

## PEMBANGUNAN PERISIAN UNTUK SISTEM TERBENAM MASA NYATA MENGUNAKAN PRINSIP-PRINSIP KEJURUTERAAN PERISIAN

Dyg. Norhayati Abg. Jawawi, Radziah  
Mohamad, Safaai Deris  
Jabatan Kejuruteraan Perisian  
Fakulti Sains Komputer dan Sistem  
Maklumat, Universiti Teknologi Malaysia  
81310 UTM, Skudai

Rosbi Mamat  
Jabatan Kejuruteraan Mekatronik dan Robotik  
Fakulti Kejuruteraan Elektrik  
Universiti Teknologi Malaysia  
81310 UTM, Skudai

### ABSTRAK

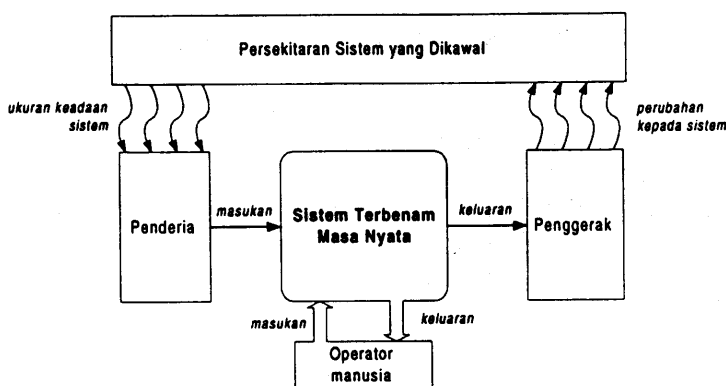
Perbezaan utama perisian untuk sistem masa nyata dengan perisian untuk sistem bukan masa nyata ialah ia mesti menghasilkan keluaran yang tepat dalam batasan masa yang terhad dan kegagalan menghasilkan keluaran dalam batasan masa yang tertentu akan menyebabkan penurunan prestasi atau kegagalan sistem. Perbezaan ini merumitkan proses pembangunan perisian bagi sistem terbenam masa nyata. Di samping itu, kebelakangan ini perisian sistem terbenam masa nyata menjadi semakin canggih dan rumit, untuk menghasilkan sistem perisian terbenam masa nyata yang berkualiti, teknik-teknik kejuruteraan perisian perlu digunakan. Kertas kerja ini bertujuan untuk membincangkan beberapa isu dalam pembangunan perisian sistem terbenam masa nyata dan bagaimana pendekatan profesional kejuruteraan perisian boleh membantu pembangunan perisian terbenam masa nyata ini. Perisian kawalan bagi robot memanjat dinding yang sedang dibangunkan di UTM akan dijadikan sebagai kajian kes dalam kertas kerja ini.

**Kata Kunci :** Kejuruteraan perisian, sistem masa nyata, sistem terbenam dan perisian robot memanjat dinding.

### 1.0 Pengenalan

Sistem terbenam adalah sistem yang mana mikrokomputer adalah salah satu komponen sistem, tetapi pengguna tidak melihat keseluruhan sistem sebagai satu sistem komputer. Sistem terbenam merangkumi skala aplikasi yang luas, contohnya jam tangan dan ketuhar gelombang mikro merupakan sistem terbenam berskala-kecil, alat pengurusan rangkaian perhubungan pula merupakan sistem terbenam berskala-sederhana sementara sistem kawalan ketenteraan merupakan sistem terbenam berskala-besar. Kebanyakan sistem masa nyata adalah sistem terbenam, yang dipanggil sistem terbenam masa nyata. Sistem masa nyata ialah sistem di mana ketepatan sistem bukan sahaja bergantung kepada hasil logik sesuatu proses tetapi juga bergantung kepada masa sesuatu hasil proses itu dikeluarkan.

Model generik bagi sistem terbenam masa nyata ditunjukkan dalam Rajah 1. Sistem ini berinteraksi dengan persekitaran melalui penderia dan penggerak. Penderia digunakan untuk menentukan keadaan persekitaran dan penggerak digunakan untuk mengawal sistem tersebut berdasarkan status persekitaran yang dibaca. Biasanya perisian terbenam masa nyata ditulis untuk aplikasi yang melibatkan pemantauan proses dan kawalan perkakasan.



Rajah 1: Model sistem terbenam masa nyata.

Mesin dan peralatan yang dikawal menggunakan sistem terbenam masa nyata menjadi semakin canggih dan cerdik, kebiasaannya sistem jenis ini terdiri daripada gabungan banyak fungsi dalam satu produk. Ekoran itu, sistem terbenam masa nyata menjadi bertambah rumit yang menyebabkan pertambahan dari segi keperluan pemprosesan, penggunaan teknologi pengamiran skala sangat besar atau *Very Large-Scale Integration (VLSI)* dan integrasi perkakasan-perisian. Penggunaan mikropemproses yang berkebolehan tinggi dalam sistem terbenam membolehkan lebih banyak pemprosesan dilakukan oleh perisian. Ini menyebabkan pembangunan perisian sistem terbenam masa nyata menjadi semakin penting.

Untuk menghasilkan satu perisian terbenam masa nyata yang berkualiti, penggunaan prinsip kejuruteraan perisian dalam pembangunan perisian ini adalah amat diperlukan. Metodologi kejuruteraan perisian yang bersesuaian dengan sistem-sistem tertentu boleh memastikan proses pembangunan sesuatu perisian terurus dan juga boleh menghasilkan perisian berkeboleharapan tinggi. Tujuan kertas kerja ini adalah untuk membincangkan bagaimana prinsip kejuruteraan perisian boleh membantu pembinaan perisian terbenam masa nyata yang berskala kecil. Sistem Robot Memanjat Dinding yang sedang dalam pembangunan di Universiti Teknologi Malaysia (UTM) akan dijadikan kajian kes untuk membincangkan isu-isu ini.

