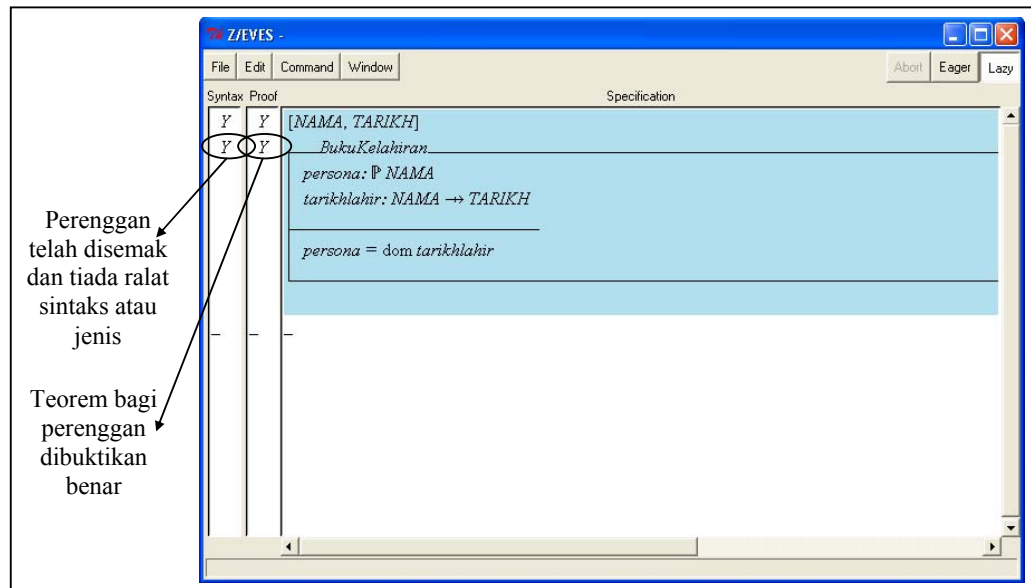
Z-Eves (Saaltink, 1997) adalah alatan yang dibangunkan oleh ORA Canada untuk menulis, menyemak sintaks dan jenis serta membuktikan teorem bagi spesifikasi Z. Rajah 11.1 menunjukkan tettingkap *Edit* yang digunakan untuk mengompos dan menyunting spesifikasi Z.



**Rajah 11.1 : Mengompos dan mengedit spesifikasi Z**

Rajah 11.2 pula menunjukkan tettingkap *Specification* yang digunakan untuk membuktikan spesifikasi Z. Status setiap perenggan ditunjukkan pada lajur *Syntax* dan *Proof*. Lajur *Syntax* menandakan status semakan sintaks dan jenis bagi perenggan. Terdapat tiga status semakan iaitu “?” menandakan perenggan belum disemak, “Y” menandakan tiada ralat sintaks atau jenis dan “N” menandakan terdapat ralat sintaks atau jenis. Lajur *Proof* pula menandakan status pembuktian teorem bagi perenggan. Terdapat tiga status pembuktian iaitu “?” menandakan perenggan belum disemak dan pembuktian teorem tidak dapat ditentukan, “Y” menandakan teorem dibuktikan benar dan “N” menandakan teorem dibuktikan tidak

benar. Teori set Zermelo-Fraenkel (Bancerek, 1990) yang dipelopori oleh Ernst Zermelo dan Adolf Fraenkel digunakan oleh Z-Eves untuk membuktikan teorem bagi spesifikasi Z. Ia adalah sistem berasaskan aksiom yang terawal dalam teori set. Tujuannya adalah untuk menangani masalah paradoks di dalam teori set.



**Rajah 11.2 : Menyemak dan membuktikan spesifikasi Z**

#### 11.4 Spesifikasi Formal Struktur Intra Agen BDI-MU

Struktur intra agen BDI-MU dibentuk oleh komponen-komponen model dunia, matlamat, pelan, struktur niat, pentafsir dan pemerhati sebagaimana yang telah dihuraikan dalam Bab IX. Seksyen 11.4.1 sehingga Seksyen 11.4.4 di dalam bab ini pula membentangkan spesifikasi formal operasi dan hubungan di antara komponen-komponen tersebut menggunakan Z. Perisytiharan simbol-simbol yang digunakan adalah seperti pada Lampiran D. Struktur intra agen adalah struktur yang penting di dalam seni bina BDI-MU kerana komponennya membentuk pakej BDI yang digunakan untuk mencorak matlamat, kepercayaan, keinginan dan keupayaan agen BDI-MU. Tujuannya adalah untuk membolehkan agen BDI-MU menaakul dan membuat keputusan secara peribadi sebagai satu perisian individu.

### 11.4.1 Kepercayaan, Matlamat dan Tindakan

Set kepercayaan agen BDI-MU adalah semua formula kepercayaan sama ada ianya merupakan atom atau penyangkalan atom.

Syntax	Proof	Specification
Y	Y	$Belief ::= \{b : BeliefFormula \mid beIvars\ b = \emptyset \cdot b\}$

Matlamat agen BDI-MU adalah keadaan peristiwa yang agen BDI-MU perlu capai atau mengenal pasti sama ada sesuatu peristiwa keadaan sedang dipegang oleh agen BDI-MU.

Syntax	Proof	Specification
Y	Y	$Goal ::= achieve\langle\langle BeliefFormula \rangle\rangle \mid query\langle\langle SituationFormula \rangle\rangle$

Tindakan agen BDI-MU dibahagikan kepada tindakan luaran dan tindakan dalaman. Tindakan luaran agen BDI-MU adalah perlakuan agen BDI-MU ke atas persekitarannya.

Syntax	Proof	Specification
Y	Y	$ExtAction$ $name : ActionSym$ $terms : seq\ Term$

Tindakan dalaman agen BDI-MU pula adalah penambahan atau penghapusan kepercayaan daripada pangkalan pengetahuan agen BDI-MU.

Syntax	Proof	Specification
Y	Y	$IntAction ::= add\langle\langle BeliefFormula \rangle\rangle \mid remove\langle\langle BeliefFormula \rangle\rangle$

### 11.4.2 Pelan, Tika Pelan dan Niat

Pelan adalah kekangan kepada kelakuan agen BDI-MU dan bertindak sebagai niat yang akan dilaksanakan oleh agen BDI-MU. Pelan agen BDI-MU terdiri daripada syarat penjana yang akan mencetuskan pelan dan jujukan tindakan yang perlu dilaksanakan oleh agen BDI-MU.

Syntax	Proof	Specification
Y	Y	<pre> Plan inv : TriggerEvent context : optional [SituationFormula] body : Body maint : SituationFormula succ : seq IntAction fail : seq IntAction </pre>

Perlakuan asas agen BDI-MU melibatkan penjanaan satu senarai pelan yang bersesuaian, membuat pemetaan pelan, memilih pelan yang relevan daripada senarai pelan dan menghasilkan tika ke atas pelan yang dipilih. Tika tersebut mengandungi salinan asal pelan dengan tambahan persekitaran dan keadaan semasa agen BDI-MU serta status semasa tika berkenaan.

Syntax	Proof	Specification
Y	Y	<pre> PlanInstance origplan : Plan env : Substitution state : State nextbranches : PBranch branch : optional [Branch] status : Status id : PlanInstanceId </pre> <hr/> <pre> state ∈ PlanStates origplan state ∈ dom End ⇒ nextbranches = ∅ nextbranches ⊆ PlanNextBranches origplan state branch ⊆ nextbranches </pre>

Tika pelan terdiri daripada tika awalan (sewaktu pelan mula dijana), tika berjaya (pelan mencapai keadaan akhirnya) dan tika gagal (pelan tidak mencapai keadaan akhir dan tiada cabang untuk dijelajahi).

Syntax	Proof	Specification
Y	Y	$InitialInstance == \{p : PlanInstance \mid p.state = PlanStartState \wedge p.origplan \wedge p.status = active\}$
Y	Y	$SucceedInstance == \{p : PlanInstance \mid p.state \in (dom End)\}$
Y	Y	$FailedInstance == \{p : PlanInstance \mid p.state \notin (dom End) \wedge p.nextbranches = \{\}\}$

Niat agen BDI-MU terdiri daripada satu jujukan tika pelan. Pelan yang berada di kedudukan teratas tindakan niat adalah pelan yang akan dilaksanakan oleh agen BDI-MU.

Syntax	Proof	Specification
Y	Y	$Intention == seq\ PlanInstance$

### 11.4.3 Operasi Agen BDI-MU

Operasi agen BDI-MU adalah berdasarkan interaksi di antara niat dan keadaan agen BDI-MU. Oleh kerana itu, spesifikasi operasi agen BDI-MU merangkumi keadaan agen BDI-MU, penjanaaan pelan relevan dan boleh guna, memproses keadaan, perlakuan niat dan pencapaian pelan.

Agen BDI-MU secara generik mengandungi komponen pustaka pelan, fungsi pemilihan niat, fungsi pemilihan keadaan, fungsi pemilihan pelan, fungsi pemilihan gantian dan fungsi pemilihan cabang.

Syntax	Proof	Specification
Y	Y	<pre> Agent planlibrary : PPlan intentionselect : P1Intention → Intention planselect : P1Plan → Plan eventselect : seq Event → Event substitutionselect : P1Substitution → Substitution selectbranch : PlanInstance → Branch </pre>

Bagi keadaan agen BDI-MU pula, penumpuan diberikan terhadap aspek-aspek yang berubah mengikut masa. Antara aspek tersebut ialah kepercayaan agen BDI-MU, niat dan jujukan keadaan yang akan diproses.

Syntax	Proof	Specification
Y	Y	<pre> AgentState Agent beliefs : PBelief intentions : PIntention event : seq Event </pre>

Pelan dianggap relevan sekiranya wujud gabungan paling umum untuk mengikat keadaan cetusan pelan dengan keadaan supaya kedua-duanya adalah setara. Ia dispesifikasikan oleh fungsi *genrelplans* yang mengambil keadaan (*e*) dan satu set

pelan ( $ps$ ). Kemudian memulangkan satu set pasangan pelan-gantian ( $p, \sigma$ ), di mana  $p$  ialah pelan relevan di dalam  $ps$  bagi  $e$  dan  $\sigma$  ialah gabungan paling umum bagi  $p$ .

Syntax	Proof	Specification
Y	Y	$\text{genrelplans} : \text{Event} \rightarrow \mathbb{P}\text{Plan} \rightarrow \mathbb{P}(\text{Plan} \times \text{Substitution})$ <hr/> $\forall e : \text{Event}; \text{lib} : \mathbb{P}\text{Plan}$ $\text{undefined } e.\text{env} \Rightarrow \text{genrelplans } e \text{ lib} =$ $\{p : \text{lib}; \sigma : \text{Substitution} \mid \text{mguevents } (e.\text{trig}, p.\text{inv}) = \sigma \cdot (p, \sigma)\} \wedge$ $\text{defined } e.\text{env} \Rightarrow \text{genrelplans } e \text{ lib} =$ $\{p : \text{lib}; \sigma : \text{Substitution} \mid \text{mguevents } (\text{ASTrigEvent } (the\ e.\text{env})\ e.\text{trig}, p.\text{inv}) = \sigma \cdot (p, \sigma)\}$

$\text{genapplplans}$  pula adalah pelan boleh guna yang memegang pelan relevan, penggantian dan satu set kepercayaan semasa. Ia mengambil satu set pelan dan kepercayaan semasa. Kemudian memulangkan pelan boleh guna dan penggantian yang telah dikemaskinikan.

Syntax	Proof	Specification
Y	Y	$\text{genapplplans} : \mathbb{P}(\text{Plan} \times \text{Substitution}) \rightarrow (\mathbb{P}\text{BeliefFormula}) \rightarrow \mathbb{P}(\text{Plan} \times \text{Substitution})$ <hr/> $\forall \text{relsubs} : \mathbb{P}(\text{Plan} \times \text{Substitution}); \text{bels} : \mathbb{P}\text{BeliefFormula} \cdot$ $\text{genapplplans } \text{relsubs } \text{bels} =$ $\{rel : \text{Plan}; \sigma, \psi : \text{Substitution} \mid (rel, \sigma) \in \text{relsubs} \wedge$ $\text{LogCons}(\text{ASSitForm } (\sigma \dagger \psi) (the\ rel.\text{context}), \text{bels}) \cdot (rel, (\sigma \dagger \psi))\}$

Terdapat dua mod pemprosesan keadaan yang bergantung kepada penimbal keadaan sama ada ianya kosong ataupun tidak. Sekiranya penimbal keadaan tidak kosong, maka keadaan dan pelan relevan dipilih daripada penimbal tersebut. Sebaliknya, pelan boleh guna akan ditentukan untuk menjana tika pelan.

Syntax	Proof	Specification
Y	Y	$\text{NewPlanInstance}$ <hr/> $\Delta \text{AgentState}$ <hr/> $\text{event} \neq \diamond$ $\text{let } \text{events} == \text{eventsselect } \text{events} \cdot$ $\quad \text{let } \text{applplans} == \text{genapplplans}(\text{genrelplans } \text{event } \text{planlibrary}) \text{ beliefs} \cdot$ $\quad \quad \text{let } \text{selectedplan} == \text{planselect}(\text{dom } \text{applplans})$ $\quad \quad \quad \text{let } \text{applunifier} == \text{applplans } \text{selectedplan} \cdot$ $\quad \quad \quad \quad \text{let } \text{instance} == \text{CreatePlanInstance } \text{selectedplan } \text{applunifier} \cdot$ $\quad \quad \quad \quad \quad \text{event} \in \text{ExternalEvent} \Rightarrow$ $\quad \quad \quad \quad \quad \text{instance} \notin (the\ \text{event}.\text{failures}) \wedge$ $\quad \quad \quad \quad \quad \text{intentions}' = \text{intentions} \cup \{\langle \text{instance} \rangle\} \wedge$ $\quad \quad \quad \quad \quad \text{events}' = (\text{events} \triangleright \{\text{event}\}) \cup$ $\quad \quad \quad \quad \quad \quad \{(events \sim \text{event}, \text{MakeEvent}(\text{event}.\text{trig}, \{\text{instance}.\text{id}\},$ $\quad \quad \quad \quad \quad \quad \quad \emptyset, \emptyset)\} \wedge$ $\quad \quad \quad \quad \quad \text{event} \in \text{SubgoalEvent} \Rightarrow$ $\quad \quad \quad \quad \quad \quad (\text{let } \text{trigint} == (\cup i : \text{intentions} \mid \text{head } i.\text{id} = (the\ \text{event}.\text{id})) \cdot$ $\quad \quad \quad \quad \quad \quad \text{intentions}' = \text{intentions} \setminus \{\text{trigint}\} \cup \{\langle \text{instance} \rangle\} \wedge \text{trigint})$



Spesifikasi bagi perlakuan niat agen BDI-MU dinyatakan oleh fungsi keadaan operasi perlakuan niat agen BDI-MU (*AgentIntExecutionOperationState*). Ia bermula dengan memilih niat (*selectedintention*), mengenal pasti pelan yang akan dilaksanakan (*executingplan*) dan memilih cabang bagi pelan yang akan dilaksanakan (*executingbranch*).

Syntax	Proof	Specification
Y	Y	<pre> AgentIntExecutionOperationState AgentState selectedIntention : optional [Intention] executingplan : optional [PlanInstance] executingbranch : optional [Branch] </pre>

Sesuatu pelan dianggap berjaya sekiranya keadaan akhir dicapai. Pada situasi ini, persekitaran gantian (*(the executing plan).env*) digunakan ke atas syarat berjaya (*(the executingplan).origplan.succ*) untuk memberikan satu jujukan tindakan dalaman (*groundsuccacts*). Selain itu, pangkalan pengetahuan dikemaskini untuk menghasilkan satu set kepercayaan yang baru (*beliefs'*).

Syntax	Proof	Specification
Y	Y	<pre> AchievePlan ΔAgentIntExecutionOperationState  the executingplan ∈ SucceedInstance let succacts == (the executingplan).origplan.succ; env == (the executingplan).env .   let groundsuccacts == map (ASIntAction env) succacts .   beliefs' = fold performintaction beliefs groundsuccacts </pre>

Pelan dianggap gagal sekiranya tiada lagi cabang alternatif yang boleh dijelajahi. Keadaan ini dispesifikasikan dengan mengimplementasikan persekitaran penggantian kepada syarat gagal bagi pelan (*failacts*) dan pelaksanaan tindakan luaran yang gagal (*groundfailacts*).

Syntax	Proof	Specification
Y	Y	<pre> FailPlan ΔAgentIntExecutionOperationState  the executingplan ∈ FailedInstance let origevent == (λ e : Event \ the e.id = (the executingplan).id);   env == (the executingplan).env .   let failacts == (the executingplan).origplan.fail .   let groundfailacts == map (ASIntAction env) failacts .   beliefs' = fold performintaction beliefs failacts ^   ran events' = (ran events \ {origevent}) ∪     {MakeEvent(origevent.trig, origevent.id, origevent.env,       (origevent.failures ∪ {executingplan})}   the selectedintention' = tail(the selectedintention) </pre>

### 11.4.4 Autonomi

Agen berautonomi ditakrifkan sebagai agen dengan motivasi dan berpotensi menilai kelakuannya. Motivasi adalah keinginan atau kecenderungan yang boleh memandu kepada penjanaaan dan penyesuaian matlamat. Ia memberi kesan kepada hasil taakulan dan tindakan bagi memenuhi sesuatu matlamat. Kelakuan agen berautonomi pula didorong oleh pengaruh dalaman dan luaran.

Syntax	Proof	Specification
Y	Y	<pre> AutonomousAgent Agent motivations : P Motivation ----- motivations ≠ {} </pre>

Motivasi dan matlamat memainkan peranan dalam menapis aspek-aspek relevan persekitaran semasa agen berautonomi membuat tanggapan.

Syntax	Proof	Specification
Y	Y	<pre> AutonomousAgentPercepts AutonomousAgent AgentPercepts autowillperceive : P Motivation → P Goal → Environment → View ----- willperceive = autowillperceive motivations dom autowillperceive = {motivations} </pre>

Tindakan yang dijana oleh agen berautonomi bergantung kepada motivasi dan kesanggupan agen tersebut melaksanakan sesuatu tindakan.

Syntax	Proof	Specification
Y	Y	<pre> AutonomousAgentActions AutonomousAgent AgentActions autoactions : P Motivation → P Goal → View → Environment → P Action ----- dom autoactions = {motivations} agentactions = autoactions motivations </pre>

Keadaan agen berautonomi dalam persekitaran bersandar kepada tanggapan agen berautonomi, tindakan agen berautonomi dan keadaan agen.

Syntax	Proof	Specification
Y	Y	<i>AutonomousAgentState</i> <i>AutonomousAgentPercepts</i> <i>AutonomousAgentActions</i> <i>AgentState</i>

## 11.5 Spesifikasi Formal Struktur Inter Agen BDI-MU

Komponen komunikasi dan komponen mobiliti adalah komponen yang membentuk struktur inter agen BDI-MU sebagaimana yang telah diuraikan dalam Bab X. Seksyen 11.5.1 dan Seksyen 11.5.2 di dalam bab ini pula membentangkan spesifikasi formal operasi dan hubungan di antara komponen-komponen tersebut menggunakan Z. Perisytiharan simbol-simbol yang digunakan adalah seperti pada Lampiran D. Komponen-komponen bagi struktur inter agen berperanan membentuk pakej MobileUNITY di dalam seni bina BDI-MU yang digunakan untuk mencorak kerjasama, penyelarasan dan hubungan di antara agen BDI-MU dengan agen lain dan persekitarannya. Tujuannya adalah untuk membolehkan agen BDI-MU bergerak di sepanjang rangkaian dan berkomunikasi dengan agen atau entiti lain bagi meningkatkan kemampuannya membuat keputusan secara berkumpulan dalam persekitaran teragih dan heterogen.

### 11.5.1 Hubungan dan Kerjasama

Agen,  $c$  mempunyai hubungan langsung dengan agen pelayan,  $s$  jika dan hanya jika terdapat perhubungan langsung di antara  $c$  dan  $s$  ( $c$  dijana oleh  $s$ ). Di mana  $s$  mengawal dan mengurus satu set agen yang beroperasi dalam hosnya.

Syntax	Proof	Specification
Y	Y	<i>Dengages</i> <i>AgentStructure</i> $dengs : Agent \leftrightarrow ServerAgent$ <hr/> $dengs = \{e : direngs \cdot (e.client, e.server)\}$

Agen,  $c$  mempunyai hubungan dengan agen pelayan yang lain,  $s$  jika terdapat rantaian hubungan di antara  $c$  dan  $s$  ( $c$  bergerak dan beroperasi di dalam hos  $s$ ).

Syntax	Proof	Specification
Y	Y	<p><i>Engages</i></p> <p><i>AgentStructure</i></p> <p><math>engages : Agent \leftrightarrow ServerAgent</math></p> <hr/> <p><math>engages = \{ec : engchains \cdot (ec.auto, head\ ec.chain)\}</math>  <math>\cup \{ec : engchains; c, s : Agent \mid ((c, s), ec.chain) \in follows \cdot (c, s)\}</math></p>

Agen,  $c$  dimiliki oleh agen pelayan,  $s$  jika  $c$  dijana oleh  $s$  dan  $c$  bergerak di sepanjang rangkaian hos-hos yang berhubungan dengan  $s$ .

Syntax	Proof	Specification
Y	Y	<p><i>Owms</i></p> <p><i>Dengages</i></p> <p><math>owms : Agent \leftrightarrow ServerAgent</math></p> <hr/> <p><math>\forall c : Agent; s : ServerAgent \cdot (c, s) \in owms \Leftrightarrow</math>  <math>(\forall ec : engchains \mid s \in ran\ ec.chain \cdot (c, s) \in owms \cap dengs)</math></p>

Agen,  $a$  bekerjasama dengan agen,  $b$  jika terdapat komunikasi di antara  $a$  (agen yang dijana) dengan  $b$  (satu set agen yang saling bekerjasama).

Syntax	Proof	Specification
Y	Y	<p><i>Cooperates</i></p> <p><i>AgentStructure</i></p> <p><math>cooperates : Agent \leftrightarrow Agent</math></p> <hr/> <p><math>cooperates = \cup \{a, b : Agent \mid</math>  <math>(\exists c : coops \cdot a = c.genagent \wedge b \in c.coopagents) \cdot \{(a, b)\}\}</math></p>

## 11.5.2 Komunikasi

Mesej dispesifikasikan oleh pembolehkan *sender*, *receiver*, *kind*, *content* dan *binary*.

Syntax	Proof	Specification
Y	Y	<p><i>Message</i></p> <p><math>sender : Sender</math></p> <p><math>receiver : Receiver</math></p> <p><math>kind : Kind</math></p> <p><math>content : Content</math></p> <p><math>binary : Binary</math></p>

Semasa menerima mesej, input adalah mesej dan keadaan agen kekal tidak berubah.

Syntax	Proof	Specification
Y	Y	<i>ReceiveMessage</i> <i>in? : Message</i> $\exists$ <i>Agent</i> <i>AgentState</i>

Semasa menerima mesej, terdapat dua kemungkinan iaitu mesej sah atau mesej tidak sah. Jika sah agen akan responsif ke atas mesej.

Syntax	Proof	Specification
Y	Y	<i>ReceiveMessageSuccess</i> <i>ReceiveMessage</i> $\Delta$ <i>AgentState</i>

Sebaliknya, ralat akan dilaporkan.

Syntax	Proof	Specification
Y	Y	<i>ReceiveMessageFail</i> <i>ReceiveMessage</i> $\exists$ <i>AgentState</i> <i>report! : Report</i>

## 11.6 Ringkasan

Komponen-komponen yang digunakan untuk membentuk struktur intra dan inter agen BDI-MU telah berjaya dispesifikasikan menggunakan Z dalam bentuk skima. Skima menghuraikan operasi dan hubungan komponen-komponen tersebut menggunakan teori set dan logik matematik dengan tujuan untuk memastikan kebenaran reka bentuk struktur intra dan inter agen BDI-MU. Struktur intra dan inter agen adalah struktur utama yang membentuk seni bina BDI-MU. Di mana gabungan komponen-komponen bagi struktur intra agen membentuk pakej BDI. Gabungan komponen-komponen bagi struktur inter agen pula membentuk pakej MobileUNITY. Melalui pembuktian menggunakan alatan Z-Eves, proses penterjemahan komponen-komponen tersebut kepada kod-kod atur cara dapat dilaksanakan dengan teratur untuk menghasilkan alatan BDI-MU kit yang teguh dan mampu menyokong operasi agen BDI-MU yang kompleks dan pelbagai senario.

## **BAB XII**

### **BDI-MU KIT: ALATAN PEMBANGUNAN AGEN MOBIL PINTAR MENGUNAKAN JAVA**

#### **12.1 Pengenalan**

Persekitaran pengkomputeran teragih dan heterogen yang menggabungkan sistem pengoperasian Linux, Windows, Macintosh dan Unix dengan perkakasan superkomputer, komputer meja, komputer buku dan komputer tapak tangan memerlukan agen mobil pintar yang inter operasi untuk membolehkan ia bergerak daripada mesin ke mesin bagi menjana penyelesaian yang terbaik. Java adalah bahasa pengaturcaraan berorientasikan objek yang sesuai digunakan untuk membangunkan agen mobil pintar kerana kod Java yang telah dikompil boleh dilarikan pada sebarang sistem pengoperasian dan perkakasan yang mempunyai JVM. Teknologi dan pakej Java turut berperanan menyokong pembangunan komponen agen mobil pintar yang berautonomi, cerdas dan dinamik. Contoh-contoh agen yang dibangunkan menggunakan Java adalah Aglets (Karjoth et al., 1997), Mole (Baumann et al., 1997), JAM (Chan dan Stolfo, 1997) dan Concordia (Wong et al., 1999).

Objektif bab ini adalah untuk mengetengahkan alatan pembangunan agen mobil pintar yang dikenali sebagai BDI-MU kit yang dibangunkan menggunakan Java. Alatan tersebut kemudiannya digunakan untuk membangunkan agen BDI-MU dan sistem pengurusan inventori teragih. Implementasi BDI-MU kit seterusnya diuji dan ciri-cirinya dibandingkan dengan beberapa alatan pembangunan agen yang lain.

## 12.2 Bahasa Java

Java merupakan bahasa pengaturcaraan yang dibangunkan oleh Sun Microsystems. Kod sumber Java (*\*.java*) tidak diterjemahkan kepada arahan mesin tetapi dikompil kepada kod bait (*\*.class*). Kod bait adalah mudah dialih dan boleh dilarikan oleh pengkompil Java yang dikenali sebagai JVM. Keadaan ini membolehkan kod atur cara ditulis sekali dan dilaksanakan dalam pelbagai sistem pengoperasian. Selain itu, ia boleh diangkut di sepanjang rangkaian dan dilarikan pada sebarang hos sasaran.

Ciri-ciri utama Java yang membolehkannya menyokong pembangunan agen mobil pintar adalah seperti berikut (Bigus dan Bigus, 1999):

- (a) Java menyokong atur cara berautonomi.  
Pelbagai bebenang digunakan untuk menyokong autonomi agen. Ia memberikan kawalan yang berasingan kepada agen bagi membantu agen sentiasa menunggu dan sedia bertindak balas terhadap perubahan persekitaran. Pemerhati dan keadaan pula membantu agen mengesan sebarang cetusan peristiwa.
- (b) Java menyokong atur cara pintar.  
Objek Java dan hubungan di antara objek digunakan untuk mengekod atribut, kelakuan dan persekutuan agen dengan entiti lain. Struktur kawalan dan struktur data Java pula digunakan untuk membentuk pengetahuan, algoritma dan mesej agen.
- (c) Java menyokong atur cara bergerak.  
Pakej *java.net*, model delegasi keadaan, soket dan kod bait Java digunakan untuk menghantar dan melarikan agen dalam mesin yang berlainan melalui rangkaian. Pensirian pula digunakan untuk menyimpan keadaan data dan keadaan kawalan agen.

JAFMAS (Chauhan, 1997), JATLite (Jeon et al., 2000), JADE (Bellifemine et al., 2001), JACK (Norling dan Ritter, 2001) dan JIAC (Fricke et al., 2001) adalah senarai alatan pembangunan agen berasaskan Java. Alatan tersebut telah digunakan untuk membangunkan berbagai-bagai jenis agen yang ditempatkan dalam pelbagai domain dan persekitaran pengkomputeran.

### 12.3 Alatan BDI-MU kit

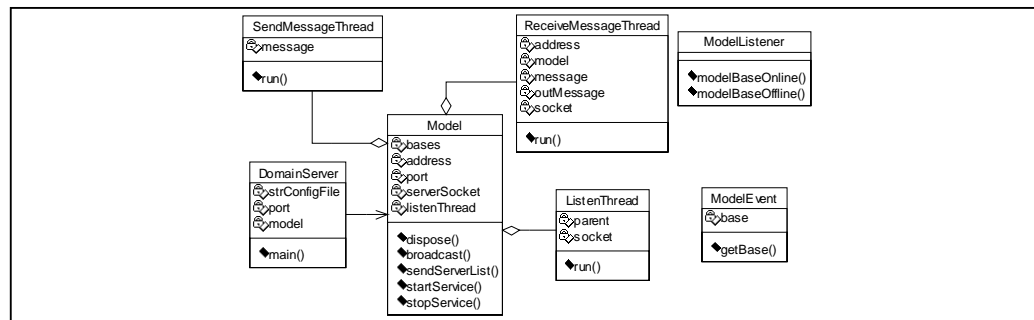
BDI-MU kit adalah alatan pembangunan agen mobil pintar yang dibina oleh komponen-komponen seni bina BDI-MU yang dikelompokkan dalam pakej sistem iaitu Domain, Pelayan, Konfigurasi, BDI dan MobileUNITY seperti yang telah dibincangkan dalam Bab VIII sehingga Bab XI. Kesemua komponen tersebut telah dibangunkan menggunakan Java dan diwakilkan dalam bentuk kelas Java seperti yang dihuraikan dalam Seksyen 12.3.1 sehingga Seksyen 12.3.5. Fungsi setiap pakej adalah seperti berikut:

- (a) Pakej Domain – berfungsi untuk membentuk domain BDI-MU yang akan mengurus satu set pelayan BDI-MU (Rajah 12.1).
- (b) Pakej Pelayan – berfungsi untuk membentuk pelayan BDI-MU yang akan mengurus satu set agen BDI-MU (Rajah 12.2).
- (c) Pakej Konfigurasi – berfungsi untuk mensetkan parameter domain BDI-MU, pelayan BDI-MU serta polisi kebenaran dan keselamatan (Jadual 12.2).
- (d) Pakej BDI – berfungsi untuk membentuk struktur intra agen BDI-MU yang akan mengurus taakulan agen BDI-MU (Rajah 12.3 sehingga Rajah 12.6).
- (e) Pakej MobileUNITY – berfungsi untuk membentuk struktur inter agen BDI-MU yang akan mengurus interaksi dan pergerakan agen BDI-MU (Rajah 12.8).

#### 12.3.1 Kelas-Kelas Domain BDI-MU

Domain BDI-MU dibentuk oleh kelas-kelas di dalam pakej Domain seperti pada Rajah 12.1 yang berperanan mengawal satu set pelayan BDI-MU. Ia juga berfungsi memastikan agen BDI-MU bergerak dan merayau di antara pelayan BDI-MU yang berdaftar kepada sesuatu domain. Ringkasan kelas *DomainServer*, *Model*, *ModelEvent*, *ModelListener*, *ListenThread*, *ReceiveMessageThread* dan *SendMessageThread* yang membentuk pakej Domain adalah seperti pada Jadual 12.1.





**Rajah 12.1 : Kelas-kelas domain BDI-MU**

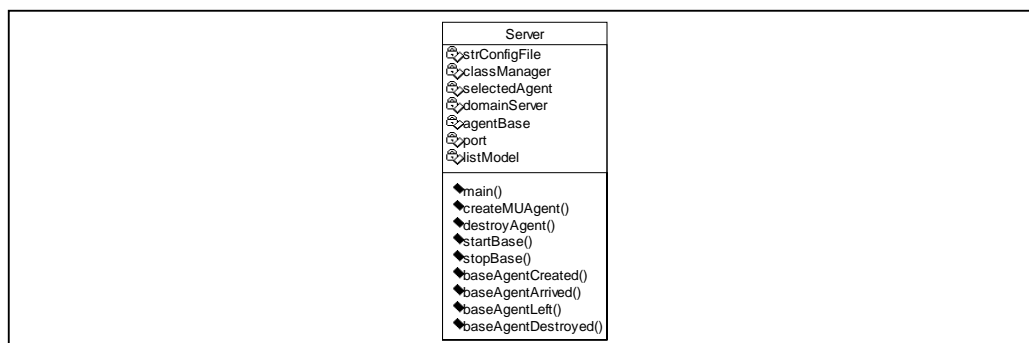
**Jadual 12.1 : Ringkasan kelas-kelas domain BDI-MU**

Kelas	Keterangan
<i>DomainServer</i>	Kelas yang melarikan domain.
<i>Model</i>	Menyediakan infrastruktur kepada domain.
<i>ModelEvent</i>	Mengurus peristiwa yang ditembak oleh domain dan memberi notis kepada pendengar mengenai pelayan yang bersambung atau terputus.
<i>ModelListener</i>	Kelas antara muka yang mengurus peristiwa yang ditembak oleh domain iaitu <i>BASE_ONLINE</i> atau <i>BASE_OFFLINE</i> .
<i>ListenThread</i>	Mendengar port domain dan mengurus agen yang masuk.
<i>ReceiveMessageThread</i>	Mengurus mesej masuk.
<i>SendMessageThread</i>	Mengurus mesej keluar.

### 12.3.2 Kelas-Kelas Pelayan BDI-MU

Pelayan BDI-MU dibentuk oleh kelas *Server* di dalam pakej Pelayan seperti pada Rajah 12.2. *destroyAgent()* membolehkan pelayan BDI-MU menamatkan bebenang agen BDI-MU yang beroperasi di dalam hosnya. *startBase()* dan *stopBase()* berfungsi untuk menukar status pelayan BDI-MU kepada dalam talian atau luar talian. *baseAgentArrived()*, *baseAgentDestroy()* dan *baseAgentLeft()* pula

akan bertindakbalas apabila berlaku perubahan ke atas agen BDI-MU yang dipantaunya seperti agen sampai, agen dihapuskan atau agen bertolak dari hos.



**Rajah 12.2 : Kelas pelayan BDI-MU**

### 12.3.3 Fail-Fail Konfigurasi BDI-MU

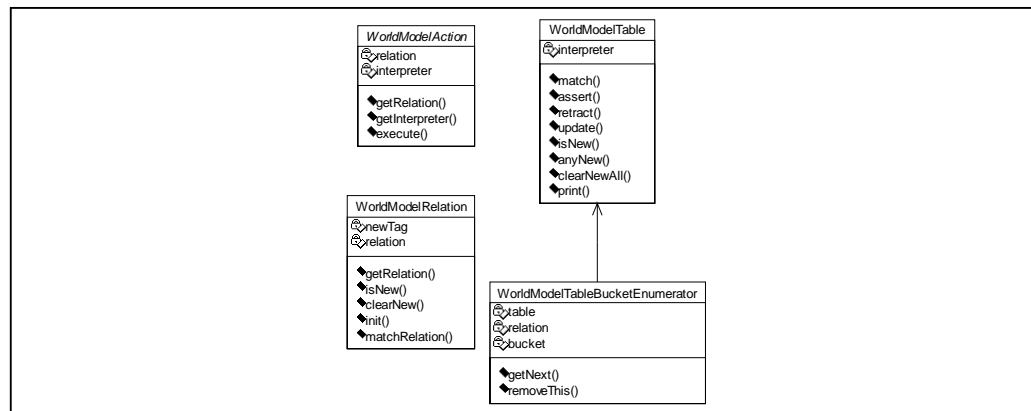
Fail-fail *BDIMU.policy*, *BDIMUDomain.config* dan *BDIMUServer.config* seperti pada Jadual 12.2 berfungsi untuk mengkonfigurasi domain BDI-MU, pelayan BDI-MU serta menentukan polisi keselamatan dan kebenaran.

**Jadual 12.2 : Ringkasan fail-fail konfigurasi BDI-MU**

Fail	Keterangan
<i>BDIMUDomain.config</i>	Mengurus konfigurasi persediaan larian domain.
<i>BDIMUServer.config</i>	Mengurus konfigurasi persediaan larian pelayan.
<i>BDIMU.policy</i>	Mengurus konfigurasi keselamatan domain dan pelayan.

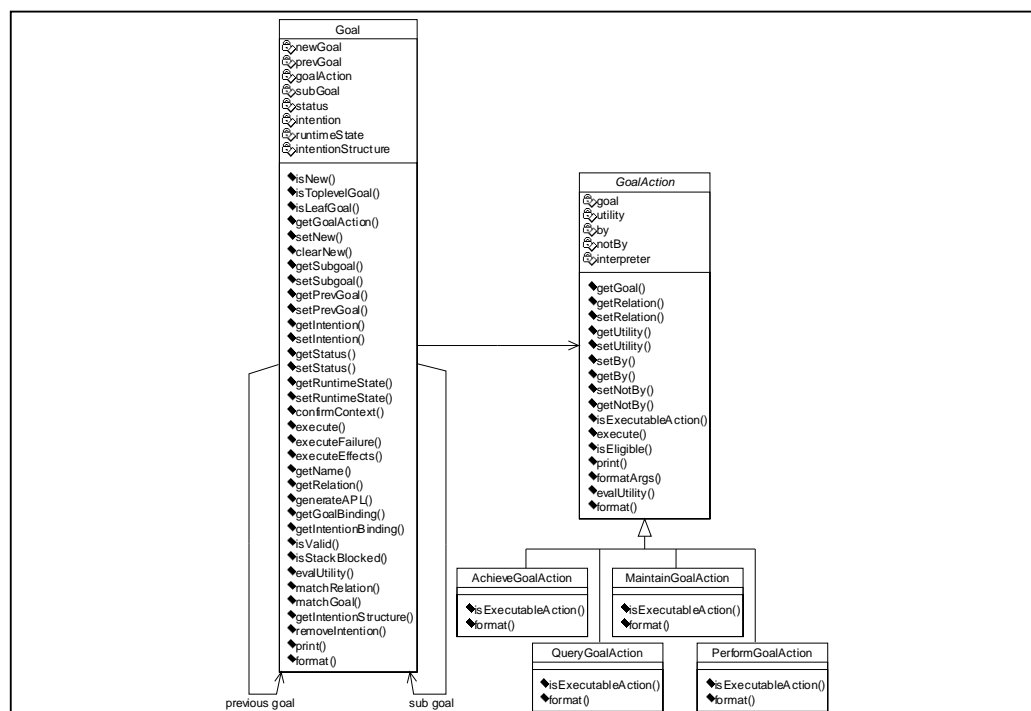
### 12.3.4 Kelas-Kelas Struktur Intra Agen BDI-MU

Struktur intra agen BDI-MU dibentuk oleh enam komponen iaitu model dunia, matlamat, pelan, pemerhati, struktur niat dan pentafsir. Komponen model dunia dibentuk oleh kelas *WorldModelTable*, *WorldModelTableBucketEnumerator*, *WorldModelRelation* dan *WorldModelAction* seperti pada Rajah 12.3.



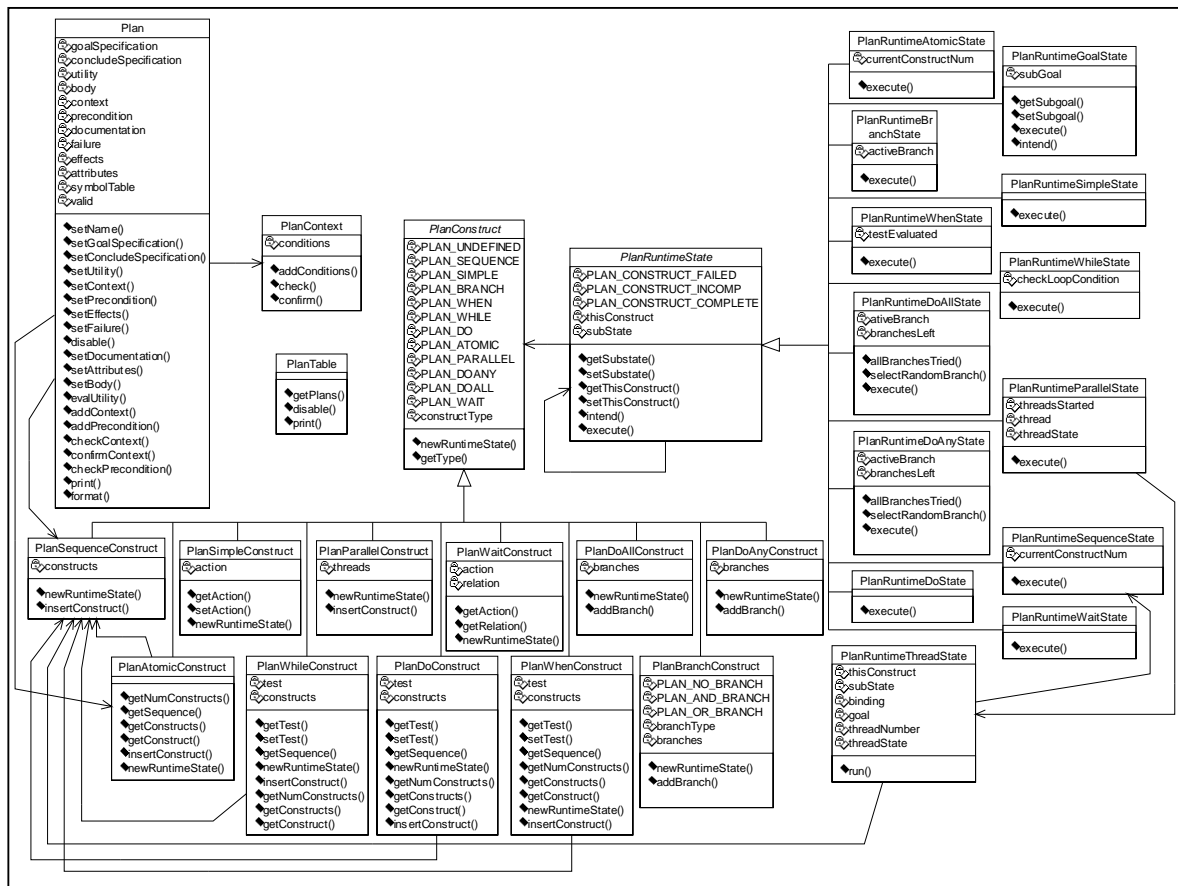
**Rajah 12.3 : Kelas-kelas komponen model dunia**

Komponen matlamat dibentuk oleh kelas *Goal*, *GoalAction*, *AchieveGoalAction*, *MaintainGoalAction*, *PerformGoalAction* dan *QueryGoalAction* seperti pada Rajah 12.4.