Kandungan kertas kerja ini diatur seperti berikut. Bahagian 2 akan membincangkan spesifikasi perkakasan elektronik bagi sistem Robot Memanjat Dinding (RMD) UTM dan isu-isu dari spesifikasi RMD yang mempengaruhi pembangunan perisian bagi perisian kawalan robot tersebut akan dikupas satu-persatu. Bagaimana prinsip kejuruteraan perisian boleh digunakan untuk pembangunan perisian kawalan RMD akan dihuraikan dan dibincangkan di Bahagian 3. Bahagian 4 akan memberi kesimpulan bagi kertas kerja ini.

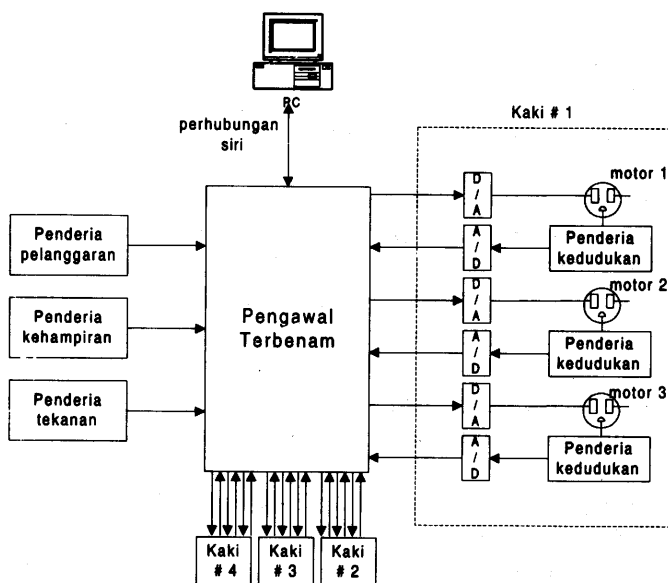
## 2.0 Spesifikasi Sistem Pengawal Robot Memanjat Dinding

Robot Memanjat Dinding (RMD) UTM, mempunyai satu badan dan empat pasang kaki yang serupa. Badan robot membawa pengawal terbenam dan komponen-komponen elektronik lain bagi robot tersebut. Setiap kaki robot mempunyai tiga sendi yang memberi tiga darjah kebebasan bergerak untuk setiap kaki. Setiap sendi digerakkan oleh motor arus-terus dan penderia kedudukan digunakan untuk mengukur sudut-sudut pergerakan bagi setiap sendi. Pad sedutan pada hujung setiap kaki robot membolehkan robot melekat pada dinding. Tekanan di dalam pad sedutan tersebut diukur dengan menggunakan penderia tekanan. Pengesan kehampiran dan pengesan pelanggaran mengesan halangan dipersekitaran robot. Keadaan di persekitaran robot tersebut perlu dipantau setiap setengah saat untuk mengesan halangan-

halangan. Robot ini boleh digerakkan ke depan dan ke belakang dengan melakukan jujukan-jujukan langkah yang telah ditentukan.

Gambarajah blok pengawal terbenam bagi RMD tersebut ditunjukkan pada Rajah 2. Pengawal terbenam robot tersebut berasaskan mikropengawal Intel 80188XL dengan 64Kb EPROM dan 64Kb RAM. Fungsi utama pengawal ini adalah untuk menggerakkan kaki-kaki robot tersebut dengan melakukan jujukan-jujukan langkah yang telah ditentukan semasa operasi memanjat dan memantau persekitaran serta bertindak-balas kepada persekitaran dengan pintar.

Pada setiap masa persampelan, isyarat kawalan yang dihantar kepada motor arus-terus, dikira dengan menggunakan algoritma kawalan terbitan berkadar. Kedudukan semasa untuk setiap sendi pada kaki robot diukur dengan menggunakan penderia kedudukan dan nilainya akan disuap balik kepada pengawal terbenam. Pengiraan bagi isyarat kawalan ini mesti disiapkan dalam masa 50-100 milisaat untuk memastikan pergerakan kaki robot yang betul. Kegagalan menghasilkan isyarat kawalan ini dalam masa tersebut akan mengakibatkan kegagalan robot ini berfungsi. Pengawal terbenam itu juga dihubungkan kepada komputer peribadi jarak jauh melalui perhubungan bersiri untuk menerima arahan-arahan aras tinggi dan menghantar maklumat kepada komputer peribadi.



Rajah. 2: Gambarajah blok pengawal terbenam.

Fungsi utama pengawal terbenam RMD ini boleh dibahagikan kepada empat tugas utama: pandu arah aras tinggi, pemantauan persekitaran, kawalan motor dan perhubungan siri dengan komputer peribadi jarak jauh.

Daripada spesifikasi perisian kawalan RMD yang telah dibincangkan di atas, terdapat beberapa isu yang perlu diketengahkan, isu-isu ini akan mempengaruhi dan menambah kerumitan proses pembangunan perisian robot tersebut. Isu-isunya ialah:

### 2.1 Batasan sistem berskala-kecil.

Sistem RMD ini dikategorikan sebagai sistem terbenam berskala kecil kerana spesifikasi pemprosesnya hanya 16 bit mikropengawal dan ingatan yang kecil, iaitu satu mikropemproses

16 bit dengan 64Kb EPROM dan 64Kb RAM. Di samping itu saiz, berat dan kuasa robot yang terhad juga menyebabkan sistem RMD ini dikelaskan sebagai sistem terbenam berskala-kecil. Pembangunan perisian RMD perlu memenuhi semua keperluan spesifikasi fungsi-fungsi dan prestasi robot dengan kekangan-kekangan (iaitu saiz fizikal, pemproses dan ingatan robot) yang diberikan. Dalam proses pembangunan perisian RMD ini, pemilihan perkakasan-perkakasan dan metodologi untuk digunakan dalam pembangunan perisian mesti mengambil kira kekangan skala sistem yang ditentukan dalam spesifikasi.