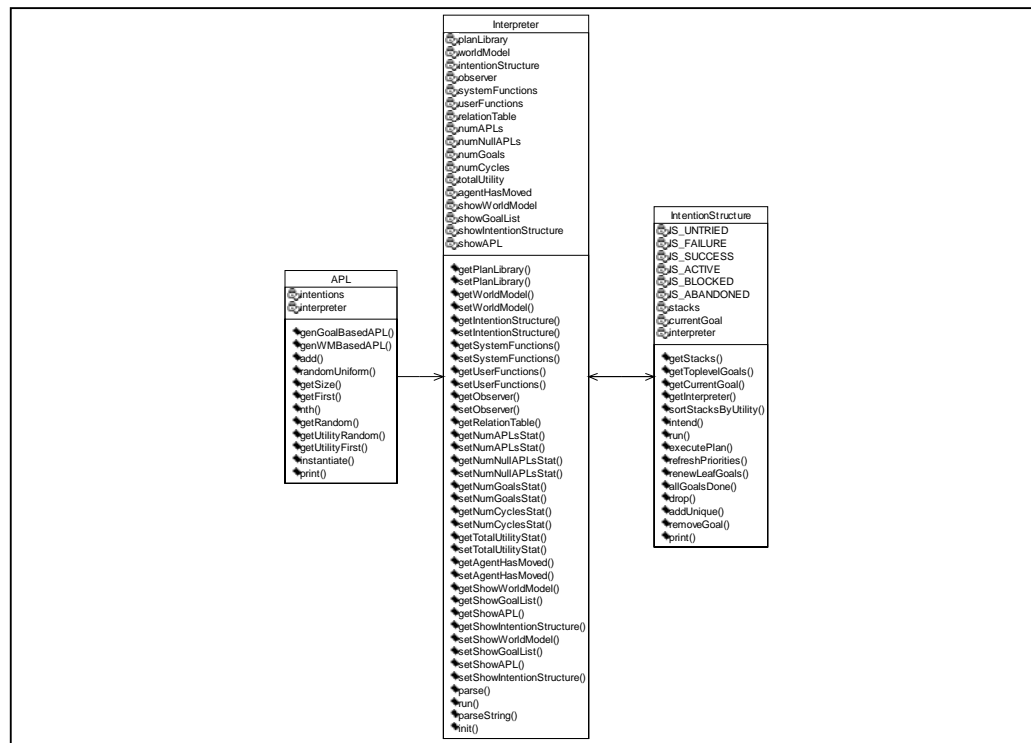
**Rajah 12.4 : Kelas-kelas komponen matlamat**

Pelan dan pemerhati adalah komponen kompleks yang dibentuk oleh kelas *Plan*, *PlanRuntimeState*, *PlanConstruct*, *PlanContext*, *PlanTable* dan lain-lain lagi seperti pada Rajah 12.5.



**Rajah 12.5 : Kelas-kelas komponen plan dan pemerhati**

Komponen struktur niat dan pentafsir dibentuk oleh kelas *Interpreter*, *IntentionStructure* dan *APL* seperti pada Rajah 12.6. Proses taakulan praktikal tertetus apabila berlaku perubahan ke atas model dunia atau matlamat baru dipos. Taakulan praktikal dilaksanakan untuk mencari plan ideal yang akan dilaksanakan ke atas situasi semasa. Pentafsir bertanggungjawab untuk memilih dan melarikan plan berdasarkan niat, plan, matlamat dan kepercayaan mengenai situasi semasa. Agen BDI-MU mungkin mempunyai sejumlah besar plan alternatif untuk menyempurnakan satu matlamat. Pentafsir akan menaakul tentang semua kombinasi alternatif di antara plan, matlamat dan pengikatan pembolehubah (berdasarkan hujahan matlamat dan kepercayaan) sebelum memilih alternatif terbaik bagi sesuatu situasi. Gelung utama pentafsir BDI yang mengawal proses taakulan praktikal agen BDI-MU adalah seperti pada Rajah 12.7.



**Rajah 12.6 : Kelas-kelas komponen struktur niat dan pentafsir**

```

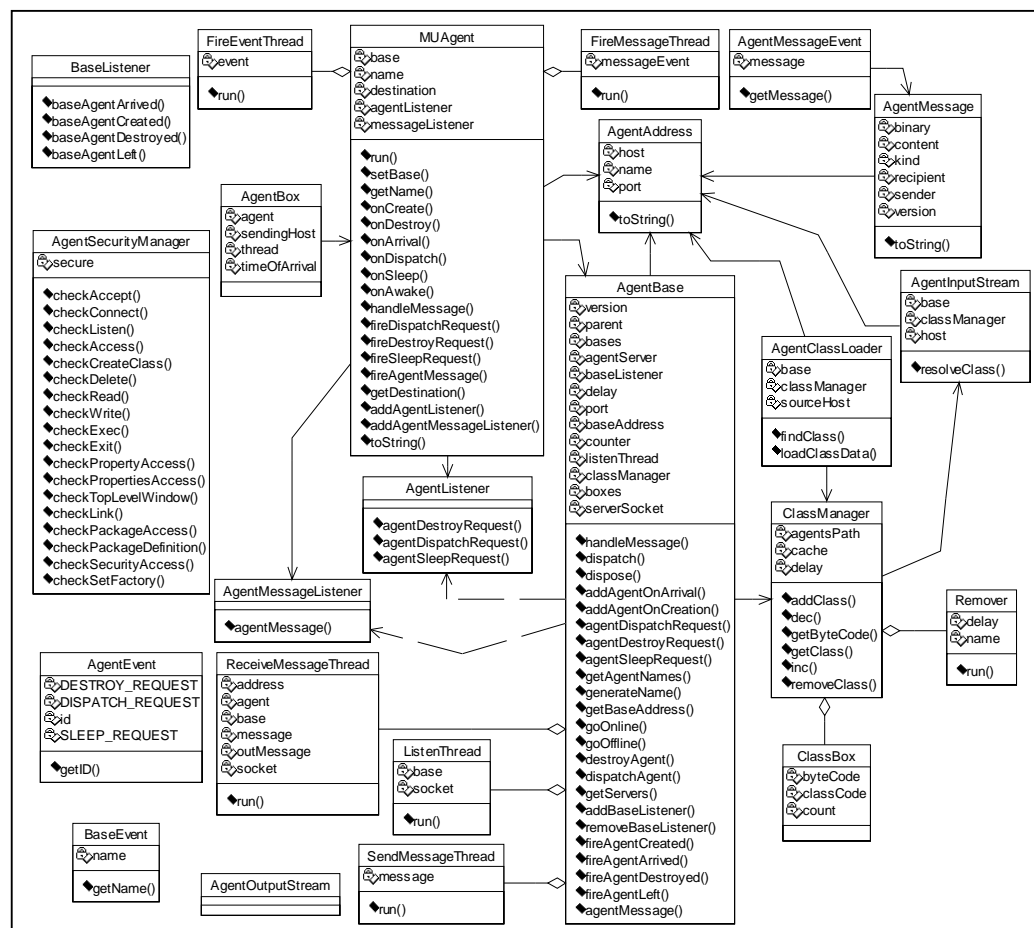
public int run() {
    // laksanakan gelung sehingga agen mencapai semua matlamat
    while (true) { //gelung tidak terhingga, mula
        // laksanakan prosidur Pemerhati
        returnValue = intentionStructure.executePlan(observer); metaLevel = 0;
        while (true) { // gelung aras meta, mula
            // jana Senarai Pelan Boleh Guna (APL), jika perlu
            apl = new APL(planLibrary, worldModel, intentionStructure, metaLevel);
            // sekiranya APL baru atau sebelumnya tidak kosong, maka tambah kemasukan
            // ke dalam Model Dunia Nyata untuk cetus taakulan aras yang berikutnya
            if (apl.getSize() != 0) { WorldModel(.).assert("APL", metaLevel, apl, aplSize); }
            // sekiranya APL kosong, maka tiada taakulan aras yang baru
            if (apl.getSize() == 0) {
                if (intentionStructure.allGoalsDone()) { return 0; }
                // sekiranya APL sebelumnya kosong, maka laksanakan sesuatu pada struktur
                // niat, sebaliknya pilih sesuatu daripada APL sebelumnya, cadangkannya dan
                // larikan sesuatu dalam struktur niat
                if (last_apl == null || last_apl.getSize() == 0) {
                    intentionStructure.run(); break; }
                else {
                    selectedElement = last_apl.getHighestUtilityRandom();
                    intentionStructure.intend(selectedElement);
                    intentionStructure.run(); last_apl = null; break; }
                } else { last_apl = apl; metaLevel++; }
            } // gelung aras meta, tamat
            getWorldModel().retract("APL"); // bersihkan Model Dunia Nyata bagi elemen APL
        } // gelung tidak terhingga, tamat
    }
}

```

**Rajah 12.7 : Gelung utama pentafsir BDI**

### 12.3.5 Kelas-Kelas Struktur Inter Agen BDI-MU

Pakej MobileUNITY mengandung kelas-kelas komponen komunikasi dan mobiliti seperti pada Rajah 12.8 yang berfungsi untuk membentuk struktur inter agen BDI-MU. Selain itu, kelas-kelas tersebut boleh disambung dan diimport untuk membentuk domain BDI-MU dan pelayan BDI-MU. Domain BDI-MU dan pelayan BDI-MU diperlukan untuk menyediakan platform dan infrastruktur larian kepada agen BDI-MU.



**Rajah 12.8 : Kelas-kelas komponen komunikasi dan mobiliti**

Ringkasan kelas-kelas yang terdapat di dalam pakej MobileUNITY seperti *MUAgent*, *AgentBase*, *AgentClassLoader*, *AgentMessage*, *AgentAddress*, *AgentBox*, *AgentSecurityManager*, *ClassManager* dan lain-lain lagi dibentangkan pada Jadual 12.3.

**Jadual 12.3 : Ringkasan kelas-kelas komponen komunikasi dan mobiliti**

<b>Kelas</b>	<b>Keterangan</b>
<i>BaseEvent</i>	Mengurus peristiwa yang ditembak oleh pelayan.
<i>ClassManager</i>	Mengurus kelas dan kod bait kelas yang dimuatkan ke dalam mesej.
<i>ClassBox</i>	Membungkus kelas dan kod bait kelas yang dimuatkan ke dalam mesej.
<i>Remover</i>	Menghapuskan kelas dan kod bait kelas yang gagal dimuatkan ke dalam mesej selepas masa lengah.
<i>MUAgent</i>	Kelas yang disambung untuk membentuk agen.
<i>FireEventThread</i>	Menembak peristiwa kepada pendengar agen.
<i>FireMessageThread</i>	Menembak mesej kepada pendengar mesej agen.
<i>AgentAddress</i>	Mengurus pengalamatan agen.
<i>AgentBase</i>	Kelas yang disambung untuk membentuk pelayan.
<i>ListenThread</i>	Mendengar port pelayan dan mengurus agen masuk.
<i>ReceiveMessageThread</i>	Mengurus mesej masuk.
<i>SendMessageThread</i>	Mengurus mesej keluar.
<i>AgentBox</i>	Menyimpan maklumat agen untuk pemantauan pelayan.
<i>AgentClassLoader</i>	Menjana kelas daripada tatasusunan bait.
<i>AgentEvent</i>	Mengurus peristiwa yang ditembak oleh agen.
<i>AgentInputStream</i>	Membaca objek agen dari strim input.
<i>AgentMessage</i>	Kelas yang membentuk struktur data mesej.
<i>AgentMessageEvent</i>	Mengurus mesej yang ditembak oleh pelayan atau agen.
<i>AgentOutputStream</i>	Menulis objek agen kepada strim output.
<i>AgentSecurityManager</i>	Mengurus keselamatan agen.
<i>BaseListener</i>	Kelas antara muka yang mengurus peristiwa yang ditembak oleh pelayan.
<i>AgentListener</i>	Kelas antara muka yang mengurus peristiwa yang ditembak oleh agen.
<i>AgentMessageListener</i>	Kelas antara muka yang mengurus mesej yang ditembak oleh pelayan atau agen.

Penghantaran dan penerimaan agen BDI-MU di antara satu set hos diuruskan oleh kelas *AgentBase*. Kelas *ReceiveMessageThread* dan *SendMessageThread* pula berfungsi untuk menghantar dan menerima mesej bagi tujuan komunikasi. Struktur data mesej pula dibentuk oleh kelas *AgentMessage*. Contoh penghantaran mesej melalui sambungan soket adalah seperti pada Rajah 12.9.

```
public void run() { // SendMessageThread
    Socket socket = null; ObjectOutputStream outStream = null;
    try {
        // bina sambungan soket ke hos destinasi dan tulis mesej ke dalam soket
        socket = new Socket(msg.recipient.host, msg.recipient.port);
        outStream = new ObjectOutputStream(socket.getOutputStream());
        outStream.writeObject(msg);
    }
    catch (IOException e) {}
}
```

**Rajah 12.9 : Menghantar mesej**

Terdapat beberapa operasi yang membantu agen BDI-MU bergerak di sepanjang rangkaian bagi menyempurnakan tugasnya dengan licin. Operasi tersebut adalah seperti pada Jadual 12.4 iaitu *onArrival()*, *onAwake()*, *onCreate()*, *onDestroy()*, *onDispatch()*, *onSleep()* dan *run()*.

**Jadual 12.4 : Operasi asas agen BDI-MU**

Operasi	Keterangan
<i>onArrival()</i>	Dilaksanakan sebaik sahaja agen sampai di sesebuah hos.
<i>onAwake()</i>	Dilaksanakan sebaik sahaja larian agen disambung semula oleh hos.
<i>onCreate()</i>	Dilaksanakan sebaik sahaja keadaan awalan agen ditakrifkan oleh hos yang menjananya.
<i>onDestroy()</i>	Dilaksanakan sebelum agen dihapuskan dari sesebuah hos.
<i>onDispatch()</i>	Dilaksanakan sebelum agen dihantar ke hos berikutnya.
<i>onSleep()</i>	Dilaksanakan sebelum larian agen digantung sementara oleh hos.
<i>run()</i>	Dilaksanakan sebaik sahaja agen dijana pada sesebuah hos.

Agen BDI-MU dibungkus ke dalam mesej untuk digerakkan di sepanjang rangkaian seperti pada Rajah 12.10. Sebelum dipindahkan, keadaan larian agen BDI-MU diberhentikan, kemudian bebenang baru dihasilkan bagi menghantar agen



BDI-MU ke hos berikutnya dan akhirnya menghapuskan agen BDI-MU dari hos semasa.

```
protected void dispatch(Agent agent, AgentAddress address) {
    // hentikan keadaan larian agen
    agent.onDispatch(); box = (AgentBox) boxes.get(agent.getName());
    if (box.thread.isAlive()) { box.thread = null; }
    // bungkus agen ke dalam mesej dan hantar ke hos berikutnya
    try {
        msgSender = new AgentAddress(baseAddress.host, port, agent.getName());
        bos = new ByteArrayOutputStream();
        mos = new AgentOutputStream(bos); mos.writeObject(agent);
        msg = new AgentMessage(msgSender, address, "BDIMU", "", bos.toByteArray());
        new SendMessageThread(msg).start();
    }
    catch (UnknownHostException e) {} catch (IOException e) {}
    // hapuskan agen dari hos semasa
    agent.onDestroy(); fireAgentLeft(agent.getName());
    // agen dimatikan sekiranya penghantaran gagal
    boxes.remove(agent.getName()); classManager.dec(agent.getName());
}
```

### Rajah 12.10 : Memindahkan agen BDI-MU

## 12.4 Implementasi BDI-MU kit dalam Pembangunan Sistem Pengurusan Inventori Teragih

Rajah 12.11 sehingga Rajah 12.18 pada Seksyen 12.4 menunjukkan contoh petikan paparan implementasi dan larian agen BDI-MU yang dibangunkan menggunakan BDI-MU kit untuk sistem pengurusan inventori teragih. Seksyen 12.5 pula membentangkan hasil pengujian dan penilaian keupayaan agen BDI-MU yang dibangunkan menggunakan BDI-MU kit.

Agan BDI-MU yang hendak dijana mestilah mengimport pakej BDI dan MobileUNITY supaya agen BDI-MU boleh menggunakan semula kelas-kelas yang terdapat di dalam pakej tersebut.

```
import BDIMU.MobileUNITY.*;
import BDIMU.MobileUNITY.MUAgent;
import BDIMU.MobileUNITY.AgentAddress;
import BDIMU.MobileUNITY.AgentEvent;
import BDIMU.BDI.*;
```

Agen BDI-MU juga perlu menyambung kepada kelas *MUAgent* supaya ia boleh mewarisi atribut dan operasi kelas tersebut.

```
public class StockTrackingAgent extends MUAgent { ... }
```

Memandangkan BDI-MU kit menyokong sistem pelbagai agen dan terdapat lebih daripada satu agen BDI-MU dalam satu set pelayan BDI-MU yang saling terangkai, maka setiap agen BDI-MU perlu mempunyai identiti unik untuk tujuan pengecaman dan identifikasi.

```
public StockTrackingAgent(String name) {  
    super("StockTrackingAgent_" + name); }
```

Setiap agen BDI-MU berupaya menjana beberapa agen pintar semasa ia berada pada sesuatu hos. Tujuannya adalah untuk pengkhususan tugas bagi mencapai matlamat secara efisien dan mengelakkan pembangunan agen yang monolitik.

```
argList[] = { "\\BDIMU\\BDIMUAgent\\SolutionFinderAgent.bdi" };  
intelligentAgent = new BDI();  
try { interpreter = new Interpreter(argList); } catch (Exception e) {}  
retVal = intelligentAgent.run(interpreter);
```

Agan pintar yang dijana pula boleh memanggil kelas Java semasa perlakuan pelan. Ia bagi membolehkan kerja-kerja kompleks yang tidak disokong oleh pentafsir BDI dilaksanakan melalui kod atur cara Java.

```
EXECUTE BDIMU.Agent.QueryStock $itemNum;  
EXECUTE BDIMU.Agent.PingHost $ipAddress $result;  
EXECUTE BDIMU.Agent.ListingOnLineHost;
```

Kontena di dalam agen BDI-MU mestilah mengimplementasi pengsirian bagi membolehkannya diangkut di sepanjang perjalanan bersama-sama dengan agen.

```
class BaseQueryResult implements Serializable {  
    public String itemnum;  
    public double stockbalance;  
    public double reorderqty;  
    public double duesinqty;  
    public double duesoutqty;  
    public double demandavg;  
    public double unitprice;  
}  
class BasesQueryResult implements Serializable {  
    public AgentAddress address;  
    public Vector result;  
}
```

Penjanaan agen BDI-MU yang terdapat di dalam pakej Agen dibuat oleh aplikasi melalui operasi *createAgent()*. Agen BDI-MU seterusnya dimuatkan ke dalam pelayan BDI-MU melalui operasi *loadAgent()*. Pada masa yang sama, pelayan BDI-MU akan mendaftarkan dan memberikan identiti kepada agen berkenaan.

```
void goAgent_actionPerformed(ActionEvent e) { // menjana agen
    createAgent("StockTrackingAgent"); }
void createAgent(String agent) { // memuatkan agen kepada pelayan
    loadAgent("BDIMUagent." + agent); }
```

Contoh matlamat agen Penentu Laluan yang akan menentukan stor-stor yang akan dijelajahi adalah seperti berikut:

```
GOALS:
    PERFORM BuatPenentuanLaluan;
    ACHIEVE JanaDirektoriLaluan : UTILITY 10;
    MAINTAIN StatusRangkaian "Yes";
```

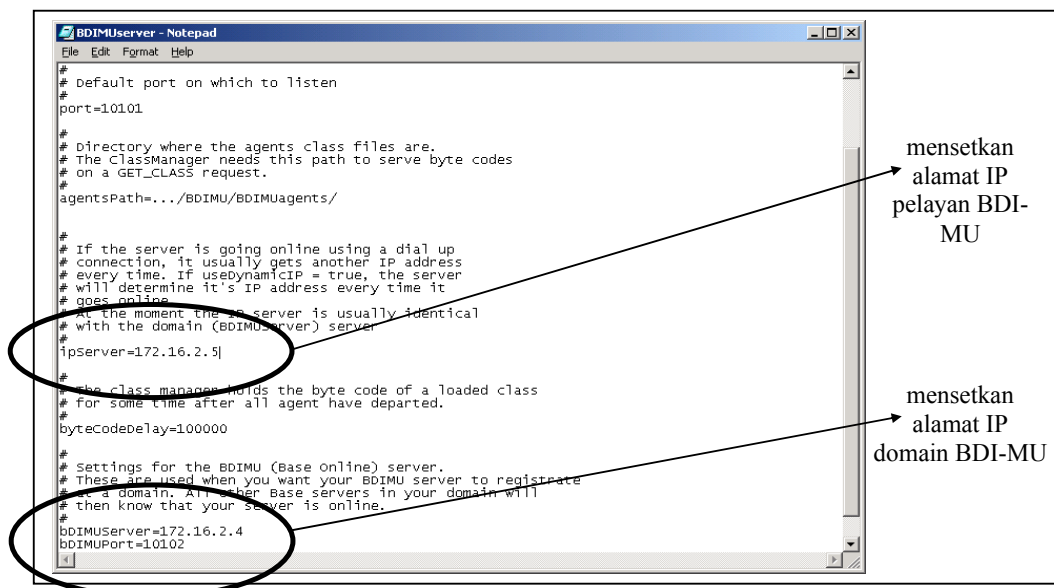
Contoh kepercayaan yang dipegang oleh agen Penanya Stok yang akan mencapai pangkalan data adalah seperti berikut:

```
FACTS:
    FACT material "01-047-134" "GASKET" 0 "EQUIPMENT";
    FACT buat_permohonan "false";
    FACT hos_sumber "01" "161.139.67.46";
```

Contoh pelan agen Pemohon Stok yang akan membuat permohonan stok di setiap stor yang dijelajahi adalah seperti berikut:

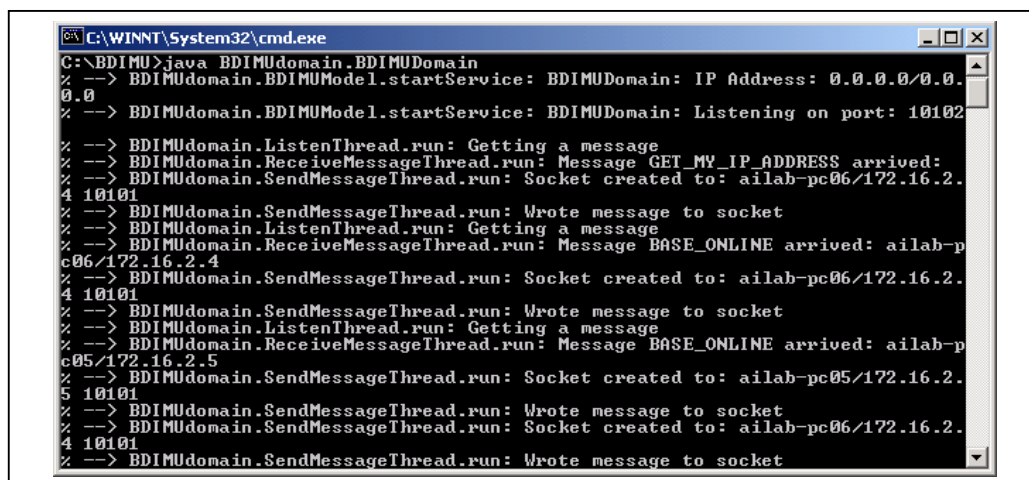
```
PLAN: {
    NAME: "Pelan 1: Tanya Stok"
    GOAL: ACHIEVE pertanyaan_dibuat $soalan $keputusan $masalah;
    BODY:
        EXECUTE BDIMUagent.GetDBConnStatus.execute $DBConn
        EXECUTE print "DB Connection Status: " $DBConn "\n"; OR
        { TEST (pertanyaanDipenuhi $soalan $jawapan); }
        { EXECUTE buatPertanyaan $soalan $keputusan;
          ACHIEVE pertanyaan_dibuat $soalan $keputusan "true"; };
    WHEN: TEST (== $masalah "false")
        { EXECUTE BDIMUagent.StoredApproval.execute
          $soalan $keputusan };
}
```

Setiap pelayan BDI-MU perlu berdaftar dengan domain BDI-MU yang akan mengurus satu set pelayan BDI-MU. Rajah 12.11 menunjukkan paparan konfigurasi BDI-MU.



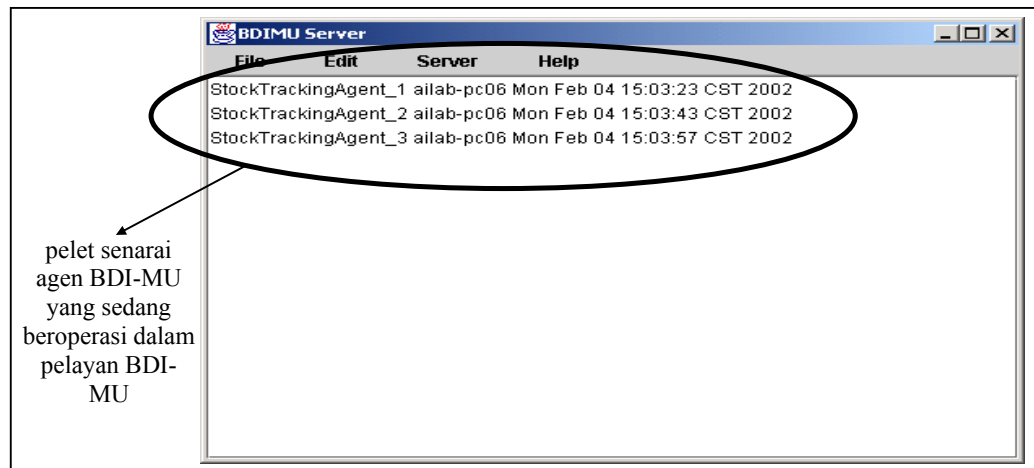
**Rajah 12.11 : Paparan konfigurasi BDI-MU**

Setiap kali terdapat pelayan BDI-MU yang berstatus dalam talian, maka domain BDI-MU akan menghebahkan mesej *BASE\_ONLINE* kepada setiap pelayan lain. Rajah 12.12 menunjukkan paparan domain BDI-MU.



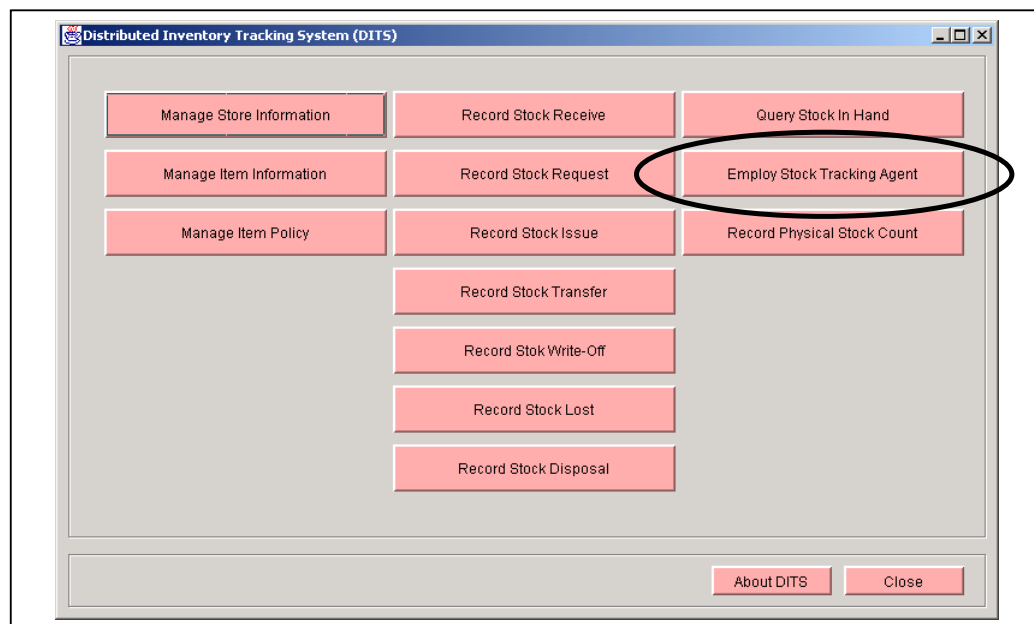
**Rajah 12.12 : Paparan domain BDI-MU**

Setiap agen BDI-MU yang beroperasi di dalam sesuatu pelayan BDI-MU akan didaftarkan. Tujuannya adalah untuk memudahkan proses pemantauan dan pengurusan yang dibuat oleh pelayan BDI-MU. Rajah 12.13 menunjukkan paparan pelayan BDI-MU.



**Rajah 12.13 : Paparan pelayan BDI-MU**

Skrin utama sistem pengurusan inventori teragih adalah seperti pada Rajah 12.14. Aplikasi berupaya melantik agen Penjejak Stok untuk menjejaki stok yang dikehendaki berdasarkan keperluan tertentu pada satu set stor yang saling terangkai.



**Rajah 12.14 : Paparan skrin utama sistem pengurusan inventori teragih**

Pengguna perlu menginput beberapa parameter keperluan kepada agen Penjejak Stok bagi membolehkannya melaksanakan tugas sebagaimana yang dipaparkan pada Rajah 12.15.

**Rajah 12.15 : Paparan menginput keperluan kepada agen penjejak stok**

Contoh paparan permohonan stok yang dibuat oleh agen Pemohon Stok ke atas stor-stor yang mempunyai lebih stok untuk tujuan pembekalan sokongan adalah seperti pada Rajah 12.16.

**Rajah 12.16 : Paparan meluluskan permohonan stok yang dibuat oleh agen penjejak stok**

Contoh langkah-langkah yang dilaksanakan, bilangan *apl* yang dijana, bilangan matlamat yang dicapai dan bilangan pusingan pentafsir BDI serta masa larian pelaksanaan proses adalah seperti pada Rajah 12.17.

```

BDIMServer
JAM Parser Version 0.61 + 0.79i:
JAM definition parse successful.
Starting to execute.
Check database connection ...
Query stock ...
Stored result ...
Plan requisition ...
Request stock ...
Stored result ...

JAM: All of the agent's top-level goals have been achieved! Exiting...

Runtime statistics follow:
Number of APLs generated: 9
Number of Null APLs: 7
Number of Goals established: 2
Number of interpreter cycles: 7

APL generation time: 0.0 seconds.
Intending time: 0.0 seconds.
Plan execution time: 0.01 seconds.
Observer execution time: 0.0 seconds.
Total run time: 0.01 seconds.
% --> BDIMUagent.StockTrackingAgent.onCreate: Agent returned: 0

```

**Rajah 12.17 : Paparan pelaksanaan pentafsir BDI**

Contoh keputusan yang dicadangkan oleh agen Penjejak Stok bagi proses penjejakan inventori teragih adalah seperti pada Rajah 12.18. Faktor yang diambil kira oleh agen Penjejak Stok adalah jarak, kuantiti yang diluluskan dan tarikh boleh diperolehi.

Stock Tracking Result										
Item Reference No.	504876-4		FUSE-05 AMP							
Required Quantity	7,000		Required Date 04-Feb-2002							
Priority	02		IMPORTANT							
Request By	824662		LM PPP ZAHIR BIN ZAKARIYA							
#	Store Code	Description	Distance	Qty Available	Demand No.	Qty Approved	Date of Availability	Approved By	Name	Solution
1	DBA	DEPOT BANTUAN ARN	367	100.000	000000000001002	4.000	01-Feb-2002	1032547	SJIN KKD NOC	
2	DEML01	DEPOT BANTUAN WIL	766	25.000	0000000000035013	7.000	02-Feb-2002	818543	LKI PPP FAIZ	
3	DEML02	DEPOT BANTUAN WIL	320	56.000	000000000100971	5.000	02-Feb-2002	820244	LKI PTI RIDZI	Transfer 5 items
4	SKN21-FRIG02	KD RAHMAT	90	2.000	0000000000000116	0.000		823881	LKII PPP NOR	
5	SKN22-CORV01	KD KASTURI	599	5.000	0000000000000233	0.000		821530	LKII TLR MOH	
6	SKN31-MPCSS01	KD SRI INDERAPURA	114	18.000	0000000000000878	4.000	01-Feb-2002	824347	LM STN(L) MK	
7	SKN31-MPCSS02	KD SRI INDERA SAKTI	98	12.000	0000000000000749	1.000	29-Jan-2002	812477	LK BDI ROSLI	Transfer 1 item
8	SKN31-MPCSS03	KD MAHAWANGSA	34	9.000	0000000000000394	1.000	26-Jan-2002	822445	APP TMK SUH	Transfer 1 item
				Total Qty Available	277.000		Total Qty Approved	22.000		
Close										

**Rajah 12.18 : Paparan penyelesaian yang dicadangkan oleh agen penjejak stok**

## 12.5 Keputusan Implementasi BDI-MU kit dalam Pembangunan Sistem Pengurusan Inventori Teragih

Pengujian agen BDI-MU dalam sistem pengurusan inventori teragih dijalankan dalam persekitaran pengkomputeran seperti pada Jadual 12.5. Sebanyak empat buah komputer digunakan untuk menyimulasikan dua buah stor pangkalan iaitu Depot Bantuan Armada (DBA) dan Depot Bantuan Wilayah Laut 1 (DBWL01) serta dua buah stor kapal iaitu KD Sri Inderapura (SKN31-MPCSS01) dan KD Kasturi (SKN22-CORV01). Setiap stor mempunyai sistem pengurusan inventori setempat dan satu set pangkalan data inventori tersendiri sepertimana yang telah dibentangkan dalam Seksyen 2.4.

**Jadual 12.5 : Persekitaran pengujian**

No	Spesifikasi Komputer	SKN31-MPCSS01	DBA	DBWL01	SKN22-CORV01
1.	Pemproses	Pentium III 500 MHz	Pentium III 500 MHz	Pentium III 450 MHz	Pentium III 500 MHz
2.	Ingatan	256 MB	256 MB	256 MB	128 MB
3.	Kad rangkaian	10/100 MBps	10/100 MBps	10/100 MBps	10/100 Bps
4.	Sistem pengoperasian	Red Hat Linux 7.2	Windows 2000 Server	Windows XP Professional	Windows 2000 Server
5.	Pangkalan data	MySQL 4.0	Oracle 8i	SQL Server 2000	Access 2002
6.	Perisian antara muka	JBuilder 3.0	JBuilder 3.0	JBuilder 3.0	JBuilder 3.0
7.	Mesin maya Java	JDK 1.3.1	JDK 1.3.1	JDK 1.3.1	JDK 1.2.2

Data pengujian yang digunakan adalah data-data yang diperolehi daripada Institut Teknologi Perisian (ITP), Universiti Teknologi Malaysia (UTM) dengan kebenaran TLDM. Ia merupakan data sebenar yang dimuat turun pada tahun 2001 bagi sistem pengurusan inventori TLDM yang menggunakan perisian MAXIMO. MAXIMO adalah sistem bukan berasaskan agen yang dibangunkan oleh Lincoln Technology Corporation yang mengandungi tiga modul utama iaitu senggaraan, inventori dan belian. Bagaimanapun, MAXIMO tidak mempunyai fungsi-fungsi yang diperlukan untuk menangani masalah sistem pengurusan inventori teragih di TLDM sebagaimana yang telah dikemukakan dalam Seksyen 2.2. Maklumat bagi data-data sistem pengurusan inventori teragih adalah seperti pada Jadual 12.6. Jarak



dikira daripada stor yang menjana agen Penjejak Stok iaitu stor kapal KD Hang Tuah (SKN21-FRIG01) ke stor-stor yang telah dinyatakan dalam Jadual 12.5. Bagi stor SKN31-MPCSS01 dan SKN22-CORV01, jarak diukur dari SKN21-FRIG01 ke pelabuhan kedua-dua kapal tersebut berlabuh.

**Jadual 12.6 : Maklumat bagi data-data sistem pengurusan inventori teragih**

No	Item	SKN31-MPCSS01	DBA	DBWL01	SKN22-CORV01
1.	Bilangan material	21,508	258,096	101,191	8,433
2.	Bilangan material <b>5H40-2812</b> dalam pegangan	15	56	39	13
3.	Bilangan pesanan semula material <b>5H40-2812</b>	10	50	45	10
4.	Bilangan material <b>VL11</b> dalam pegangan	46	108	71	29
5.	Bilangan pesanan semula material <b>VL11</b>	15	80	50	15
6.	Bilangan material <b>GGGB490</b> dalam pegangan	54	88	178	37
7.	Bilangan pesanan semula material <b>GGGB490</b>	25	150	120	25
8.	Bilangan material <b>1509-16</b> dalam pegangan	7	35	42	21
9.	Bilangan pesanan semula material <b>1509-16</b>	15	60	50	15
10.	Bilangan material <b>41F1614</b> dalam pegangan	39	199	144	52
11.	Bilangan pesanan semula material <b>41F1614</b>	40	180	135	40
12.	Bilangan material <b>99584</b> dalam pegangan	17	24	31	21
13.	Bilangan pesanan semula material <b>99584</b>	5	35	20	5
14.	Jarak dari SKN21-FRIG01	114 KM	367 KM	766 KM	599 KM

Agan Penjejak Stok yang dijana oleh SKN21-FRIG01 mempunyai bilangan kepercayaan, pelan dan matlamat seperti pada Jadual 9.1. Sampel material yang dicari adalah *SPRING-MED 1 5-16 OD UNL (5H40-2812)*, *SOCKET-SKT WR 1-4 SQ DR (VL11)*, *BLOCK-MTL 12IN SNH R25 (GGGB490)*, *HOSE-RUB 1.000 ID HIP (1509-16)*, *FILE-HND WRD 6IN (41F1614)* dan *ORDALT 30296 MK15MOD1 WEAPON SYSTEM CLOSE-IN (99584)*. Contoh keputusan yang dijana oleh agan Penjejak Stok adalah seperti pada Jadual 12.7.

**Jadual 12.7 : Contoh penyelesaian yang dicadangkan oleh agen penjejak stok**

No	Material	Keperluan Pengguna			Stor	Kelulusan		Jarak (km)	Cadangan
		Kuantiti	Tarikh	Keutamaan		Kuantiti	Tarikh		Penyelesaian
1.	5H40-2812	7	25-01-02	02 - Penting	SKN31-MPCSS01	4	24-01-02	114	Ambil 4 item
2.					DBA	4	24-01-02	367	Ambil 3 item
3.					DBWL01	7	25-01-02	766	-
4.					SKN22-CORV01	3	22-01-02	599	-
5.	VL11	16	16-03-02	02 - Penting	SKN31-MPCSS01	12	15-03-02	114	Ambil 12 item
6.					DBA	11	18-03-02	367	-
7.					DBWL01	9	15-03-02	766	-
8.					SKN22-CORV01	8	16-03-02	599	Ambil 4 item
9.	GGGB490	27	05-03-02	02 - Penting	SKN31-MPCSS01	15	08-03-02	114	-
10.					DBA	21	10-03-02	367	-
11.					DBWL01	27	04-03-02	766	Ambil 27 item
12.					SKN22-CORV01	10	05-03-02	599	-
13.	1509-16	8	11-04-02	02 - Penting	SKN31-MPCSS01	0	-	114	-
14.					DBA	5	11-04-02	367	Ambil 2 item
15.					DBWL01	6	09-04-02	766	Ambil 6 item
16.					SKN22-CORV01	5	10-04-02	599	-
17.	41F1614	35	27-05-02	02 - Penting	SKN31-MPCSS01	5	26-05-02	114	-
18.					DBA	13	27-05-02	367	Ambil 13 item
19.					DBWL01	15	27-05-02	766	Ambil 13 item
20.					SKN22-CORV01	9	25-05-02	599	Ambil 9 item
21.	99584	10	07-02-02	02 - Penting	SKN31-MPCSS01	8	10-02-02	114	-
22.					DBA	5	07-02-02	367	Ambil 5 item
23.					DBWL01	7	05-02-02	766	-
24.					SKN22-CORV01	7	07-02-02	599	Ambil 5 item

Sebelum agen Penjejak Stok mengemukakan laporan seperti pada Jadual 12.7 kepada penggunanya, ia akan melaksanakan proses penaakulan secara berulang kali sehingga semua matlamatnya tercapai. Proses penaakulan ini dibuat oleh pentafsir BDI berdasarkan kepada arahan-arahan atur cara di dalam gelung utama pentafsir BDI seperti pada Rajah 12.7. Agen Penjejak Stok akan melarikan pelan yang bersesuaian sebagai tindakan berpandukan kepada niat, pelan, matlamat dan kepercayaan mengenai situasi semasanya. Proses penaakulan ini membolehkan agen Penjejak Stok melaksanakan pelan-pelan seperti bergerak ke sesuatu stor yang disasarkan, menyemak kesegaran rangkaian, mendapatkan data-data yang diperlukan dari pangkalan data, berkomunikasi dengan agen atau entiti lain, membuat permohonan stok dan sebagainya. Maklumat-maklumat yang telah dikumpul membantu agen Penjejak Stok menjana penyelesaian seperti pada Jadual 12.7. Maklumat dan cadangan penyelesaian yang dikemukakan oleh agen Penjejak Stok membolehkan penggunanya menentukan stor yang sesuai untuk mendapatkan material yang dicari dan seterusnya memanipulasikan maklumat-maklumat tersebut untuk melicinkan lagi proses pengurusan inventori teragih. Selain itu, kelebihan lain yang diperolehi dengan menggunakan agen ini ialah memudahkan kerja-kerja mengidentifikasi inventori terperap; mengenal pasti kedudukan dan pergerakan inventori di antara stor; membuat telahan dan kawalan inventori; menentukan bajet penyimpanan dan pengendalian inventori; mempercepatkan proses pembekalan inventori kepada pemohon; dan sebagainya.

Pengujian juga dijalankan untuk menilai kelakuan autonomi agen Penjejak Stok. Jadual 12.8 memaparkan perbandingan di antara agen Penjejak Stok\_1 yang mempunyai aras autonomi yang lebih tinggi berbanding dengan agen Penjejak Stok\_2. Di mana bagi agen Penjejak Stok\_1,  $\lambda_{goal} = 9$ ,  $\lambda_{bel} = 9$ ,  $\lambda_{cap} = 9$  dan  $\lambda_{int} = 5$ . Bagi agen Penjejak Stok\_2 pula,  $\lambda_{goal} = 1$ ,  $\lambda_{bel} = 1$ ,  $\lambda_{cap} = 1$  dan  $\lambda_{int} = 3$ .

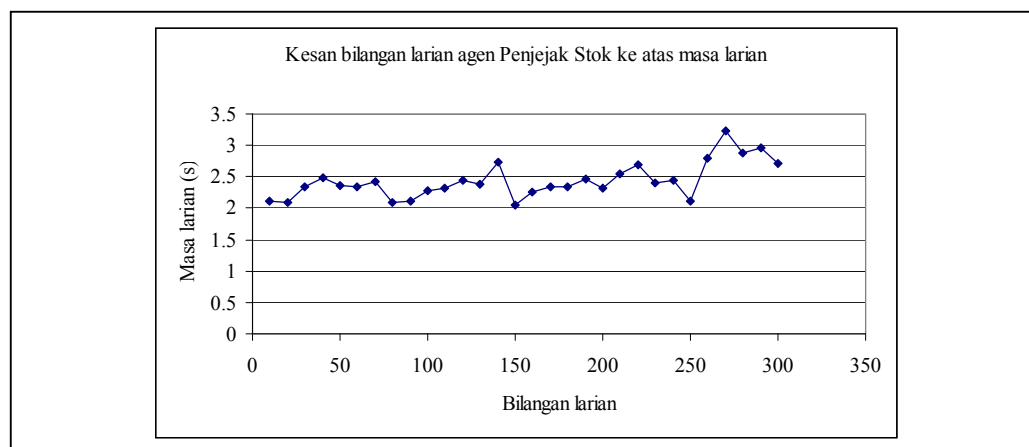
**Jadual 12.8 : Kelakuan agen penjejak stok dengan aras autonomi yang berbeza**

No	Kelakuan	Agen Penjejak Stok_1	Agen Penjejak Stok_2
1.	Semakan pengetahuan	88	27
2.	Semakan matlamat	15	7
3.	Hebahan mesej	35	15
4.	Pergerakan di antara stor	8	5
5.	Kitaran pentafsir BDI	27	10

Maklumat yang dipaparkan pada Jadual 12.8 menunjukkan bahawa paras autonomi yang tinggi akan mengakibatkan agen Penjejak Stok lebih aktif, berwaspada dan saling bekerjasama. Pengujian pelaksanaan pula dilakukan untuk menilai keupayaan taakulan, autonomi, mobiliti dan komunikasi agen Penjejak Stok. Penilaian dikaji terhadap kesan masa larian agen Penjejak Stok. Empat kes eksperimen telah dijalankan sebagaimana yang telah dibentangkan dalam Bab IX dan Bab X, iaitu:

- (a) Masa larian bagi pusingan pentafsir BDI (Rajah 9.3).
- (b) Masa larian bagi saringan autonomi (Rajah 9.4).
- (c) Masa larian bagi pemindahan agen Penjejak Stok di antara dua buah komputer (Rajah 10.6).
- (d) Masa larian bagi hebahan mesej di antara dua agen Penjejak Stok (Rajah 10.7).