## ***2.2 Keserempakan dan pembilangtugasan.***

Sistem RMD ini bertindakbalas dengan persekitarannya dan melakukan proses-proses secara berkala, di mana sistem perlu mencapai maklumat daripada persekitaran melalui penerima-penerima dan komputer peribadi jarak jauh, mengemaskini sistem keadaan dalaman dan mengawal motor-motor pada kaki. Untuk mengawal proses-proses dan bertindak kepada persekitaran yang tidak berketentuan ini, perisian RMD ini memerlukan kemampuan keserempakan dan pembilangtugasan. Untuk merekabentuk sistem yang berkeserempakan dan berbilangtugasan, pereka perisian mesti mengambilkira isu-isu seperti teknik penguraian masalah kepada proses-proses atau tugasan-tugasan dan pengurusan tugasan serempak.

## ***2.3 Sistem masa nyata kritikal.***

Spesifikasi yang menyebabkan sistem RMD satu sistem masa nyata kritikal ialah keperluan sistem untuk bertindakbalas dengan tepat kepada perubahan perisetikaran dalam tempoh masa yang ditetapkan. Oleh itu, ketepatan sistem robot ini tidak bergantung kepada ketepatan fungsi sistem sahaja tetapi juga keluaran yang tepat dalam batasan masa yang terhad. Contohnya, salah satu spesifikasi RMD yang kritikal ialah isyarat kawalan perlu dihantar kepada motor di setiap sendi robot, setiap 100 milisaat. Pengawal RMD ini bukan sahaja dikehendaki mengira isyarat kawalan untuk setiap motor dengan betul tetapi mesti mengira dan menghantar isyarat kawalan tersebut dalam tempoh kurang daripada 100 milisaat. Jika had tempoh ini tidak boleh dipenuhi, robot akan menjadi tidak stabil dan jatuh dari dinding. Penambahan faktor ketepatan masa di samping fungsi robot yang betul merupakan cabaran bagi perekabentuk sistem ini.

Berdasarkan spesifikasi fungsi-fungsi dan spesifikasi masa RMD ini, satu metodologi perisian yang tertentu diperlukan dalam pembinaan perisian ini untuk memastikan spesifikasi masa dan fungsi dipenuhi. Metodologi ini perlu menyokong dan mengintegrasikan spesifikasi masa di sepanjang kitaran hayat pembangunan perisian robot tersebut.

## ***2.4 Spesifikasi RMD yang berkembang dengan beransur-ansur.***

Spesifikasai keperluan RMD yang ditulis akan berkembang kerana fungsi-fungsi baru dan kepintaran baru akan ditambahkan kepada perisian robot ini. Oleh itu, kecanggihan perisian kawalan robot ini semakin meningkat setiap kali fungsi dan kepintaran ditambah kepada perisian robot tersebut. Walaupun fungsi dan kecanggihan robot ini bertambah, tetapi pengawal terbenam ini tetap dikelaskan sebagai sistem terbenam berskala-kecil. Spesifikasi keperluan RMD yang berkembang secara beransur-ansur ini perlu ditangani dengan betul kerana perubahan keperluan fungsi robot akan memberi kesan kepada keperluan bukan fungsi seperti masa dan prestasi perisian. Kebolehsenggaraan adalah satu lagi isu yang timbul daripada keperluan yang berubah-ubah ini, di mana perisian robot ini perlu anjal dan mudah disenggara. Oleh itu, pemilihan teknik analisa spesifikasi dan teknik rekabentuk yang sesuai

untuk digunakan dalam perisian RMD ini adalah penting untuk menghasilkan perisian robot yang mempunyai ciri-ciri kebolehsuaian dan kebolehesanan yang tinggi.

## 2.5 Perkakasan sasaran.

Satu lagi isu yang perlu dipertimbangkan semasa pembangunan perisian RMD ini ialah sistem perkakasan yang menjadi sasaran untuk perisian tersebut. Pembangunan perisian RMD tanpa mengambil kira ciri-ciri perkakasan, adalah tidak mencukupi. Keseluruhan rekabentuk sistem robot perlu dipertimbangkan secara tekal dalam keputusan merekabentuk perisian robot tersebut. Maka, metodologi RMD ini perlu menyediakan mekanisma yang membolehkan pembangunan perkakasan dan perisian dilakukan secara serentak dan tekal.

Isu-isu yang dibincangkan di atas akan menambahkan kerumitan proses pembangunan perisian dan merupakan cabaran kepada jurutera perisian RMD tersebut. Untuk memastikan kualiti perisian kawalan RMD ini, proses pembangunan robot ini perlu mengambil kira isu-isu ini di sepanjang kitaran hayat pembangunan perisian RMD tersebut.

## 3.0 Prinsip Kejuruteraan Perisian dalam Pembinaan Perisian RMD

### 3.1 Metodologi Kejuruteraan Perisian untuk Sistem Masa Nyata

Tujuan kejuruteraan perisian (KP) adalah untuk mempromosikan pendekatan tertentu bagi menyelesaikan masalah-masalah perisian (Ghenzzi, Mehdi dan Dino, 1997). Metodologi kejuruteraan perisian memberi panduan dalam proses pembangunan perisian dan menyediakan tatatanda untuk proses analisis, rekabentuk dan dokumentasi dalam pembangunan perisian.