Masa keseluruhan yang diambil oleh agen Penjejak Stok untuk melaksanakan proses penjejukan material seperti pada Jadual 12.7 ditunjukkan oleh graf kesan larian pada Rajah 12.19. Keputusan yang diperolehi mendapati larian agen Penjejak Stok sebanyak 300 larian adalah stabil dan munasabah kerana masa yang diperlukan hanya di antara 2 – 3 saat sahaja.



**Rajah 12.19 : Graf kesan bilangan larian agen penjejak stok ke atas masa larian**

Kesimpulan yang boleh dibuat daripada keputusan implementasi agen Penjejak Stok yang dibangunkan menggunakan BDI-MU kit untuk sistem pengurusan inventori teragih menunjukkan bahawa agen Penjejak Stok dapat

menghasilkan output seperti yang dikehendaki dan tidak bertindak sebaliknya untuk mengagalkan matlamatnya. Hasil pengujian turut mendapati agen Penjejak Stok mengambil masa yang munasabah untuk menyelesaikan tugasnya dalam persekitaran pengkomputeran teragih dan heterogen.

## **12.6 Contoh Implementasi BDI-MU kit dalam Lain-Lain Aplikasi**

Selain diimplementasi ke atas sistem pengurusan inventori teragih untuk TLDM, BDI-MU kit juga telah digunakan bagi membangunkan pelbagai jenis sistem agen seperti berikut:

- (a) Sistem pengurusan tabung darah teragih (Amiza Amir, 2002) – agen berfungsi menguruskan satu set bank darah yang teragih.
- (b) Sistem pengurusan inventori ladang teragih (Hashimah Samion et al., 2002) – agen berfungsi menguruskan inventori ladang yang teragih.
- (c) Sistem pengurusan pelayan teragih (Ezatul Suzarafah Taib et al., 2002) – agen berfungsi menguruskan inventori perkakasan kafeteria siber yang teragih.
- (d) Sistem pengurus aplikasi rangkaian (Izahmunirah Mohammed et al., 2002) – agen berfungsi menguruskan pemasangan perisian di makmal komputer yang teragih.

## **12.7 Perbandingan BDI-MU kit dengan Lain-Lain Alatan Pembangunan Agen**

Perbandingan juga dibuat untuk melihat perbezaan ciri-ciri di antara BDI-MU kit dengan lain-lain alatan pembangunan agen sama ada yang berasaskan Java atau lain-lain bahasa. Alatan pembangunan agen yang dipilih untuk perbandingan adalah JADE, IBM Aglets, JAFMAS, dMARS dan INTERRAP. Di antara ciri-ciri perbandingan yang diambil kira adalah bahasa pengaturcaraan yang digunakan; sistem pengoperasian yang menyokong larian agen; adakah ia menyokong

pembangunan agen berautonomi, cerdik dan berupaya bergerak; protokol komunikasi yang digunakan; corak seni bina, struktur agen dan ciri-ciri keselamatan; dan adakah ia menyokong pembangunan sistem pelbagai agen serta mempunyai metodologi pembangunan agen yang khusus. Jadual 12.9 membandingkan ciri-ciri BDI-MU kit dengan alatan pembangunan agen yang lain. Hasil perbandingan ini menunjukkan bahawa BDI-MU kit mempunyai ciri-ciri minima untuk menyokong pembangunan agen mobil pintar yang teragih, serempak, cerdik dan berupaya bergerak dalam persekitaran pengkomputeran teragih dan heterogen. BDI-MU kit mempunyai ciri-ciri penting ini kerana ia disokong oleh metodologi BDI-MU dan seni bina BDI-MU yang membolehkan agen mobil pintar dibangunkan secara sistematik bermula daripada penganalisan, penukaran kepada reka bentuk sehingga penjelmaan kepada kod-kod atur cara serta boleh dilarikan dalam persekitaran teragih dan heterogen.

**Jadual 12.9 : Perbandingan BDI-MU kit dengan lain-lain alatan pembangunan agen**

<b>Ciri-Ciri</b> \ <b>Alatan</b>	<b>BDI-MU kit</b>	<b>JADE</b>	<b>IBM Aglets</b>	<b>JAFMAS</b>	<b>dMARS</b>	<b>INTERRAP</b>
Bahasa pembangunan	Java	Java	Java	Java	C++	Oz
Sistem pengoperasian	Semua	Semua	Semua	Semua	Unix	Unix
Mempunyai metodologi pembangunan	Metodologi BDI-MU	Tiada	Tiada	Metodologi JAFMAS	Metodologi BDI	Tiada
Corak seni bina	Heterogen	Berlapis	Berlapis	Berlapis	Pentafsir	Berlapis
Protokol komunikasi	Mesej	FIPA	ATCI	Tindakan penuturan	Mesej	ACL
Agensi mempunyai autonomi	Ya	Ya	Ya	Ya	Ya	Tiada
Agensi mempunyai kecerdikan	Ya	Ya	Tiada	Tiada	Ya	Ya
Konsep kecerdikan agensi	BDI	Ontologi	Tiada	Tiada	BDI	Perancangan
Agensi berupaya bergerak	Ya	Ya	Ya	Ya	Ya	Tidak
Konsep pergerakan agensi	Mobile UNITY	RMI	RMI	RMI	RPC	Tiada
Struktur agensi	Berhirarki	Monolitik	Monolitik	Monolitik	Monolitik	Monolitik
Menyokong sistem pelbagai agensi	Ya	Ya	Tidak	Tidak	Ya	Tidak
Ciri-ciri keselamatan	Berasaskan keselamatan Java	Berasaskan keselamatan Java	Berasaskan keselamatan Java	Berasaskan keselamatan Java	Tiada	Tiada

## 12.8 Ringkasan

Kejayaan pembangunan BDI-MU kit adalah kerana Java digunakan untuk menterjemahkan seni bina agen mobil pintar kepada kod atur cara yang boleh dilarikan. Berbanding dengan lain-lain bahasa, Java adalah lebih sesuai kerana ia menyokong pembangunan komponen berautonomi, cerdik dan bergerak. Selain itu, kod baitnya boleh dilarikan pada persekitaran pengkomputeran teragih dan heterogen. Keupayaan sedemikian membolehkan Java menyokong pelbagai fungsi kritikal dan penting yang perlu dimiliki oleh sesuatu agen. Ini dibuktikan dengan penggunaannya dalam membangunkan agen BDI-MU untuk menyelesaikan masalah-masalah sistem pengurusan inventori teragih. Fakta ini diperkukuhkan lagi dengan hasil pengujian dan pengukuran yang telah dibentangkan dalam Seksyen 12.5 yang membuktikan kemampuan dan keupayaan agen BDI-MU yang dibangunkan menggunakan BDI-MU kit yang disokong oleh seni bina BDI-MU dan metodologi BDI-MU. Seni bina BDI-MU yang merupakan tunjang kajian ini telah dibuktikan fleksibiliti dan keterbukaannya melalui implementasi ke atas pelbagai domain aplikasi teragih seperti yang telah dinyatakan dalam Seksyen 12.6. Melalui implementasi tersebut, kesahihan fungsi-fungsi seni bina BDI-MU yang membolehkannya menyokong larian agen mobil pintar disahkan dengan menilai output yang dihasilkan oleh agen BDI-MU bagi setiap senario input yang dimasukkan. Malahan, melalui metodologi BDI-MU seperti yang telah dibincangkan dalam Bab V, kerjasama dan komunikasi di antara pembangun sistem agen dengan pengguna sistem agen memainkan peranan penting untuk mengesahkan pelaksanaan dan kefungsian seni bina BDI-MU dan larian agen BDI-MU.

## **BAB XIII**

### **KESIMPULAN**

#### **13.1 Pengenalan**

Dalam era globalisasi, bidang sains komputer sedang mengalami krisis perisian yang meluas ekoran dari perkembangan domain bio-informatik dan pengetahuan. Fenomena ini membawa implikasi kepada bidang kejuruteraan perisian dan kepintaran buatan berasaskan agen. Teras kejuruteraan perisian adalah kaedah dan alatan untuk membina perisian. Teras kepintaran buatan pula adalah teknik untuk menjadikan perisian pintar. Kedua-duanya diperlukan untuk membantu jurutera perisian memodelkan komponen dunia nyata ke dalam bentuk perisian. Oleh itu, penyelidikan intensif dalam bidang tersebut diperlukan untuk memastikan perisian berasaskan agen dapat menyokong pengeluaran produk generik atau langganan.

Objektif bab ini adalah untuk menerangkan sumbangan yang diperolehi dan mencadangkan kerja masa hadapan yang boleh disambung dari kajian yang telah dijalankan. Selain itu, rumusan daripada pelaksanaan kajian ini turut dibentangkan.

#### **13.2 Kesimpulan**

Perkembangan teknologi rangkaian dan kewujudan organisasi dengan jaringan yang besar telah merancakkan pembangunan aplikasi teragih. Ini kerana



aplikasi teragih diperlukan untuk mengendalikan dan mengurus data, sumber dan kepakaran organisasi yang teragih pada pelbagai lokasi dengan struktur geografi dan topologi yang berbeza. Pengagihan ini bertambah kompleks jika ia melibatkan persekitaran pengkomputeran yang heterogen dan jumlah data atau sumber yang banyak. Data, sumber dan kepakaran yang teragih ini perlu diurus secara cekap kerana ia merupakan aset kepada sesebuah organisasi. Pengurusan yang cekap dapat mengurangkan kos dan meningkatkan produktiviti pengeluaran.

TLDM merupakan sebuah organisasi yang besar dan mempunyai rangkaian stor-stor yang berselerak di seluruh negara. Setiap stor menyimpan pelbagai material dengan bilangan, harga dan kos simpanan yang tertentu. Pengurusan material yang cekap dan berkesan merupakan salah satu faktor utama yang menentukan tahap kesiapsiagaan kapal TLDM. Material adalah sumber penting kepada kerja-kerja senggaraan dan pembaikan kapal TLDM. Oleh kerana itu, sistem pengurusan inventori teragih diperlukan untuk mengendalikan material supaya kerja-kerja pengurusan data dan maklumat material serta pengawasan dan kawalan material di antara stor dapat dilaksanakan dengan lancar. Pengurusan material yang lancar dapat mengurangkan kos senggaraan dan meningkatkan peratus kesiapsiagaan kapal.

Agen mobil pintar adalah mekanisme penyelesaian yang mampu mengendalikan masalah yang wujud di dalam aplikasi teragih. Peranannya dapat membantu meningkatkan tahap pengurusan material di TLDM. Bagaimanapun, teknologi sedia ada perlu diperbaiki agar ia dapat digunakan untuk meningkatkan kemampuan agen mobil pintar. Di dalam kajian ini, tiga elemen pembangunan agen mobil pintar telah dikenal pasti sebagai faktor kepada kebolehpayaan agen mobil pintar. Elemen-elemen tersebut telah diselidiki, dimajukan dan diimplementasi ke atas agen mobil pintar agar ia mempunyai keupayaan maksima untuk mengendalikan sistem pengurusan inventori teragih TLDM.

Elemen pertama adalah metodologi pembangunan sistem berasaskan agen mobil pintar. Metodologi diperlukan untuk menyediakan peraturan dan amalan yang sistematik bagi mengurangkan risiko kos dan masa pembangunan. Di dalam kajian ini, metodologi ringan dan kejuruteraan serempak telah digabungkan untuk membina metodologi BDI-MU. Melalui perlaksanaan langkah-langkah kejuruteraan keperluan,

pemodelan domain dan pemodelan agen, reka bentuk seni bina, pengekodan, penggabungan dan transisi ia telah digunakan untuk membangunkan sistem pengurusan inventori teragih. Pemodelan yang merupakan langkah kritikal dalam pembangunan sistem agen mobil pintar telah diberi perhatian khusus dengan menghasilkan teknik pemodelan BDI-MU. Teknik pemodelan ini memodelkan sistem agen mobil pintar daripada tiga sudut pandangan yang berbeza iaitu pandangan domain, pandangan intra agen dan pandangan inter agen. Notasi UML telah digunakan untuk memvisualisasikan pandangan-pandangan tersebut. Hasil implementasi metodologi BDI-MU dan teknik pemodelan BDI-MU ke atas pembangunan sistem pengurusan inventori teragih telah berjaya menghasilkan sistem agen mobil pintar yang boleh dipercayai dan tegap serta dapat dibangunkan dalam jangka masa yang pendek dengan risiko yang rendah.

Elemen kedua adalah seni bina agen mobil pintar yang diperlukan untuk menghuraikan struktur dan komponen sistem agen mobil pintar. Di dalam kajian ini, corak seni bina heterogen dan model pandangan “4+1” digunakan untuk menghuraikan seni bina BDI-MU. Seni bina BDI-MU adalah seni bina terbuka yang dihasilkan untuk membentuk struktur dan komponen sistem agen mobil pintar dalam persekitaran teragih. Seni bina BDI-MU dibentuk oleh dua struktur utama iaitu struktur intra agen dan struktur inter agen. Struktur intra agen menyediakan komponen untuk agen mobil pintar membuat taakulan. Struktur inter agen pula menyediakan komponen untuk agen mobil pintar berkomunikasi dan bergerak di antara mesin. Struktur intra agen dibentuk oleh seni bina BDI dan autonomi. Gabungan kedu-duanya telah berjaya menghasilkan agen mobil pintar yang berupaya menaakul secara praktikal dan berautonomi. Struktur inter agen pula dibentuk oleh mesej dan konsep bahasa Mobile UNITY. Integrasi ini telah berjaya menghasilkan agen mobil pintar yang berkeupayaan bersosial dan saling bekerjasama. Hasilnya, agen mobil pintar yang dibina berdasarkan struktur intra dan inter agen telah berjaya mengendalikan masalah-masalah di dalam pengurusan inventori teragih di TLDM.

Seni bina BDI-MU yang dihasilkan dalam kajian ini disokong oleh rangka kerja organisasi holonik dan hirarki ambients yang digunakan untuk membangunkan agen mobil pintar yang berkomponen dan mirip kepada organisasi sebenar. Agen mobil pintar yang dihasilkan adalah berlandaskan kepada peranan dan tugas-tugas

yang wujud dalam organisasi pengurusan inventori di TLDM. Hirarki ambients pula telah berjaya meningkatkan kemampuan agen mobil pintar memindahkan data dan maklumat material di sepanjang pergerakannya dalam rangkaian.

Komponen-komponen yang telah direka bentuk bagi menyokong struktur intra dan inter agen mobil pintar dapat dibuktikan kebenaran dan konsistensinya dengan cara menformalisasikan reka bentuk berkenaan. Tujuannya adalah untuk memastikan ianya lengkap dan betul. Oleh itu, spesifikasi formal telah dibangunkan menggunakan Z untuk membuktikan kebenaran sintaks dan semantiknya. Hasil daripada pembangunan spesifikasi Z secara lelaran telah berjaya menghasilkan reka bentuk yang konsisten dan dapat memudahkan kerja-kerja pengkodan.

Elemen ketiga adalah alatan pembangunan agen mobil pintar. Alatan pembangunan ini diperlukan untuk mengekod agen mobil pintar dan menyediakan infrastruktur lariannya. Di dalam kajian ini, alatan pembangunan agen mobil pintar yang dipanggil BDI-MU kit yang disokong oleh seni bina BDI-MU dan metodologi BDI-MU telah dibangunkan menggunakan Java. Hasilnya, agen mobil pintar yang inter operasi, cerdik dan berupaya bergerak di antara mesin telah berjaya dibangunkan menggunakan BDI-MU kit. Agen ini telah ditempatkan ke dalam sistem pengurusan inventori teragih di TLDM dan berjaya melaksanakan tugasnya dengan sempurna sebagaimana matlamat yang telah ditetapkan. Hasil larian sistem pengurusan inventori teragih juga mendapati bahawa agen BDI-MU yang dihasilkan melalui ketiga-tiga elemen pembangunan ini telah berjaya mengendalikan masalah-masalah yang wujud dengan pelaksanaan dan kemampuan yang terbaik dan munasabah. Selain itu, ia juga telah diimplementasi di dalam lain-lain aplikasi teragih seperti pengurusan bank darah teragih, sistem pengurusan pelayan teragih, sistem pengurus aplikasi rangkaian dan sistem pengurusan inventori ladang teragih.

### **13.3 Sumbangan Kajian**

Sumbangan kajian ini secara umumnya ialah seni bina terbuka untuk aplikasi teragih berasaskan agen mobil pintar yang berciri unik telah berjaya dibangunkan. Ia

menggabungkan ciri-ciri mentalistik (matlamat, kepercayaan, keinginan dan niat) dengan keupayaan fizikalistik (komunikasi dan mobiliti). Selain itu, komponennya boleh diguna semula untuk pelbagai domain dan dapat dilarikan dalam pelbagai persekitaran. Jadual 13.1 memaparkan sumbangan kajian yang diperolehi secara terperinci berdasarkan kepada isu-isu penting yang dikenal pasti semasa menjalankan kajian ini.

**Jadual 13.1 : Sumbangan kajian**

No	Isu	Matlamat	Sumbangan	Bab
1.	Masalah memahami agen mobil pintar dan aplikasi teragih.	Menghasilkan spesifikasi masalah aplikasi teragih dan kajian literatur agen mobil pintar.	Spesifikasi masalah dan kajian literatur.	Bab II dan Bab III.
2.	Masalah membangunkan sistem agen mobil pintar.	Menghasilkan metodologi yang dapat mengendalikan kompleksiti dan kekangan masa pembangunan sistem agen mobil pintar.	Metodologi BDI-MU.	Bab V.
3.	Masalah memodelkan dan menganalisa agen mobil pintar.	Menghasilkan teknik pemodelan yang dapat menganalisa agen mobil pintar dan domainnya.	Teknik pemodelan BDI-MU.	Bab VI.
4.	Masalah membangunkan seni bina agen mobil pintar.	Menghasilkan rangka kerja yang dapat menukar analisa kepada reka bentuk seni bina serta membentuk agen mobil pintar yang berkomponen dan tidak monolitik.	Rangka kerja organisasi holonik dan hirarki ambients.	Bab VII.
5.	Masalah pelbagaian domain aplikasi teragih berasaskan agen mobil pintar.	Menghasilkan seni bina agen mobil pintar yang terbuka berasaskan taakulan praktikal berautonomi dan keupayaan bersosial berasaskan mesej.	Seni bina BDI-MU.	Bab VIII, Bab IX dan Bab X.
6.	Masalah mengesahkan operasi dan hubungan komponen-komponen seni bina agen mobil pintar.	Menghasilkan spesifikasi formal komponen-komponen seni bina agen mobil pintar yang dapat dibuktikan secara matematik.	Spesifikasi formal BDI-MU.	Bab XI.
7.	Masalah mengekod agen mobil pintar.	Menghasilkan alatan untuk membina agen mobil pintar yang inter operasi.	BDI-MU kit.	Bab XII.

Terdapat tujuh sumbangan kajian yang telah dihasilkan daripada kerja-kerja penyelidikan dan pembangunan yang telah dijalankan, iaitu:

- (a) Spesifikasi masalah dan kajian literatur – kajian terperinci untuk memperkukuhkan pemahaman mengenai agen mobil pintar dari perspektif teori, masalah sehingga kepada implementasi.
- (b) Metodologi BDI-MU – metodologi pembangunan agen mobil pintar yang menggabungkan kejuruteraan serempak dan metodologi ringan untuk menangani kekangan masa dan kos projek.
- (c) Teknik Pemodelan BDI-MU – teknik pemodelan agen mobil pintar yang menggabungkan UML dengan peraturan pemilihan agen dan peraturan meta agen untuk memodelkan domain, struktur intra dan struktur inter agen mobil pintar.
- (d) Rangka kerja organisasi holonik dan hirarki ambients – rangka kerja pembangunan agen mobil pintar yang menggabungkan konsep organisasi agen, holonik, hirarki agen dan ambients untuk membina agen yang berkomponen dan menyamai organisasi sebenar.
- (e) Seni bina BDI-MU – seni bina terbuka agen mobil pintar yang menggabungkan seni bina BDI, autonomi, konsep bahasa Mobile UNITY dan mesej untuk membina agen mobil pintar yang cerdas, berautonomi, berkeupayaan bergerak di antara mesin dan bersosial.
- (f) Spesifikasi formal BDI-MU – spesifikasi formal Z yang menggabungkan logik matematik dan teori set untuk membuktikan dan mengesahkan atribut, kelakuan dan interaksi komponen-komponen seni bina BDI-MU.
- (g) BDI-MU kit – alatan pembangunan agen mobil pintar yang menggunakan Java untuk membina agen yang inter operasi.