Terdapat beberapa metodologi kejuruteraan perisian yang telah diperkenalkan seperti Analisis Berstruktur dan Rekabentuk Berstruktur (*Structured Analysis and Structured Design-SA/SD*), Rekabentuk Berstruktur Jackson (*Jackson Structured Design-JSD*), Rekabentuk berorientasikan objek (*Object-Oriented Design-OOD*) dan lain-lain (Pressman, 1992). Walaubagaimanapun, kebanyakan metodologi ini tidak memenuhi keperluan pembangunan perisian masa nyata seperti perisian RMD ini. Ini kerana metodologi-metodologi ini tidak mempertimbangkan keperluan bukan fungsi seperti keperluan masa semasa proses pembangunan perisian tersebut (Burn and Wellings, 1994). Oleh itu, beberapa metodologi kejuruteraan perisian baru telah dicadangkan khusus untuk perisian sistem masa nyata. (Bennett, 1994) (Calvez, 1993).

Metodologi KP baru tersebut, terhasil dengan melakukan beberapa perubahan terhadap metodologi yang sedia ada. Ia menyediakan kemudahan yang boleh membantu pembangunan sistem masa nyata sebagai contoh metodologi pembangunan berstruktur untuk sistem masa nyata (*Structured Development for Real-time System-SDRTS* oleh Ward and Mellor, 1985) berasal dari metodologi *Structured Analysis and Structured Design-SA/SD*. Metodologi-metodologi untuk sistem masa nyata juga terhasil dengan membangunkan metodologi baru dari mula. Kebiasaannya metodologi jenis ini berdasarkan aplikasi-aplikasi tertentu, contohnya seperti metodologi *ObjectGeode* khusus untuk sistem masa nyata sistem telekomunikasi, metodologi *Aerospace System Simulation, Engineering and Test (ASSET)* untuk sistem aeroangkasa dan *Software Requirement Engineering Methodology (SREM)* untuk sistem bersenjata automatik. Dalam bahagian 3 kertas kerja ini, satu metodologi yang dinamakan Metodologi Hibrid atau Metodologi Campuran digunakan dalam pembangunan perisian RMD

ini, akan dijelaskan dan bagaimana metodologi tersebut dapat membantu pembangunan perisian ini juga akan dibincangkan.

### 3.2 Metodologi Hibrid

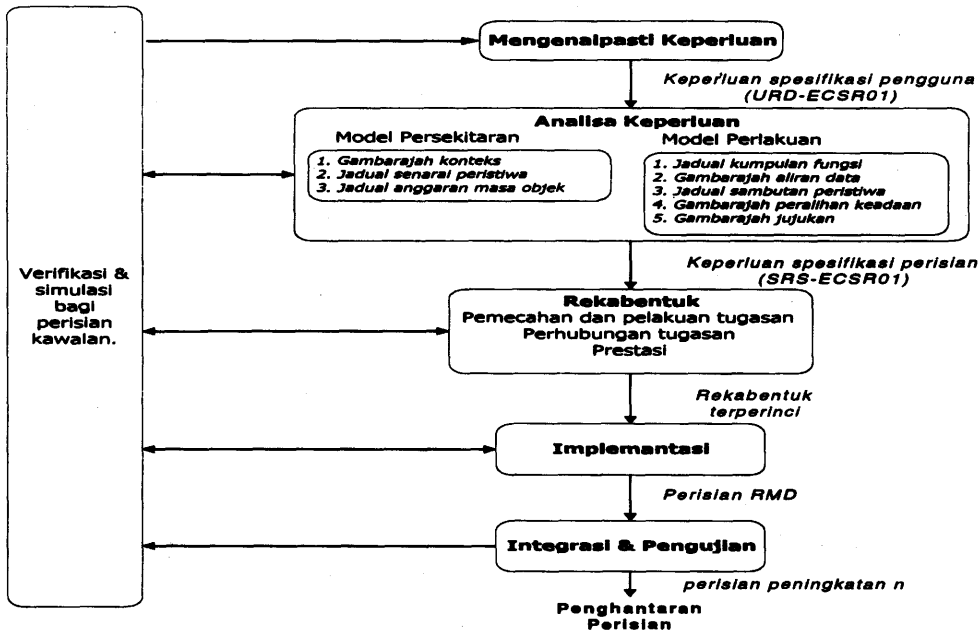
Metodologi hibrid dicadangkan untuk pembangunan perisian RMD ini. Kaedah-kaedah dan tatatanda yang terdapat dalam metodologi hibrid ini dipinjam dari metodologi-metodologi *Structured Development for Real-Time Systems* (SDRTS) atau dipanggil metodologi Ward-Mellor (Ward and Mellor, 1985), *Unified Modeling Language for Real-Time* (UML-RT) (Douglass, 1998) dan *Hard Real-Time Hierarchical Object Oriented Design* (HRT-HOOD) (Burns and Wellings, 1995). Jadual 1 menerangkan model-model dan teknik-teknik yang digunakan di dalam spesifikasi dan rekabentuk metodologi hibrid. Metodologi campuran digunakan dalam spesifikasi dan rekabentuk perisian kawalan robot ini. Kebaikan utama menggunakan metodologi hibrid ini, ialah kaedah-kaedah dan tatatanda yang dipilih dari ketiga-tiga metodologi boleh mewakili semua bentuk spesifikasi robot baik spesifikasi fungsi (contohnya kelakuan robot) mahupun spesifikasi bukan fungsi (contohnya pemasaan dan prestasi sistem robot). Keburukan utama menggunakan metodologi ini ialah tiadanya sokongan peralatan CASE, namun demikian disebabkan skala sistem RMD ini, peralatan *Computer Aided Software Engineering* (CASE) tidak begitu diperlukan, ini boleh ditangani secara manual.

<i>Num.</i>	<i>Fasa</i>	<i>Tenik dan peralatan</i>	<i>Metodologi</i>
1.	<b>Model persekitaran</b>	Penstrukturian luar-ke-dalam	Ward-Mellor
		<i>Peralatan</i> Gambarajah konteks Jadual senarai peristiwa Jadual anggaran masa objek	Ward-Mellor Ward-Mellor HRT-HOOD
2.	<b>Model perlakuan</b>		
2.1.	First-level decomposition	Model struktur-objek	HRT-HOOD
		<i>Peralatan</i> Gambarajah aliran data Jadual kumpulan fungsi	Ward-Mellor HRT-HOOD
2.2.	Detail functions decomposition	<i>Tools</i> Gambarajah aliran data	Ward-Mellor
2.3.	Non-functional decomposition	<i>Tools</i> Jadual sambutan peristiwa Gambarajah peralihan keadaan Gambarajah jujukan	Ward-Mellor Ward-Mellor UML-RT
3.	<b>Rekabentuk</b>		
3.1.	Pemecahan tugas-tugas dan pelaksanaan tugas	<i>Tools</i> Gambarajah keadaan	UML-RT
	Perhubungan dan persegerakan tugas	<i>Tools</i> Gambarajah tugas	-
	Prestasi masa	<i>Tools</i> Jadual kepentingan	HRT-HOOD

**Jadual 1 :** Model dan teknik yang digunakan dalam fasa spesifikasi dan rekabentuk perisian RMD.

Disebabkan tabii spesifikasi RMD ini yang berkembang dengan beransur-ansur, proses model peningkatan adalah sesuai untuk digunakan di sini. Proses model peningkatan menggabungkan

elemen-elemen model jujukan lurus dengan model prototaip berulang, yang membolehkan jurutera perisian menyiapkan perisian secara beransur-ansur dalam setiap versi perisian. Kitaran hayat bagi pembangunan perisian RMD untuk setiap model peningkatan ditunjukkan di Rajah 3. Terdapat lima fasa dalam kitaran hayat perisian RMD ini, bermula dengan fasa *mengenalpasti keperluan*, diikuti oleh *analisa keperluan*, *rekabentuk*, *implimentasi* dan fasa yang terakhir *integrasi dan pengujian*. Dalam setiap fasa kitaran hayat perisian RMD terdapat aktiviti-aktiviti tertentu dan setiap aktiviti-aktiviti terdapat produk-produk yang dihasilkan. Produk-produk dari fasa-fasa tersebut ditunjukkan dengan perkataan condong di tepi anak panah yang keluar dari setiap fasa di dalam Rajah 3. Aktiviti-aktiviti dan notasi-notasi yang digunakan dalam aktiviti analisa dan rekabentuk perisian tersebut juga di senaraikan di Rajah 3.



Rajah 3: Kitaran Hayat pembangunan perisian RMD.

Untuk tujuan verifikasi dan pengujian perisian RMD, satu sistem simulasi dibangunkan. Sistem ini membolehkan proses verifikasi dan validasi dilakukan di sepanjang kitaran hayat pembinaan setiap versi  $n$  peningkatan perisian.

### 3.3 Fasa Mengenalpasti Keperluan dan Analisa Perisian

Fasa pertama dilakukan dalam pembangunan perisian ialah fasa mendapatkan keperluan sistem dari pelanggan atau pengguna, dari keperluan ini rekabentuk dan implementasi boleh dilakukan. Kini fasa mengenalpasti dan menganalisa keperluan sesuatu sistem, terutamanya sistem terbenam masa nyata menjadi semakin sukar. Ini adalah kerana peningkatan taraf kesukaran sesuatu sistem masa nyata, bukan sahaja merumitkan sistem fungsi tetapi juga sistem bukan fungsi seperti pemasaan dan keselamatan. Oleh itu, isu-isu spesifikasi yang dibincangkan di Bahagian 2 perlu ditekankan dalam proses mengenalpasti dan menganalisa keperluan perisian RMD ini.

Projek ini dimulakan dengan mengenalpasti keperluan sistem kawalan robot dari pengguna. Tujuan fasa mengenalpasti keperluan, adalah untuk mendapatkan huraian terperinci sistem kawalan RMD tersebut, di mana dalam proses ini berlaku perpindahan maklumat dan kepakaran mengenai RMD empat kaki UTM dari Kumpulan Penyelidikan Robot Bergerak UTM (KPRB-UTM). Setelah keperluan robot dikenalpasti, keperluan perisian RMD tersebut