#### **13.4 Kelemahan Kajian dan Cadangan Kerja Masa Hadapan**

Walaupun sumbangan kajian telah berjaya menangani masalah kajian seperti yang telah dinyatakan dalam Bab II dan memenuhi matlamat serta objektif kajian seperti pada Bab I tetapi terdapat beberapa kelemahan yang boleh diperbaiki, iaitu:

- (a) Metodologi BDI-MU – tidak mempunyai piawaian dan dokumentasi.
- (b) Seni bina BDI-MU – tidak mempunyai protokol komunikasi, perkongsian dan guna semula pengetahuan serta keselamatan.
- (c) Pangkalan pengetahuan agen BDI-MU – tidak mempunyai pengurusan pangkalan yang berasaskan perwakilan pengetahuan.
- (d) Antara muka BDI-MU kit – tidak bergrafikal dan ramah pengguna.

Berdasarkan kepada kelemahan kajian yang telah dinyatakan di atas, maka di antara kerja-kerja masa hadapan yang boleh dilaksanakan untuk mengembangkan kajian ini adalah seperti berikut:

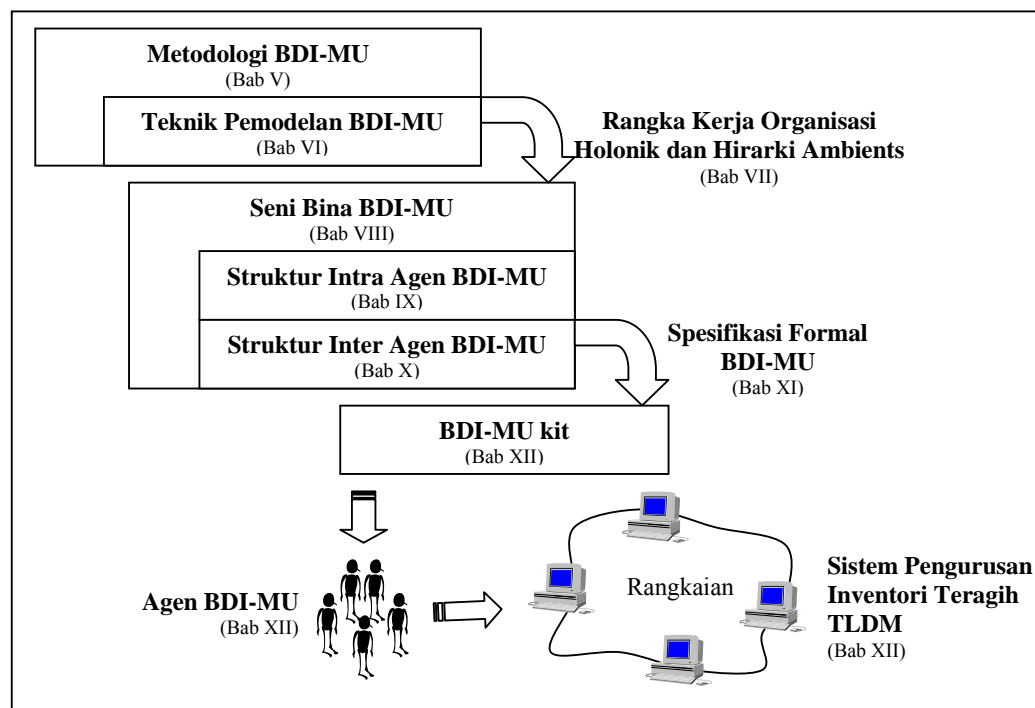
- (a) Sambungan metodologi – menambah dokumentasi yang kemas dan padat. Selain piawaian, pengurusan dan konfigurasi perisian yang lebih teratur dan sistematik.
- (b) Sambungan seni bina terbuka – menambah piawaian seperti KIF (Singh et al., 1995) untuk protokol komunikasi, KSE (Musen, 1992) untuk protokol perkongsian dan guna semula pengetahuan serta OASIS (Hayton, 1996) untuk protokol keselamatan.
- (c) Pengurusan pangkalan pengetahuan onto-objek – menambah konsep pengurusan pangkalan pengetahuan berasaskan hubungan ontologi dan objek melalui takrifan hubungan di antara kelas, slot dan penseketikaan yang lebih bermakna. Tujuannya untuk memudahkan operasi manipulasi dan meminimakan masa melombong pengetahuan.
- (d) Antara muka pengguna bergrafikal – menambah antara muka ramah pengguna untuk memudahkan pemahaman dan meningkatkan kemampuan pengguna menggunakan BDI-MU kit.

### **13.5 Penutup**

Aplikasi teragih seperti sistem pengurusan inventori teragih di TLDM adalah aplikasi yang terdiri daripada satu set komponen yang ditempatkan pada lokasi dan topologi yang bebeza tetapi saling bekerjasama dan berhubungan melalui rangkaian. Data, kawalan, kepakaran dan sumber yang teragih, dinamik dan heterogen adalah masalah pengagihan dan kompleksiti semulajadi dalam aplikasi teragih. Salah satu cara untuk menangani masalah ini adalah dengan menggunakan agen mobil pintar.

Agen mobil pintar adalah perisian yang mempunyai keupayaan untuk berpindah di antara hos-hos teragih melalui rangkaian, berkomunikasi dengan entiti lain serta berautonomi dan cerdas dalam membuat keputusan. Sifat-sifat ini membolehkan agen mobil pintar mencapai maklumat pada hos-hos yang teragih dengan pergi ke sumber tersebut dan berupaya membuat cadangan kepada penggunaanya secara reaktif, proaktif dan tanpa campur tangan manusia atau entiti lain.

Bagi menghasilkan agen mobil pintar yang dapat memenuhi ciri-ciri teragih, serempak, cerdas dan berupaya bergerak, maka tiga elemen pembangunan agen mobil pintar seperti yang telah diketengahkan pada Seksyen 3.8 diperlukan. Tujuannya adalah untuk meningkatkan keupayaan agen mobil pintar supaya ia mempunyai kemampuan minima untuk menangani masalah aplikasi teragih seperti yang telah diuraikan dalam Bab II. Oleh kerana itu, metodologi kajian seperti yang telah dibentangkan dalam Seksyen 4.2 dilaksanakan untuk memenuhi matlamat tersebut. Ia bertujuan untuk menghasilkan produk-produk sebagaimana yang telah dikemukakan dalam Seksyen 4.3. Ringkasan perjalanan kajian yang telah dilaksanakan ini adalah seperti pada Rajah 13.1.



**Rajah 13.1 : Ringkasan perjalanan kajian**

Terdapat tiga produk penyelidikan utama yang dihasilkan daripada kajian ini sepertimana yang ditunjukkan pada Rajah 13.1, iaitu:

- (a) Metodologi BDI-MU – tujuannya adalah untuk menangani masalah pembangunan agen mobil pintar dalam aplikasi teragih.
- (b) Seni bina BDI-MU – tujuannya adalah untuk menangani masalah membina komponen-komponen agen mobil pintar dalam aplikasi teragih.
- (c) BDI-MU kit – tujuannya adalah untuk menangani masalah dalam menyediakan platform operasi dan mengekod agen mobil pintar dalam aplikasi teragih.

Metodologi BDI-MU (Rajah 5.3) adalah metodologi berkategori ringan yang mengimplementasi kejuruteraan serempak dalam pelaksanaan fasa-fasanya. Ia membolehkan pembangun sistem agen mobil pintar menghasilkan artifak yang minima dan melaksanakan langkah-langkah pembangunan yang penting secara sistematik. Teknik pemodelan BDI-MU (Rajah 6.1) pula diselitkan dalam metodologi BDI-MU untuk memudahkan kerja-kerja mengecam dan memodelkan agen mobil pintar. Teknik pemodelan BDI-MU ini mengimplementasi notasi dan model UML untuk menghurai, memvisualisasi, membangun dan mendokumentasi hubungan, atribut dan kelakuan agen mobil pintar.

Seni bina BDI-MU (Rajah 8.2) adalah seni bina terbuka yang dibentuk oleh struktur intra dan inter agen mobil pintar; domain, pelayan dan konfigurasi agen mobil pintar; dan agen serta aplikasi. Kesemua struktur tersebut dibina oleh komponen-komponen dalam pakej sistem dan pakej takrifan pengguna. Struktur intra agen mobil pintar dibentuk menggunakan seni bina BDI dan autonomi (Bab IX). Tujuannya adalah untuk membolehkan agen BDI-MU membuat taakulan praktikal melalui tafsiran struktur data kepercayaan, keinginan dan niat secara berautonomi melalui saringan beberapa lapisan kepercayaan, matlamat, niat dan keupayaan. Struktur inter agen mobil pintar pula dibentuk menggunakan mesej dan konsep bahasa Mobile UNITY (Bab X). Tujuannya adalah untuk membolehkan agen BDI-MU berkomunikasi dan bergerak dalam rangkaian melalui hebahan mesej dan berasaskan konsep atur cara, interaksi dan komponen. Rangka kerja organisasi holonik (Rajah 7.4) dan hirarki ambients (Rajah 7.8) pula digunakan sebagai templat



untuk membentuk komponen-komponen dalam pakej takrifan pengguna. Spesifikasi formal operasi dan hubungan komponen-komponen yang membentuk struktur intra dan inter agen mobil pintar telah dibentangkan dalam Bab XI. Tujuannya adalah untuk mewakili operasi dan hubungan komponen-komponen tersebut dalam spesifikasi Z yang berasaskan teori set dan logik matematik supaya ia boleh disemak dan dibuktikan.

BDI-MU kit (Seksyen 12.3) adalah alatan pembangunan agen mobil pintar yang dibangunkan menggunakan Java. Java dipilih kerana kod baitnya adalah mudah dialih dan inter operasi dalam persekitaran pengkomputeran teragih dan heterogen. Java juga menyokong komponen cerdik dan berautonomi melalui konsep pelbagai bebenang dan objek; pakej pemerhati dan keadaan; dan struktur data vektor dan jadual cincangan. BDI-MU kit kemudiannya digunakan untuk membangunkan agen BDI-MU dan seterusnya diimplementasikan dalam sistem pengurusan inventori teragih di TLDM untuk menangani masalah-masalah yang telah dibentangkan dalam Seksyen 2.2.

Akhir sekali, untuk memastikan kebolegunaan kajian ini, implementasi dan keputusan komputasi telah dikemukakan pada akhir setiap Bab V sehingga Bab XII. Hasil dan keputusan yang diperolehi membuktikan bahawa metodologi BDI-MU, seni bina BDI-MU dan BDI-MU kit merupakan elemen-elemen yang berupaya menyokong pembangunan agen mobil pintar yang serempak, cerdik, teragih dan berupaya bergerak dalam persekitaran pengkomputeran heterogen. Secara tidak langsung teknologi agen mobil pintar telah berjaya dibangunkan dari perspektif kejuruteraan perisian (Bab V sehingga Bab VIII dan Bab XI) dan kepintaran buatan (Bab IX dan Bab X). Selain itu juga, kajian ini telah berjaya menyelesaikan masalah dunia sebenar melalui implementasi ke atas sistem pengurusan inventori teragih di TLDM. Kesimpulannya, agen mobil pintar yang tegap dan efisien bagi menyelesaikan masalah aplikasi teragih telah berjaya dibangunkan dalam kajian ini dan ianya boleh dikembangkan lagi melalui kajian masa depan yang telah dicadangkan bagi menyemarakkan penyelidikan dan pembangunan dalam bidang agen untuk aplikasi teragih.

### SENARAI PENERBITAN

No	Tajuk	Pengarang	Jurnal/ Persidangan	Bab Berkaitan
1	Formal Method for Distributed Concurrent Intelligent Mobile Agent System: A Survey of Belief-Desire-Intention (BDI) Model and Logic-Based Representation and Reasoning.	Muhamad Razib Othman, Azura Salleh and Safaai Deris (2000).	IRPA and UPP Progress Report. Faculty of Computer Science and Information Systems, Universiti Teknologi Malaysia. Skudai, Malaysia.	Bab III
2	A Concurrent Lightweight Agent-Oriented Methodology for Distributed Heterogeneous Multi-Agent Systems Development.	Muhamad Razib Othman, Safaai Deris, Kept. Sukhdev Singh and Lt. Kdr. Isa Yahaya (2003).	Proc. of the IEEE Int. Conf. on Robotics, Vision and Parallel Processing (ROVISP'03). January 22-24. Nibong Tebal, Malaysia: IEEE (Accepted).	Bab V
3	Analisa Masalah Domain Agen Mobil Pintar Menggunakan Bahasa Pemodelan Terpadu (UML).	Muhamad Razib Othman and Safaai Deris (2001).	Proc. of the Nat. Artificial Intelligence Seminar (AIS'01). November 1-3. Sintok, Malaysia: Penerbit UUM, 13-26.	Bab VI
4	An Organizational Based Framework for the Development of Intelligent Mobile Agent Architectures.	Muhamad Razib Othman, Safaai Deris, Kept. Sukhdev Singh and Lt. Kdr. Isa Yahaya (2002).	Proc. of the Malaysian Science and Technology Congress (MSTC'02). September 19-21. Johore Bahru, Malaysia: COSTAM, 37-44.	Bab VII

No	Tajuk	Pengarang	Jurnal/ Persidangan	Bab Berkaitan
5	An Intelligent Mobile Agent Framework for Distributed Applications.	Muhamad Razib Othman and Safaai Deris (2001).	Colloquium. Department of Software Engineering, Faculty of Computer Science and Information Systems, Universiti Teknologi Malaysia. Skudai, Malaysia.	Bab VII
6	Struktur Intra Agen Mobil Pintar Berasaskan Taakulan Praktikal Menggunakan Seni Bina Kepercayaan-Keinginan-Niat (BDI) untuk Aplikasi Teragih.	Muhamad Razib Othman, Safaai Deris, Kept. Sukhdev Singh and Lt. Kdr. Isa Yahaya (2002).	Proc. of the Nat. Conf. on Computer Graphics and Multimedia (CoGRAMM'02). October 7-9. Malacca, Malaysia: Penerbit UTM, 102-109.	Bab IX
7	Integrating Message and Mobile UNITY for Intelligent Mobile Agent Inter-Structure.	Muhamad Razib Othman, Safaai Deris, Kept. Sukhdev Singh and Lt. Kdr. Isa Yahaya (2002).	Proc. of the MMU Int. Symposium on Information and Communications Technologies (M2USIC'02). October 2-3. Petaling Jaya, Malaysia: MMU, 132-135.	Bab X
8	A Java Based Intelligent Mobile Agent for Distributed Heterogeneous Operating System.	Muhamad Razib Othman, Safaai Deris, Kept. Sukhdev Singh and Lt. Kdr. Isa Yahaya (2002).	Proc. of the Nat. Open Source Conf. and Exhibition (OpenSource'02). August 6-7. Kuala Lumpur, Malaysia: UPSI, 25-30.	Bab XII

**SENARAI DOKUMEN RUJUKAN**

- Agre, P. E. (1995). Computational Research on Interaction and Agency: *Artificial Intelligence*. 72(1-2): 1-52.
- Agre, P. E. and Horswill, I. (1997). Lifeworld Analysis: *J. of Artificial Intelligence Research*. 6(1): 111-145.
- Amiza Amir (2002). *Prototaip Sistem Pengurusan Tabung Darah Teragih Berasaskan Agen Mobil*. Universiti Teknologi Malaysia, Malaysia: Tesis B.Sc.
- Ardissono, L., Goy, A., Petrone, G. and Segnan, M. (2002). Personalization in Business-to-Customer Interaction: *Communications of the ACM*. 45(5): 52-53.
- Bancerek, G. (1990). A Model of ZF Set Theory Language: *J. of Formalized Mathematics*. 1(1): 131-145.
- Bapst, F., Brugger, R., Zramdini, A. and Ingold, R. (1996). Integrated Multi-Agent Architecture for Assisted Recognition. *Proc. of the 1st. IAPR Int. Workshop on Document Analysis Systems (DAS'96)*. October 14-16. Malvern, USA: Springer-Verlag, 129-134.
- Bates, J. (1994). The Role of Emotion in Believable Agents: *Communications of the ACM*. 37(7): 122-125.
- Bauer, B., Muller, J. P. and Odell, J. (2001). Agent UML: A Formalism for Specifying Multiagent Software Systems: *Int. J. of Software Engineering and Knowledge Engineering*. 11(3): 207-230.

- Baumann, J., Hohl, F., Rothermel, K. and Strasser, M. (1997). Mole: Concepts of a Mobile Agent System: *WWW Journal – Special Issue on Applications and Techniques of Web Agents*. 1(3): 133-137.
- Bellavista, P., Corradi, A. and Stefanelli, C. (1999). An Open Secure Mobile Agent Framework for Systems Management: *J. of Network and Systems Management*. 7(3): 323-339.
- Bellifemine, F., Poggi, A. and Rimassa, G. (2001). Developing Multi-Agent Systems with a FIPA-Compliant Agent Framework: *Software – Practice and Experience*. 31(2): 103-128.
- Bigus, J. P. and Bigus J. (1999). *Constructing Intelligent Agents with Java: A Programmer's Guide to Smarter Applications*. New York, USA: John Wiley & Sons.
- Boehm, B. W. (1988). A Spiral Model of Software Development and Enhancement: *IEEE Computer*. 21(5): 61-72.
- Bongaerts, L. (1998). *Integration of Scheduling and Control in Holonic Manufacturing Systems*. Katholieke Universiteit Leuven, Belgium: Tesis Ph.D.
- Booch, G. (1994). Coming of Age in an Object-Oriented World: *IEEE Software*. 11(6): 33-41.
- Booch, G., Rumbaugh, J. E. and Jacobson, I. (1999). The Unified Modeling Language User Guide: *J. of Database Management*. 10(4): 51-52.
- Bowen, J. P. and Hinchey, M. G. (1995). Seven More Myths of Formal Methods: *IEEE Software*. 12(4): 34-41.
- Bradshaw, J. M., Dutfield, S., Carpenter, B., Jeffers, R. and Robinson, T. (1995). KAoS: A Generic Agent Architecture for Aerospace Applications. *Proc. of the*

*4th. Int. Conf. on Information and Knowledge Management (CIKM'95).*

December 1-2. Baltimore, USA: ACM SIGART, 375-418.

Bratman, M. E. (1987). *Intentions, Plans and Practical Reasoning*. Cambridge, USA: Harvard University Press.

Bratman, M. E., Israel, D. J. and Pollack, M. E. (1988). Plans and Resource-Bounded Practical Reasoning: *Computational Intelligence*. 4(4): 349-355.

Brazier, F. M. T., Dunin-Keplicz, B. M., Jennings, N. R. and Treur, J. (1997). DESIRE: Modelling Multi-Agent Systems in a Compositional Formal Framework: *Int. J. of Cooperative Information Systems*. 1(6): 67-94.

Brewington, B. E. and Cybenko, G. (2000). How Dynamic is the Web?: *Computer Networks*. 33(1-6): 257-276.

Broy, M., Deimel, A., Henn, J., Koskimies, K., Plasil, K., Pomberger, G., Pree, W., Stal, M. and Szyperski, C. A. (1998). What Characterizes a (Software) Component?: *Software – Concepts and Tools*. 19(1): 49-56.

Bruderlin, A., Fels, S., Esser, S. and Mase, K. (1997). Hierarchical Agent Interface for Animation. *Proc. of the 15th. Int. Joint Conf. on Artificial Intelligence (IJCAI'97)*. August 23-29. Nagoya, Japan: Taylor & Francis, 27-31.

Bryson, J. (2000). Cross-Paradigm Analysis of Autonomous Agent Architecture: *J. of Experimental and Theoretical Artificial Intelligence*. 12(2): 165-190.

Burckert, H. J., Fischer, K. and Vierke, G. (2000). Holonic Transport Scheduling with TELETRUCK: *Applied Artificial Intelligence*. 14(7): 697-725.

Burmeister, B. (1996). Models and Methodology for Agent-Oriented Analysis and Design. *Proc. of the 20th. Annual German Conf. on Artificial Intelligence (KI'96)*. September 17-19. Saarbrücken, Germany: Springer-Verlag, 96-106.

- Busetta, P. and Ramamohanarao, K. (1998). An Architecture for Mobile BDI Agents. *Proc. of the 13th. ACM Symposium on Applied Computing (SAC'98)*. February 27-28. Atlanta, USA: ACM Press, 445-452.
- Cabri, G. and Leonardi, L. (2000). Mobile-Agent Coordination Models for Internet Applications: *IEEE Computer*. 33(2): 82-89.
- Cabri, G., Leonardi, L. and Zambonelli, F. (2000). MARS: A Programmable Coordination Architecture for Mobile Agent: *IEEE Internet Computing*. 4(4): 26-35.
- Cardelli, L. (1995). A Language with Distributed Scope: *Computing Systems*. 8(1): 27-59.
- Cardelli, L. and Gordon, A. (2000). Mobile Ambients: *Theoretical Computer Science*. 240(1): 177-213.
- Castelfranchi, C. (1998). Modelling Social Action for AI Agents: *Artificial Intelligence*. 103(1-2): 157-182.
- Castro, J., Kolp, M. and Mylopoulos, J. (2002). Towards Requirements-Driven Information Systems Engineering: The Tropos Project: *Information Systems*. 27(6): 365-389.
- Cavalcanti, A. and Woodcock, J. (1998). ZRC: A Refinement Calculus for Z: *Formal Aspects of Computing*. 10(3): 267-289.
- Chaib-Draa, B., Moulin, B., Mandiau, R. and Millot, P. (1992). Trends in Distributed Artificial Intelligence: *Artificial Intelligence Review*. 6(1): 35-66.
- Chan, P. K. and Stolfo, S. J. (1997). On the Accuracy of Meta-Learning for Scalable Data Mining: *J. of Intelligent Information Systems*. 8(1): 5-28.