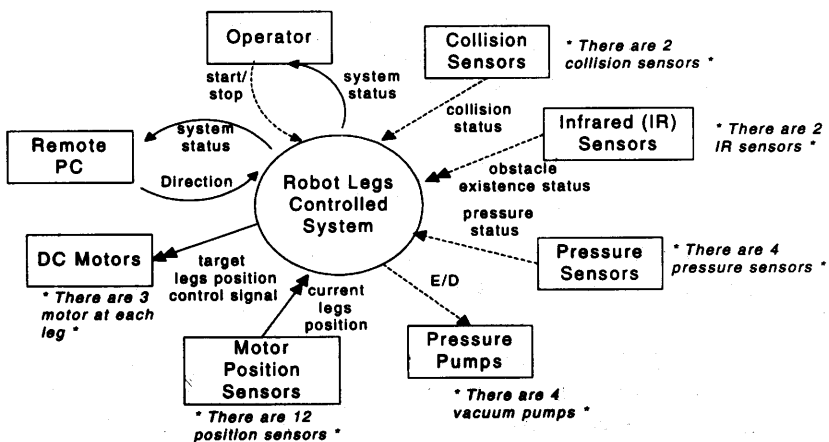
perlu ditakrifkan dan diuraikan supaya kedua-dua pihak, KPRB-UTM dan pembangun perisian, bersetuju ke atas keperluan perisian sistem robot yang akan dibangunkan. Dokumen keperluan pengguna yang bertajuk *User Requirements Definition Phase for Embedded Control Software of UTM Four-Legged Wall Climbing Robot (URD-ECSR01)* telah disediakan untuk tujuan tersebut. Dokumen ini disediakan dengan kerjasama KPRM-UTM, mengikut format dan garis panduan spesifikasi keperluan yang disediakan oleh *European Space Agency* bertaraf IEEE/ANSI 830-1984 (Mazza *et al.*, 1996).

Setelah keperluan perisian RMD dikenalpasti, keperluan ini perlu dianalisa untuk mendapatkan model logikal bagi spesifikasi tersebut; model logikal ini juga dipanggil analisa keperluan perisian. Untuk membangunkan model ini kaedah kejuruteraan perisian tertentu perlu digunakan untuk menghasilkan analisa keperluan perisian yang tepat. Fasa analisa keperluan dalam metodologi hibrid ini dilakukan dalam dua model iaitu model persekitaran dan model perlakuan. Tatatanda yang digunakan dalam kedua-dua model disenaraikan di Rajah 3. Disamping menggunakan tatatanda separa formal yang senaraikan ini, spesifikasi perisian RMD ini juga ditulis secara formal menggunakan kaedah spesifikasi formal Z. Tujuan spesifikasi perisian RMD ini ditulis dalam kaedah formal ini adalah untuk meningkatkan keboleharapan perisian tersebut, disamping membolehkan perisian tersebut disemak pada tahap awal pembangunan perisian. Rajah 4a) hingga Rajah 4c) menunjukkan sebahagian dari kaedah-kaedah yang digunakan dalam spesifikasi perisian RMD.

### 3.4 Fasa Rekabentuk Perisian

Dua isu penting yang telah dibincangkan di Bahagian 2 perlu diambil kira di dalam fasa rekabentuk perisian RMD ini ialah isu masa nyata dan isu keserempakan dan pembilangtugasan. Untuk mempertimbangkan isu-isu ini, rekabentuk robot ini dilakukan dalam tiga bahagian utama iaitu:

- i) Pemecahan tugasan-tugasan dan pelakuan tugasan.
- ii) Perhubungan dan persegerakan tugasan.
- iii) Prestasi masa.



Rajah 4a): Gambarajah Konteks perisian RMD.

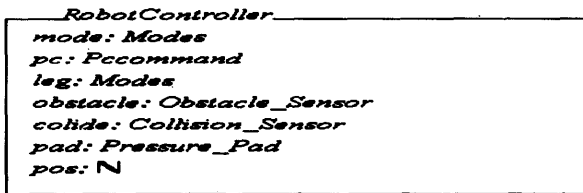
Daripada model persekitaran dan perlakuan yang dianalisa dalam fasa analisa keperluan, lima proses atau modul yang berserempak boleh dikenal pasti dengan jelas iaitu *operator console*, *monitor environment*, *serial communication*, *motor control* dan *high level control*. Lima proses ini perlu dipecahkan atau diuraikan kepada proses-proses, tugasan-tugasan dan fungsi-fungsi



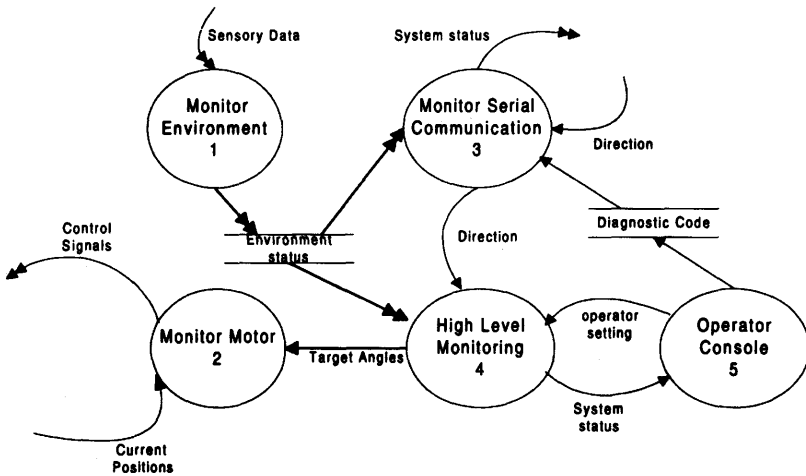
yang lebih kecil supaya senang untuk dijadualkan dan diuruskan dengan menggunakan kernel masa nyata. Contohnya, Rajah 5 menunjukkan bagaimana proses *environment monitoring* diuraikan kepada tiga tugas dan fungsi-fungsi kecil.

```
Modes ::= Idle | Suctioning_Pad | Scanning_Collision | Scanning_Obstacle
| Scanning_Leg | Moving | Stop_Move | Waiting_Pc | Scanning_Body
YesNo ::= Yes | No
OnOff ::= On | Off
```

```
RobotState == {Moving, Stop_Move, Idle}
Collision_Sensor == {On, Off}
Obstacle_Sensor == {On, Off}
Pressure_Pad == {Yes, No}
PressureActivate ::= active | passive
Pccommand ::= Start_Move | Stop
```