- Chandy, K. M. and Misra, J. (1979). Distributed Simulation: A Case Study in the Design and Verification of Distributed Programs: *IEEE Transactions on Software Engineering*. 5(5): 440-452.
- Chandy, K. M. and Misra, J. (1986). How Processes Learn: *Distributed Computing*. 1(1): 40-52.
- Chauhan, D. (1997). *JAFMAS: A Java-Based Agent Framework for Multiagent Systems Development and Implementation*. University of Cincinnati, USA: Tesis Ph.D.
- Cheriton, D. R. (1988). The V Distributed System: *Communications of the ACM*. 31(3): 314-333.
- Chesnevar, C. I., Dix, J., Stolzenburg, F. and Simari, G. R. (2002). Relating Defeasible and Normal Logic Programming Through Transformation Properties: *Theoretical Computer Science*. 290(1): 499-529.
- Cohen, P. R. and Levesque, H. J. (1990). Intention is Choice with Commitment: *Artificial Intelligence*. 42(3): 213-261.
- Covaci, S., Zhang, Tianning and Busse, I. (1997). Java-Based Intelligent Mobile Agents for Open Systems Management. *Proc. of the 9th. Int. Conf. on Tools with Artificial Intelligence (TAI'97)*. November 3-8. Newport Beach, USA: IEEE Computer Society Press, 492-501
- Craig, I. D. (1991). *Formal Specification of Advanced AI Architectures*. Chichester, UK: Ellis Horwood Limited.
- d'Inverno, M. and Luck, M. (1998). Engineering AgentSpeak(L): A Formal Computational Model: *J. of Logic and Computation*. 8(3): 233-260.
- Dale, J. (1997). *A Mobile Architecture for Distributed Information Management*. Southampton University, UK: Tesis Ph.D.



- Dardenne, A., Lamsweerde, A. V. and Fickas, S. (1993). Goal-Directed Requirements Acquisition: *Science of Computer Programming*. 20(1): 3-50.
- Dasgupta, P., Narasimhan, N., Moser, L. and Melliari-Smith, P. M. (1999). MAgNET: Mobile Agents for Networked Electronic Trading: *IEEE Transactions on Knowledge and Data Engineering*. 11(4): 509-525.
- De Nicola, R., Ferrari, G. and Pugliese, R. (1998). KLAIM: A Kernel Language for Agents Interaction and Mobility: *IEEE Transactions on Software Engineering*. 24(5): 315-330.
- Decker, K. and Sycara, K. P. (1997). Intelligent Adaptive Information Agents: *J. of Intelligent Information Systems*. 9(3): 239-260.
- Deloach, S. A., Wood, M. F. and Sparkman, C. H. (2001). Multiagent Systems Engineering: *Int. J. of Software Engineering and Knowledge Engineering*. 11(3): 231-258.
- Do, T. T., Faulkner, S. and Kolp, M. (2002). The Structure-in-5 as an Agent Architectural Pattern. *Proc. of the 4th. Int. Bi-Conf. Workshop on Agent Oriented Information Systems (AOIS'02)*. May 27-28. Toronto, Canada: iCue Publishing, 14-28.
- Dolin, R., Alschuler, L., Boyer, S. and Beebe, C. (2000). An Update on HL7's XML-Based Document Representation Standards. *Proc. of the 20th. American Medical Informatics Association Symposium (AMIA'00)*. November 4-8. Los Angeles, USA: IEEE Computer Society, 190-194.
- Dorigo, M., Maniezzo, V. and Colorni, A. (1996). The Ant System: Optimization by a Colony of Cooperating Agents: *IEEE Transactions on Systems, Man and Cybernetics – Part B*. 26(2): 29-41.
- Egges, A., Nijholt, A. and op den Akker, R. (2001) Dialogs with BDP Agents in Virtual Environments. *Proc. of the 2nd. IJCAI Workshop on Knowledge and*

*Reasoning in Practical Dialogue Systems (IJCAI-WS'01)*. August 5-6. Seattle, USA: AAAI Press, 148-173.

- Egyed, A. and Kruchten, P. (1999). Rose/Architecture: A Tool to Visualize Architecture. *Proc. of the 32nd. Annual Hawaii Int. Conf. on System Sciences (HICSS'32)*. January 5-8. Maui, USA: IEEE Computer Society, 2-16.
- Eng, L. Y. D., Hih, T. K. S., Huang, Tehsheng, Liao, Yichun, Wang, Yinghong and Hsu, Huihuang (2002). Distributed Mobile Agent Framework for Maintaining Persistent Distance Education: *J. of Information Science and Engineering*. 18(1): 489-506.
- Etzioni, O. (1997). Moving Up the Information Food Chain: Deploying Softbots on the World Wide Web: *AI Magazine*. 18(2): 11-18.
- Ezatul Suzarafah Taib, Masura Rahmat, Siti Asma Mohamad Nor and Tengku Maria Idayu Tengku Abidin (2002). An Inventory Agent in Distributed Server Management System. *Proc. of the 1st. Class Conf. on Agent Technologies and Applications (ATA'02)*. August 17-18. Skudai, Malaysia: Penerbit AILab, 13-19.
- Falcone, R. and Castelfranchi, C. (2000). Grounding Autonomy Adjustment on Delegation and Trust Theory: *J. of Experimental and Theoretical Artificial Intelligence*. 12(2): 149-151.
- Faulstich, L. and Spiliopoulou, M. (2000). Building Hyperview Wrappers for Publisher Web Sites: *Int. J. on Digital Libraries*. 3(1): 3-18.
- Finin, T. W., Fritzon, R., McKay, D. and McEntire, R. (1994). KQML as an Agent Communication Language. *Proc. of the 3rd. Int. Conf. on Information and Knowledge Management (CIKM'94)*. December 2-3. Gaithersburg, USA: ACM Press, 456-463.
- Fischer, K. (1999). Agent-Based Design of Holonic Manufacturing Systems: *J. of Robotics and Autonomous Systems*. 27(1-2): 3-13.

- Fisher, M. (1996). Temporal Semantics for Concurrent METATEM: *J. of Symbolic Computation*. 22(5-6): 627-648.
- Fisher, M., Muller, H. J., Schroeder, M., Staniford, G. and Wagner, G. (1997). Methodological Foundations for Agent-Based Systems: *Knowledge Engineering Review*. 12(3): 323-329.
- Fox, M. S., Barbuceanu, M. and Teigen, R. (2000). Agent-Oriented Supply-Chain Management: *Int. J. of Flexible Manufacturing Systems*. 12(2-3): 165-188.
- Fricke, S., Bsufka, K., Keiser, J., Schmidt, T., Sessler, R. and Albayrak, S. (2001). Agent-Based Telematic Services and Telecom Applications: *Communications of the ACM*. 44(4): 43-48.
- Fujita, K. and Akagi, S. (1999). Agent-Based Distributed Design System Architecture for Basic Ship Design: *J. of Concurrent Engineering: Research and Applications*. 7(2): 83-94.
- Fuxman, A. D. (2001). *Formal Analysis of Early Requirements Specifications*. University of Toronto, Canada: Tesis Ph.D.
- Garey, M. R., Johnson, D. S. and Stockmeyer, L. (1976). Some Simplified NP-Complete Graph Problems: *Theoretical Computer Science*. 1(3): 237-267.
- Garlan, D. (1995). Research Directions on Software Architecture: *ACM Computing Surveys*. 27(2): 257-261.
- Gavalas, D., Greenwood, D., Ghanbariand, M. and O'Mahony, M. (2000). Advanced Network Monitoring Applications Based on Mobile/Intelligent Agent Technology: *Computer Communications*. 23(8): 720-730.
- Genesereth, M. and Ketchpel, S. (1994). Software Agents: *Communications of the ACM*. 37(7): 48-53.

- Georgeff, M. P. and Lansky, A. L. (1986). Procedural Knowledge: *IEEE Special Issue on Knowledge Representation*. 74(10): 1383-1398.
- Georgeff, M. P., Lansky, A. L. and Bessiere, P. (1985). A Procedural Logic. *Proc. of the 9th. Int. Joint Conf. on Artificial Intelligence (IJCAI'85)*. August 18-23. Los Angeles, USA: Morgan Kaufmann, 516-523.
- Gerber, A. and Klusch, M. (2002). CASA: Agent-Based Integrated Services Network for Timber Production and Sales: *IEEE Intelligent Systems*. 17(2): 33-39.
- Gerber, C., Siekmann, J. and Vierke, G. (1999). Flexible Autonomy in Holonic Agent Systems. *Proc. of the 16th. Nat. Conf. on Artificial Intelligence (AAAI'99)*. July 18-19. Orlando, USA: AAAI Press, 697-725.
- Gilbert, N. and Terna, P. (2000). How to Build and Use Agent-Based Models in Social Science: *Mind and Society*. 1(1): 57-72.
- Glaser, N. (1996). *Contribution to Knowledge Modelling in a Multi-Agent Framework (the CoMoMAS Approach)*. L'Université Henri Poincaré, France: Tesis Ph.D.
- Goodwin, R. (1995). Formalizing Properties of Agents: *J. of Logic and Computation*. 5(6): 763-781.
- Graham, J. R. and Decker, K. S. (1999). Towards a Distributed, Environment-Centered Agent Framework. *Proc. of the 6th. Int. Workshop on Agent Theories, Architectures and Languages (ATAL'99)*. July 15-17. Orlando, USA: Springer-Verlag, 290-304.
- Gruber, T. R. (1993). A Translation Approach to Portable Ontologies: *Knowledge Acquisition*. 5(2): 199-220.
- Hashimah Samion, Muhamad Razib Othman and Mohd. Zaki A. Hadi (2002). A Mobile Agent for Integrated Distributed Inventory Management. *Proc. of the 1st.*

*Nat. Conf. on Computer Graphics and Multimedia (CoGRAMM'02)*. October 7-9. Malacca, Malaysia: Penerbit UTM, 68-74.

Hattori, M., Kase, N., Ohsuga, A. and Honiden, S. (1999). Agent-Based Drivers' Information Assistance System: *New Generation Computing*. 17(4): 359-367.

Hayes-Roth, B. (1995). An Architecture for Adaptive Intelligent System: *Artificial Intelligence: Special Issue on Agents and Interactivity*. 72(1): 329-365.

Hayton, R. (1996). *OASIS: An Open Architecture for Secure Interworking Services*. University of Cambridge, UK: Tesis Ph.D.

Hewitt, C. (1977). Viewing Control Structure as Patterns of Passing Messages: *Artificial Intelligence*. 8(3): 323-364.

Huber, M. J. and Hadley, T. (1997). Multiple Roles, Multiple Teams, Dynamic Environment: Autonomous Netrek Agents. *Proc. of the 1st. Int. Conf. on Autonomous Agents (Agents '97)*. February 5-8. Marina del Rey, USA: ACM Press, 332-339.

Huhns, M. N. and Singh, N. P. (1997). Agents on the Web: Conversational Agents: *IEEE Internet Computing*. 1(2): 73-75.

Iglesias, C. A., Garijo, M. and Gonzalez, J. C. (1998). A Survey of Agent-Oriented Methodologies. *Proc. of the 5th. Int. Workshop on Agent Theories, Architectures and Languages (ATAL '98)*. December 19-20. Paris, France: Springer-Verlag, 317-330.

Iglesias, C. A., Garijo, M., Gonzalez, J. C. and Velasco, J. R. (1996). A Methodological Proposal for Multiagent Systems Development Extending CommonKADS. *Proc. of the 10th. Int. Knowledge Acquisition for Knowledge-Based Systems Workshop (KAW'96)*. April 18-23. Banff, Canada: KSI Publication, 17-25.

- Iglesias, C., Garijo, M., Gonzales, J. C. and Velasco, J. R. (1997). Analysis and Design of Multiagent Systems using MAS-CommonKADS. *Proc. of the 4th. Int. Workshop on Agent Theories, Architectures and Languages (ATAL'97)*. July 24-26. Rhode Island, USA: Springer-Verlag, 313-327.
- Izahmunirah Mohammed, Marina Helianda Harun, Rozilawati Ismail, Noremi Che Azmi and Norazrina Ahmad Shatop (2002). Network Application Manager. *Proc. of the 1st. Class Conf. on Agent Technologies and Applications (ATA'02)*. August 17-18. Skudai, Malaysia: Penerbit AILab, 20-28.
- Jacobson, I. (1993). Is Object Technology Software's Industrial Platform?: *IEEE Software*. 10(1): 24-30.
- Jacobson, I., Bylund, S., Jonsson, P. and Ehneboom, S. (1995). Using Contracts and Use Cases to Build Pluggable Architectures: *J. of Object-Oriented Programming*. 8(2): 18-24.
- Jacobson, G. and Weissman, M. (1993). Alarm Correlation: *IEEE Network*. 7(6): 52-59.
- Jennings, N. R. (1993). Specification and Implementation of a Belief Desire Joint-Intention Architecture for Collaborative Problem Solving: *J. of Intelligent and Cooperative Information Systems*. 2(3): 289-318.
- Jennings, N. R. (2001). An Agent-Based Approach for Building Complex Software Systems: *Communications of the ACM*. 44(4): 35-41.
- Jennings, N. R. and Wooldridge, M. J. (1995). Applying Agent Technology: *Applied Artificial Intelligence*. 9(4): 357-369.
- Jennings, N. R. and Wooldridge, M. J. (1996). Software Agents: *IEEE Review*. 42(1): 17-20.

- Jennings, N. R., Corera, J. M., Laresgoiti, I., Mamdani, E. H., Perriollat, F., Skarek, P. and Varga, L. Z. (1996). Using ARCHON to Develop Real-World DAI Applications for Electricity Transportation Management and Particle Accelerator Control: *IEEE Expert*. 11(6): 64-70.
- Jennings, N. R., Sycara, K. and Wooldridge, M. J. (1998). A Roadmap of Agent Research and Development: *J. of Autonomous Agents and Multi-Agent Systems*. 1(1): 7-36.
- Jeon, H., Petrie, C. and Cutkosky, M. R. (2000). JATLite: A Java Agent Infrastructure with Message Routing: *IEEE Internet Computing*. 4(2): 87-96.
- Johnson, R. and Foote, B. (1988). Designing Reusable Classes: *J. of Object-Oriented Programming*. 1(2): 22-35.
- Karjoth, G., Lange, D. B. and Oshima, M. (1997). A Security Model for Aglets: *IEEE Internet Computing*. 1(4): 68-77.
- Kawamura, T., Hasegawa, T., Ohsuga, A. and Honiden, S. (2000). Bee-gent: Bonding and Encapsulation Enhancement Agent Framework for Development of Distributed Systems: *J. of Systems and Computers in Japan*. 31(13): 42-56.
- Kazman, R. and Carriere, J. (1999). Playing Detective: Reconstructing Software Architecture from Available Evidence: *J. of Automated Software Engineering*. 6(2): 107-138.
- Kazman, R., Abowd, G., Bass, L. and Clements, P. (1999). Scenario-Based Analysis of Software Architecture: *IEEE Software*. 13(6): 47-55.
- Kim, Myungchul, Chanson, S. T. and Vuong, S. T. (1999). Concurrency Model for Distributed Systems: *J. of Parallel and Distributed Computing*. 59(3): 445-464.
- Kinny, D. and Georgeff, M. P. (1997). Modelling and Design of Multi-Agent Systems. *Proc. of the 3rd. Int. Workshop on Agent Theories, Architectures and*

*Languages (ATAL '96)*. August 12-13. Heidelberg, Germany: Springer-Verlag, 1-20.

Kinny, D., Georgeff, M. P. and Rao, A. S. (1996). A Methodology and Modelling Technique for Systems of BDI Agents. *Proc. of the 7th. European Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW'96)*. January 22-25. Eindhoven, The Netherlands: Springer-Verlag, 56-71.

Koblick, R. (1999). Concordia: *Communications of the ACM*. 42(3): 96-97.

Koestler, A. (1989). *A Ghost in the Machine*. New York, USA: Arkana Books.

Kotz, D., Cybenko, G., Gray, R. S., Jiang, Guofei, Peterson, R. A., Hofmann, M. O., Chacon, D. A., Whitebread, K. R. and Hendler, J. A. (2002). Performance Analysis of Mobile Agents for Filtering Data Streams on Wireless Networks: *Mobile Networks and Applications*. 7(2): 163-174.

Kotz, D., Gray, R. S., Nog, S., Rus, D., Chawla, S. and Cybenko, G. (1997). AGENT TCL: Targeting the Needs of Mobile Computers: *IEEE Internet Computing*. 1(4): 58-67.

Kruchten, P. (1995). Architectural Blueprints – The “4+1” View Model of Software Architecture: *IEEE Software*. 12(6): 42-50.

Lakemeyer, G. (1994). Limited Reasoning in First-Order Knowledge Bases: *Artificial Intelligence*. 71(2): 213-255.

Lee, Jaeho. (1996). *Structured Circuit Semantics*. University of Michigan, USA: Thesis Ph.D.

Lejter, M. and Dean, T. (1996). A Framework for the Development of Multi-Agent Architectures: *IEEE Expert*. 11(6): 47-59.



- Lohse, G. L. and Spiller, P. (1998). Electronic Shopping: *Communications of the ACM*. 41(7): 81-87.
- Lozano-Perez, T. (1985). Compliance in Robot Manipulation: *Artificial Intelligence*. 25(1): 5-12.
- Luck M. and d'Inverno, M. (1995). Structuring Z Specification to Provide a Formal Framework for Autonomous Agent Systems. *Proc. of the 9th. Int. Conf. of Z User Meeting (ZUM'95)*. September 7-8. Limerick, Ireland: Springer-Verlag, 48-62.
- Luck, M. and d'Inverno, M. (2001). A Conceptual Framework for Agent Definition and Development: *The Computer Journal*. 44(1): 1-20.
- Maes, P. (1991). The Agent Network Architecture: *SIGART Bulletin*. 2(4): 115-120.
- Martin, D. L., Cheyer, A. J. and Moran, D. B. (1999). The Open Agent Architecture: A Framework for Building Distributed Software Systems: *Applied Artificial Intelligence*. 13(1): 92-128.
- Mascolo, C. (2001). *Formalization, Analysis and Prototyping of Mobile Code Systems*. University of Bologna, Italy: Tesis Ph.D.
- Mayhew, D. J. (1998). The Usability Engineering Lifecycle. *Proc. of the 9th. Conf. on Human Factors in Computing Systems (CHI'98)*. April 18-23. Los Angeles, USA: ACM Press, 127-128.
- McCann, P. J. and Roman, G. C. (1998). Compositional Programming Abstractions for Mobile Computing: *IEEE Transactions on Software Engineering*. 24(2): 97-110.
- McFarlane, D., Marett, B., Elsley, G., Jarvis, D. and Wilbers, P. (1995). Application of Holonic Methodologies to Problem Diagnosis in a Rod Mill Cooling Control

- System. *Proc. of the 12th. IEEE Int. Conf. on Systems, Man and Cybernetics (SMC'95)*. April 9-13. Vancouver, Canada: IEEE SMC Society, 28-35.
- McGrath, S., Chacon, D. and Whitebread, K. (2000). Intelligent Mobile Agents in Military Command and Control. *Proc. of the 4th. Int. Conf. on Autonomous Agents (Agents'00)*. June 3-7. Barcelona, Spain: ACM Press, 30-43.
- Michel, G. (1990). Micro-Kernel Architecture Key to Modern Operating Systems Design: *Unix Review*. 8(11): 1-10.
- Millington, D. and Stapleton, J. (1995). Developing a RAD Standard: *IEEE Software*. 12(5): 54-55.
- Milojicic, D. S., Agha, G., Bernadat, P., Chauhan, D., Guday, S., Nadeem Jamali, Lambright, D. and Travostino, F. (2002). Case Studies in Security and Resource Management for Mobile Object Systems: *J. of Autonomous Agents and Multi-Agent Systems*. 5(1): 45-79.
- Moizumi, K. (1998). *The Mobile Agent Planning Problem*. Dartmouth College, USA: Tesis Ph.D.
- Monroe, R. T., Kompanek, A., Melton, R. E. and Garlan, D. (1993). Architectural Styles, Design Patterns and Objects: *IEEE Software*. 14(1): 43-52.
- Moore, R. C. (1996). Logic and Representation: *Computing Reviews*. 37(5): 246-247.
- Muller, J. P. (1995). (Multi)-Agent Systems Engineering. *Proc. of the 1st. Knowledge Engineering Forum: Distributed Expertised (KEF'95)*. August 23-24. Karlsruhe, Germany: Springer-Verlag, 19-35.
- Muller, J. P. (1996). A Cooperation Model for Autonomous Agents. *Proc. of the 3rd. Int. Workshop on Agent Theories, Architectures and Languages (ATAL'96)*. August 12-13. Heidelberg, Germany: Springer-Verlag, 245-260.

- Muller, J. P. (1997). Towards Agent Systems Engineering: *Int. J. on Data and Knowledge Engineering*. 23(3): 217-245.
- Muller, J. P., Pischel, M. and Thiel, M. (1994). Modelling Reactive Behaviour in Vertically Layered Agent Architectures. *Proc. of the 1st. Int. Workshop on Agent Theories, Architectures and Languages (ATAL '94)*. August 8-9. Amsterdam, The Netherlands: Springer-Verlag, 261-276.
- Murphy, A. L., Roman, G. C. and Varghese, G. (2002). Tracking Mobile Units for Dependable Message Delivery: *IEEE Transactions on Software Engineering*. 28(5): 433-447.
- Musen, M. A. (1992). Dimensions of Knowledge Sharing and Reuse: *J. of Computers and Biomedical Research*. 25(1): 435-467.
- Mylopoulos, J., Borgida, A. and Yu, E. S. K. (1997). Representing Software Engineering Knowledge: *Automated Software Engineering*. 4(3): 291-317.
- Mylopoulos, J., Chung, L. and Yu, E. S. K. (1999). From Object-Oriented to Goal-Oriented Requirements Analysis: *Communications of the ACM* 42(1): 31-37.
- Nord, R. L., Paulish, D. J., Soni, D. and Hofmeister, C. (2001). Effective Software Architecture Design: From Global Analysis to UML Descriptions. *Proc. of the 23rd. Int. Conf. on Software Engineering (ICSE '01)*. May 12-19. Toronto, Canada: IEEE Computer Society, 741-742.
- Norling, E. and Ritter, F. E. (2001). Embodying the JACK Agent Architecture. *Proc. of the 14th. Australian Joint Conference on Artificial Intelligence (AI'01)*. December 10-14. Adelaide, Australia: Springer-Verlag, 368-377.
- Noy, N. F., Sintek, M., Decker, S., Crubezy, M., Ferguson, R. W. and Musen, M. A. (2001). Creating Semantic Web Contents with Protege-2000: *IEEE Intelligent Systems*. 16(2): 60-71.