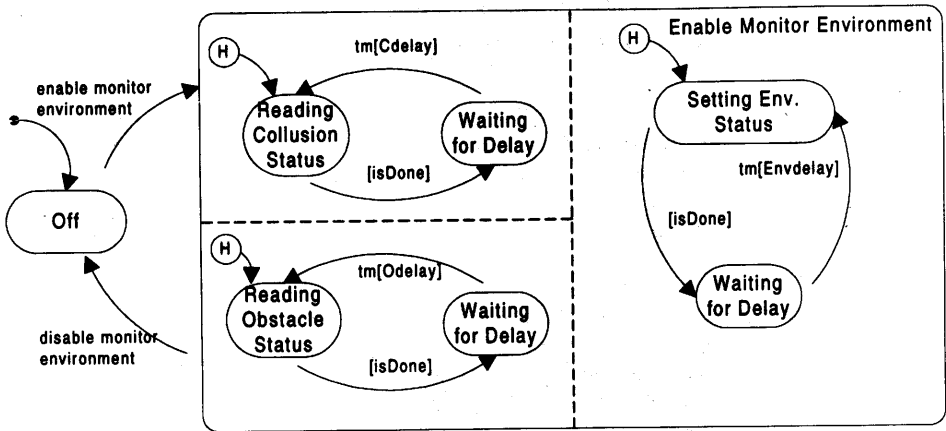


Rajah 4b) : Spesifikasi asas perisian RMD menggunakan kaedah formal Z.

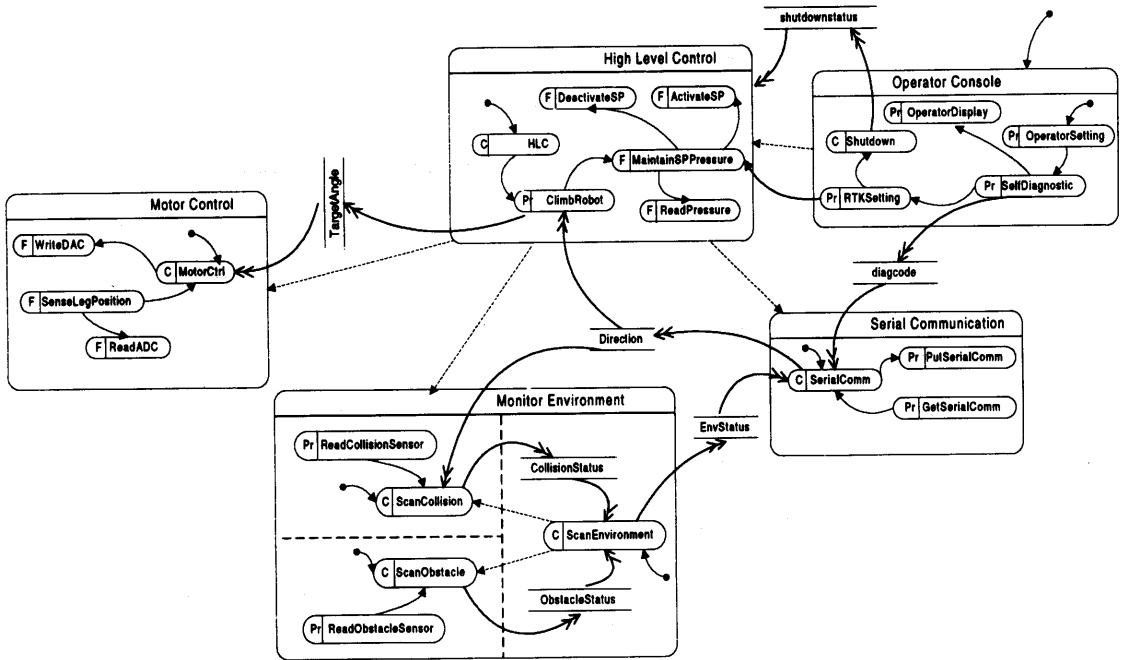


Rajah 4c): Aras pertama gambarajah aliran data bagi perisian RMD.

Setelah semua tugas-tugas dan fungsi-fungsi dikenal pasti, perserentakkan tugas-tugas dan data yang dikongsi antara tugas-tugas perlu dimodelkan dengan jelas. Model rekabentuk perisian RMD ini, dimodelkan dengan menggunakan gambarajah tugas, seperti ditunjukkan dalam Rajah 6. Daripada Rajah 6, rekabentuk perisian RMD dibahagikan kepada lima proses, dan dalam setiap proses ada subkeadaan-subkeadaan. Setiap subkeadaan akan ditanda "C", "Pr" atau "F". "C" menunjukkan subkeadaan yang tugasannya berpusing-pusing atau dalam gelung sepanjang perisian itu dilaksanakan. Tanda "Pr" menunjukkan subkeadaan ini tidak boleh disampuk oleh subkeadaan lain dan tanda "F" pula menunjukkan subkeadaan fungsi biasa yang boleh disampuk oleh mana-mana sub keadaan.



Rajah 5 : Gambarajah keadaan untuk proses *environment monitoring*.



Rajah 6 : Gambarajah tugas.

Terdapat tujuh tugas yang berserempak dikenalpasti dari gambarajah tugas perisian RMD. Untuk menjadualkan pelaksanaan ketujuh-tujuh tugas ini dengan menggunakan kernel masa nyata *preemptive*, kepentingan setiap tugas perlu diketahui. Dengan menggunakan teknik *Rate Monotonic Scheduling (RMS)* (Burn dan Wellings, 1996), kepentingan setiap tugas dapat ditentukan berdasarkan tempoh pelaksanaan tugas. Jadual 2 menunjukkan ciri-ciri masa bagi setiap tugas.