- Nwana, H. S. (1996). Software Agents: An Overview: *Knowledge Engineering Review*. 11(3): 205-244.
- Odell, J. and Parunak H. V. D. (2000). Agents: Technology and Usage (Part 1): *Distributed Computing Architecture*. 3(4): 1-29.
- Odell, J., Parunak, H. V. D. and Bauer, B. (2001). Representing Agent Interaction Protocols in UML. *Proc. of the 2nd. Int. Workshop on Agent-Oriented Software Engineering (AOSE'01)*. May 29-31. Montreal, Canada: Springer-Verlag, 121-140.
- Ohsuga, A., Nagai, Y., Irie, Y., Hattori M. and Honiden, S. (1997). Plangent: An Approach to Making Mobile Agents Intelligent: *IEEE Internet Computing*. 1(4): 50-57.
- Palen, J. (1997). The Need for Surveillance in Intelligent Transportation Systems: *Intellimotion*. 6(1): 1-16.
- Parunak, H. V. D. (1997). Go to the Ant: Engineering Principles from Natural Agent Systems: *Annals of Operations Research*. 75(1): 69-101.
- Parunak, H. V. D., Sauter, J., Fleischer, M. and Ward, A. (1999). The RAPPID Project: Symbiosis between Industrial Requirements and MAS Research: *J. of Autonomous Agents and Multi-Agent Systems*. 2(2): 111-140.
- Paulk, M. C. (2001). Extreme Programming from a CMM Perspective: *IEEE Software*. 18(6): 19-26.
- Perry, D. E. and Wolf, A. L. (1992). Foundations for the Study of Software Architecture: *ACM SIGSOFT Software Engineering Notes*. 17(4): 40-52.
- Plasil, F. and Stal, M. (1998). An Architectural View of Distributed Objects and Components in CORBA, Java RMI and COM/DCOM: *Software – Concepts and Tools*. 19(1): 14-28.

- Pohl, K. and Jacobs, S. (1994). Concurrent Engineering: Enabling Traceability and Mutual Understanding: *J. of Concurrent Engineering: Research and Applications*. 2(4): 279-290.
- Prietula, M. J. and Carley, K. M. (1994). Computational Organization Theory: Autonomous Agents and Emergent Behavior: *J. of Organizational Computing*. 41(1): 41-83.
- Prietula, M. J. and Carley, K. M. (1999). Exploring the Effects of Agent Trust and Benevolence in a Simulated Organizational Task: *Applied Artificial Intelligence*. 13(3): 321-338.
- Rao, A. S. and Georgeff, M. P. (1995). BDI Agents: From Theory to Practice. *Proc. of the 1st. Int. Conf. on Multi-Agent Systems (ICMAS'95)*. June 12-14. San Francisco, USA: AAAI Press, 312-319.
- Rao, A. S. and Georgeff, M. P. (1998). Decision Procedures for BDI Logics: *J. of Logic and Computation*. 8(3): 293-342.
- Rising, L. and Janoff, N. S. (2000). The SCRUM Software Development Process for Small Teams: *IEEE Software*. 17(4): 26-32.
- Roman, G. C., McCann, P. J. and Plun, J. Y. (1997). Mobile UNITY: Reasoning and Specification in Mobile Computing: *ACM Transactions on Software Engineering and Methodology*. 6(3): 250-282.
- Royce, W. W. (1970). Managing the Development of Large Software Systems: Concepts and Techniques: *IEEE WESTCON*. 14(1): 1-9.
- Rumbaugh, J. E. (1995). OMT: The Development Process: *J. of Object-Oriented Programming*. 8(2): 8-16.
- Russell, S. (1997). Rationality and Intelligence: *Artificial Intelligence*. 94(1-2): 57-77.

- Saaltink, M. (1997). The Z/EVES System. *Proc. of the 10th. Int. Conf. of Z User Meeting (ZUM'97)*. April 3-4. Reading, UK: Springer-Verlag, 52-71.
- Satoh, I. (2001). MobiDoc: A Mobile Agent-based Framework for Compound Documents: *Int. J. of Computing and Informatics*. 25(4): 493-500.
- Sen, S. (1997). Developing an Automated Distributed Meeting Scheduler: *IEEE Expert*. 12(4): 41-45.
- Shaw, M. (1995). Comparing Architectural Design Styles: *IEEE Software*. 12(6): 27-41.
- Shaw, M., DeLine, R., Klein, D. V., Ross, T. L., Young, D. M. and Zelesnik, G. (1995). Abstractions for Software Architecture and Tools to Support Them: *IEEE Transactions on Software Engineering*. 21(4): 314-335.
- Shoham, Y. (1993). Agent-Oriented Programming: *Artificial Intelligence*. 60(1): 51-92.
- Siegel, J. (1998). OMG Overview: CORBA and the OMA in Enterprise Computing: *Communications of the ACM*. 41(10): 37-43.
- Sierra, C., Wooldridge, M. J. and Sadeh, N. M. (2000). Agents on the Web: Agent Research and Development in Europe: *IEEE Internet Computing*. 4(5): 81-83.
- Singh, N. P., Genesereth, M. R. and Syed Mustafa Ali (1995). A Distributed and Anonymous Knowledge Sharing Approach to Software Interoperation: *Int. J. of Cooperative Information Systems*. 4(4): 339-367.
- Spivey, J. M. (1996). Richer Types for Z: *Formal Aspects of Computing*. 8(5): 565-584.
- Straber, M. and Rothermel, K. (1998). Reliability Concepts for Mobile Agents: *Int. J. of Cooperative Information Systems*. 7(4): 355-382.

- Sycara, K. and Dajun, Zeng (1996). Coordination of Multiple Intelligent Software Agents: *Int. J. of Cooperative Information Systems*. 5(2-3): 181-211.
- Tarau, P. and Dahl, V. (2001). High-Level Networking with Mobile Code and First Order AND-Continuations: *Theory and Practice of Logic Programming*. 1(3): 359-380.
- Tardo, J. and Valente, L. (1996). Mobile Agent Security and Telescript. *Proc. of the 41st. Int. Conf. of the IEEE Computer Society (CompCon '96)*. February 25-28. Santa Clara, USA: IEEE-CS Press, 58-63.
- Terry, D., Theimer, M., Petersen, K., Demers, A., Spreitzer, M. and Hauser, C. (1995). Managing Update Conflicts in Bayou, a Weakly Connected Replicated Storage System: *Operating Systems Review*. 29(5): 172-183.
- Thomas, S. R. (1993). *PLACA: An Agent-Oriented Programming Language*. Stanford University, USA: Thesis Ph.D.
- Turgut, A. E., Durna, M., Erkmen, I., Erkmen, A. M. and Erden, A. (1999). Design of a Holonic Self-Reconfigurable Robotic System. *Proc. of the 1st. Int. Conf. on Robotics and Application (RA '99)*. September 3-5. California, USA: ACTA Press, 345-354.
- Vere, S. and Bickmore, T. (1990). A Basic Agent: *Int. J. of Computational Intelligence*. 6(1): 41-60.
- Voelker, G. M. and Bershad, B. N. (1998). Mobisaic: An Information System for a Mobile Wireless Computing Environment. *Proc. of the 4th. Annual ACM/IEEE Int. Conf. on Mobile Computing and Networking (MobiCom '98)*. August 15-19. Dallas, USA: ACM Press, 185-190.
- Wilkins, D. E. and Myers, K. L. (1995). A Common Knowledge Representation for Plan Generation and Reactive Execution: *J. of Logic and Computation*. 5(6): 731-761.

- Wong, D., Paciorek, N. and Moore, D. (1999). Java-Based Mobile Agents: *Communications of the ACM*. 42(3): 92-102.
- Wooldridge M. J., Jennings N. R. and Kinny D. (2000). The Gaia Methodology for Agent-Oriented Analysis and Design: *J. of Autonomous Agents and Multi-Agent Systems*. 3(3): 285-312.
- Wooldridge, M. J. (1997). Agent-Based Software Engineering: *IEEE Proceedings on Software Engineering*. 144(1): 26-37.
- Wooldridge, M. J. (1998). Agent-Based Computing: *Interoperable Communication Networks*. 1(1): 71-97.
- Wooldridge, M. J. and Jennings, N. R. (1994). Intelligent Agents: Theory and Practice: *Knowledge Engineering Review*. 10(2): 115-152.
- Yim, Hongsoon, Cho, Kyeheun, Kim, Jongwoo and Park, Sungjoo (2000). Architecture-Centric Object-Oriented Design Method for Multi-Agent Systems. *Proc. of the 4th. Int. Conf. on Multi-Agent Systems (ICMAS'00)*. July 8-9. Boston, USA: IEEE Computer Society Press, 469-470.
- Yu, E. S. K. (1995). *Modelling Strategic Relationships for Process Reengineering*. University of Toronto, Canada: Tesis Ph.D.
- Yu, E. S. K. and Mylopoulos, J. (1996). Using Goals, Rules and Methods to Support Reasoning in Business Process Reengineering: *Int. J. of Intelligent Systems in Accounting, Finance and Management*. 5(1): 1-13.
- Zunino, A. and Amandi, A. (2000). Brainstorm/J: Framework for Intelligent Agents. *Proc. of the 2nd. Argentinian Symposium on Artificial Intelligence (ASAI'00)*. September 9-13. Buenos Aires, Argentina: JAIIO, 43-58.



## Notasi struktur intra agen BDI-MU

$::=$	Is defined as
$a: A$	Declarations
$( \dots )$	Group terms
$[ \dots ]$	Optional terms
$\{ \dots \}$	Repetitive terms
$*$	Zero or more repetitions of a term
$+$	One or more repetitions of a term
$\bullet$	Concatenation
$ $	Logical disjunction, <i>or</i>
$\vee$	Logical disjunction, <i>or</i>
$x \in A$	Set membership
$A \times B \times \dots$	Cartesian product
$A \rightarrow B$	Total functions

## Bentuk Backus-Naur struktur intra agen BDI-MU

*ParseAgentDefinition* ::= *agent\_components*  
*agent\_components* ::= (<GOAL\_DECL> (*goal*)<sup>\*</sup> | <FACT\_DECL> (*fact*)<sup>\*</sup> |  
 <PLAN\_START> (*plan*)<sup>\*</sup> | <OBSERVER\_DECL> (  
 “{” *plan\_body\_elements* “}”)<sup>+</sup>)<sup>+</sup>  
*goal* ::= *goal\_action* “;”  
*fact* ::= <FACT> <IDENTIFIER> *explist* “;”  
*plan* ::= “{” *plan\_components* “}”  
*plan\_components* ::= *plan\_component* (*plan\_component*)<sup>\*</sup>  
*plan\_component* ::= *plan\_name* | *plan\_doc* | *plan\_goalspec* |  
*plan\_concludespec* | *plan\_context* | *plan\_precondition* |  
*plan\_utility* | *plan\_effects* | *plan\_failure* |  
*plan\_attributes* | *plan\_body*  
*plan\_name* ::= <PLAN\_NAME> (<STRING>)?  
*plan\_doc* ::= <PLAN\_DOC> (<STRING>)?  
*plan\_attributes* ::= <PLAN\_ATTRIBUTES> (<STRING>)?  
*plan\_goalspec* ::= <PLAN\_GOALSPEC> (*goal\_action* “;”)?  
*plan\_concludespec* ::= <PLAN\_CONCLUDESPEC> (*relation* “;”)?  
*plan\_context* ::= <PLAN\_CONTEXT> (*condition\_list*)?  
*plan\_precondition* ::= <PLAN\_PRECONDITION> (*condition\_list*)?  
*plan\_utility* ::= <PLAN\_UTILITY> (*expression* “;”)?  
*plan\_effects* ::= <PLAN\_EFFECTS> (*plan\_body\_elements*)?  
*plan\_failure* ::= <PLAN\_FAILURE> (*plan\_body\_elements*)?  
*plan\_body* ::= <PLAN\_BODY> (*plan\_body\_elements*)?  
*plan\_branch* ::= “{” *plan\_body\_elements* “}”  
*plan\_body\_elements* ::= *plan\_body\_element* (*plan\_body\_element*)<sup>\*</sup>  
*plan\_body\_element* ::= *action* “;” | <PLAN\_BODY\_AND> *plan\_branch* (  
*plan\_branch*)<sup>\*</sup> “;” | <PLAN\_BODY\_OR> *plan\_branch*  
(*plan\_branch*)<sup>\*</sup> “;” | <PLAN\_BODY\_PARALLEL>  
*plan\_branch* (*plan\_branch*)<sup>\*</sup> “;” |

```

    <PLAN_BODY_DO_ALL> plan_branch (
    plan_branch ) * “;” | <PLAN_BODY_DO_ANY>
    plan_branch ( plan_branch ) * “;” | wait_construct |
    <PLAN_BODY_WHEN> “:” action plan_branch “;” |
    <PLAN_BODY_WHILE> “:” action plan_branch “;” |
    <PLAN_BODY_DO> plan_branch
    <PLAN_BODY_WHILE> “:” action “;” |
    <PLAN_BODY_ATOMIC> plan_branch “;” | <IF>
    action plan_branch ( <ELSEIF> action plan_branch ) *
    ( <ELSE> plan_branch )? “;”
wait_construct ::= <PLAN_BODY_WAIT> “:” action “;” |
    <PLAN_BODY_WAIT> relation “;”
condition_list ::= condition ( condition ) *
    condition ::= expression “;” | <FACT> <IDENTIFIER> explist “;” |
    <RETRIEVE> <IDENTIFIER> explist “;”
    action ::= exec_action | misc_action | goal_action
exec_action ::= <EXECUTE> <IDENTIFIER> “:” <IDENTIFIER>
    explist | <EXECUTE> <IDENTIFIER> explist |
    <LOAD> explist | <PARSE> explist | <RETRIEVE>
    <IDENTIFIER> explist | <TEST> expression |
    <ASSERT> <IDENTIFIER> explist | <RETRACT>
    <IDENTIFIER> explist | <UPDATE> “(”
    <IDENTIFIER> explist “)” “(” <IDENTIFIER> explist
    “)”
misc_action ::= <ASSIGN> expression expression | <FACT>
    <IDENTIFIER> explist | <FAIL> | <POST>
    goal_action | <UNPOST> goal_action
goal_action ::= goal_action_basic ( <KEYWORD_UTILITY>
    expression )? ( <KEYWORD_BY> explist )? (
    <KEYWORD_NOT_BY> explist )? | goal_action_basic
goal_action_basic ::= <ACHIEVE> relation | <PERFORM> relation |
    <MAINTAIN> relation | <QUERY> relation
relation ::= <IDENTIFIER> explist

```

*expression* ::= *value* | *variable* | *funcall* | *predicate*  
*explist* ::= ( *expression* ( *expression* ) \* )?  
*value* ::= <INTEGER> | <FLOAT> | <STRING>  
*variable* ::= <VARIABLE>  
*funcall* ::= “( ( <SPECIAL\_FUNCTION\_NAME> |  
<IDENTIFIER> ) *explist* “)”  
*predicate* ::= “( <FACT> <IDENTIFIER> *explist* “)” | “(”  
<RETRIEVE> <IDENTIFIER> *explist* “)” | “(”  
<ACHIEVE> <IDENTIFIER> *explist* “)”  
*errorSkipTo* ::= JAVA code  
*getMatchedText* ::= JAVA code  
*getSpecialTokenText* ::= JAVA code

## Notasi bahasa Mobile UNITY

$\square$	Separation of statements
$\parallel$	Parallel execution
$[]$	Sequence execution
$x := e \text{ if } r$	Assignment
$s :: x := e \text{ if } r$	Labeled assignment
$s :: \langle s_1; s_2; \dots; s_n \rangle \text{ if } r$	Transaction
<b>inhibit</b> $s \text{ when } r$	Inhibiting clause
$x := e \text{ reacts-to } r$	Reactive statement
$A.x \approx B.y \text{ when } r$	Shared variable
$A.x \leftarrow B.y \text{ when } r$	Read-only shared variable
<b>engage</b> $(A.x, B.y)$	Engage clause
<b>when</b> $r$	
<b>value</b> $\delta_1, \delta_2$	
<b>disengage</b> $(A.x, B.y)$	Disengage clause
<b>when</b> $r$	
<b>value</b> $\delta_1, \delta_2$	
$A.s \} \{ B.t \text{ when } r$	Exclusive coselection
$\text{coexecute}(A.s, B.t, r)$	Coexecution
$\text{xcoexecute}(A.s, B.t, r)$	Exclusive coexecution

```

System <system_name>
  program <program_name>
    declare ...
    initially ...
    assign ...
  end
  Components
  ...
  Interactions
  ...
end

```

## Notasi bahasa Z

**Names**

a,b	Identifiers
d,e	Declarations (e.g., a: A; b, ...: B ...)
f,g	Functions
m,n	Numbers
p,q	Predicates
s,t	Sequences
x,y	Expressions
A,B	Sets
C,D	Bags
Q,R	Relations
S,T	Schemas
X	Schema text (e.g., d, d   p, or S)

**Definitions**

a == x	Abbreviation definition
a ::= b   ...	Free type definition
[a]	Introduction of a given set (or [a,...])
a _	Prefix operator
_ a	Postfix operator
_ a _	Infix operator

**Logic**

true	Logical true constant
false	Logical false constant
¬p	Logical negation, <i>not</i>
p ∧ q	Logical conjunction, <i>and</i>
p ∨ q	Logical disjunction, <i>or</i>
p ⇒ q	Logical implication
p ⇔ q	Logical equivalence
∀X • q	Universal quantification

$\exists X \bullet q$  Existential quantification

**(let**  $a == x; \dots \bullet p$ ) Local definition

### Sets and Expressions

$x = y$  Equality

$x \neq y$  Inequality

$x \in A$  Set membership

$x \notin A$  Non-membership

$\emptyset$  Empty set

$A \subseteq B$  Set inclusion

$A \subset B$  Strict set inclusion

$\{x, y, \dots\}$  Set display

$\{X \bullet x\}$  Set comprehension

$(\lambda X \bullet x)$  Lambda expression

**(let**  $a == x; \dots \bullet y$ ) Local definition

**If**  $p$  **then**  $x$  **else**  $y$  Conditional expression

$(x, y, \dots)$  Tuple

$(x, y)$  Pair

$A \times B \times \dots$  Cartesian product

$\mathbb{P}A$  Power set

$A \cap B$  Set intersection

$A \cup B$  Set union

$A \setminus B$  Set difference

first  $x$  First element of an ordered pair

second  $x$  Second element of an ordered pair

$\#A$  Number of elements in a set

### Relations

$A \leftrightarrow B$  Binary relation ( $\mathbb{P}(A \times B)$ )

$A \mapsto b$  Maplet  $((a, b))$

**dom**  $R$  Domain of a relation

<b>ran</b> R	Range of a relation
$Q \circlearrowleft R$	Forward relational composition
$Q \circlearrowright R$	Backward relational composition ( $R \circlearrowleft Q$ )
$A \triangleleft R$	Domain restriction
$A \triangleleft R$	Domain anti-restriction
$A \triangleright R$	Range restriction
$A \triangleright R$	Range anti-restriction
$R \{ A \}$	Relational image
$R^{-1}$	Inverse of relation
$R^{+}$	Transitive closure
$Q \oplus R$	Relational overriding
$A \underline{R} b$	Infix relation

### Functions

$A \rightarrow B$	Partial functions
$A \rightarrow B$	Total functions
$A \rightarrowtail B$	Partial injections
$A \rightarrowtail B$	Total injections
$A \xrightarrow{\sim} B$	Bijections
$f x$	Function application (or $f(x)$ )

### Numbers

$\mathbb{Z}$	Set of integers
$\mathbb{N}$	Set of natural numbers $\{ 0, 1, 2, \dots \}$
$\mathbb{N}_1$	Set of strictly positive numbers $\{ 1, 2, \dots \}$
$m+n$	Addition
$m-n$	Subtraction
$m*n$	Multiplication
$m \mathbf{div} n$	Division
$m \mathbf{mod} n$	Remainder (modulus)
$m \leq n$	Less than or equal



$m < n$	Less than
$m \geq n$	Greater than or equal
$m > n$	Greater than
$m .. n$	Number range
$\min A$	Minimum of a set of numbers
$\max A$	Maximum of a set of numbers

### Sequences

$\text{seq } A$	Set of finite sequences
$\text{seq}_1 A$	Set of non-empty finite sequences
$\text{iseq } A$	Set of finite injective sequences
$\diamond$	Empty sequence
$\langle x, y, \dots \rangle$	Sequence display
$s \hat{\ } t$	Sequence concatenation
$\text{head } s$	First element of a sequence
$\text{tail } s$	All but the head element of a sequence
$\text{last } s$	Last element of a sequence
$\text{front } s$	All but the last element of a sequence
$s \text{ in } t$	Sequence segment relation

### Schema Calculus

$S \equiv [ X ]$	Horizontal schema
$[ T; \dots   \dots ]$	Schema inclusion
$z.a$	Component selection (given $z:S$ )
$\Theta S$	Binding
$\neg S$	Schema negation
$S \wedge T$	Schema conjunction
$S \vee T$	Schema disjunction
$S \S T$	Schema composition
$S \gg T$	Schema piping

**Conventions**

$a?$	Input to an operation
$a!$	Output from an operation
$a$	State component before an operation
$a'$	State component after an operation
$S$	State schema before an operation
$S'$	State schema after an operation
$\Delta S$	Change of state
$\Xi S$	No change of state

**Axiomatic Definition**

d	_____
p	_____

**Schema**

S	_____
d	_____
p	_____

**Generic Definition**

[a, ...]	_____
d	_____
p	_____