Berdasarkan kepentingan dari Jadual 2, tugas yang paling penting dalam sistem perisian ini adalah tugas "*Motorctrl*" manakala tugas "*Shutdown*" mempunyai kepentingan paling rendah.

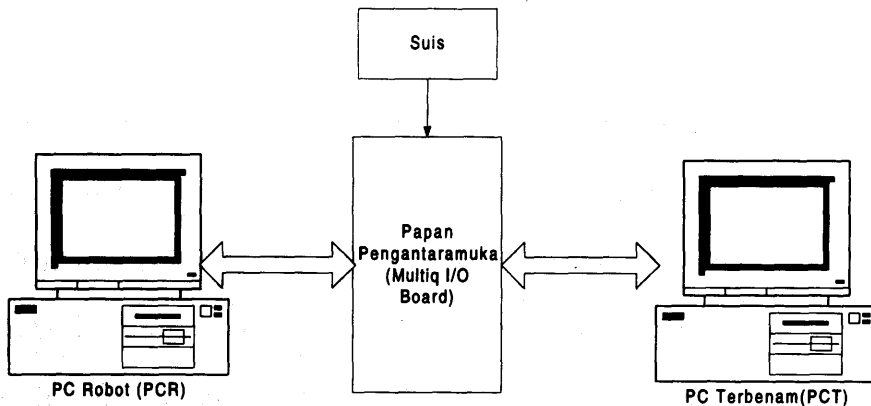
<i>Tugasan</i>	<i>Anggaran masa</i>	<i>Kepentingan</i>
<i>Motorctrl</i>	50 milisaat – 100 milisaat	7
<i>ScanEnvironment</i>	1 saat – 10 saat	5
<i>ScanCollision</i>	0.5 saat – 20 saat	6
<i>ScanObstacle</i>	2 saat – 30 saat	3
<i>SerialComm</i>	1 saat – 10 saat	4
<i>HLC</i>	10 saat – 30 saat	2
<i>Shutdown</i>	30 saat – 60 saat	1

Jadual 2: Ringkasan ciri-ciri masa bagi setiap tugasan.

### 3.5 Fasa Implementasi, Integrasi dan Pengujian

Alatan-alatan perisian yang digunakan dalam fasa implementasi ialah pengkompil Borland C/C++ 3.1, penempat ROM untuk menghasilkan kod yang boleh ditempatkan ke dalam ROM dan kernel masa nyata  $\mu$ C/OS-II. Pembangunan perisian dilakukan dengan menggunakan komputer peribadi di bawah sistem pengendalian MS-DOS/WINDOW95. Pembangunan perisian yang menggunakan komputer peribadi ini membolehkan penggunaan alatan pembangunan perisian yang murah dan berkualiti. Oleh kerana mikropemproses dalam pengawal terbenam bagi robot ini adalah serasi dengan mikropemproses di dalam komputer peribadi, ia membolehkan kerja-kerja pengujian dan penyahpejatan dilakukan di atas komputer peribadi tersebut.

Sebelum perisian tersebut ditempatkan pada robot, pengujian perisian mestilah dilakukan di atas komputer peribadi. Satu perisian penyelakuan ditulis untuk membolehkan pengujian ini dilakukan. Persekitaran bagi strategi penyelakuan RMD ini ditunjukkan di Rajah 7. Pengujian ini dilakukan dengan menggunakan dua komputer peribadi yang di panggil PC robot (PCR) dan PC terbenam (PCT). PCR menyelakukan perkakasan robot yang dikawal oleh perisian tersebut. PCT melaksanakan perisian RMD yang telah dibangunkan. Kedua-dua komputer peribadi ini dihubungkan oleh papan pengantaramuka *Multiq I/O*.



Rajah 7 : Persekitaran bagi penyelakuan RMD.

Fungsi bagi setiap proses-proses atau modul-modul dalam perisian RMD ini diuji secara beransur-ansur dan berasingan antara setiap modul semasa proses penulisan kod dilakukan. Manakala integrasi antara modul-modul dalam persekitaran yang berbagai-bagai diuji menggunakan strategi penyelakuan RMD. Prestasi masa perisian dianggarkan melalui sukatan masa yang disukai dari penyelakuan komputer peribadi.

Penyelakuan adalah kaedah yang digunakan dalam pembangunan sistem perisian, yang boleh meningkatkan kualiti, mengurangkan kos pembangunan perisian dan menyingkatkan masa pembangunan sesuatu perisian (Vetterli, 1999). Penyelakuan bagi RMD ini boleh menghindarkan kerosakan pada perkakasan robot semasa pengujian dilakukan, setetusnya boleh mengurangkan kos pembangunan, kerana kos bagi perkakasan robot adalah mahal.

Antara kebaikan-kebaikan lain yang ditawarkan oleh penyelakuan RMD ini ialah:

- i. Pengujian perisian boleh dilakukan serentak dengan pembangunan perkakasan robot.
- ii. Komunikasi melalui perkakasan papan antara dua komputer peribadi membolehkan perisian kawalan RMD tersebut diselenggarakan berasingan dari perisian penyelakuan.
- iii. Kod-kod bagi antaramuka papan perkakasan boleh menghasilkan masa yang hampir sama dengan perkakasan antaramuka robot yang sebenar.

Daripada strategi pengujian ini, perisian RMD dapat diuji dari segi fungsi-fungsi dan perlakuan robot; anggaran prestasi masa bagi tugas yang kritikal; perhubungan dan persegerakan tugas-tugas dan ukuran saiz perisian.

#### 4.0 Kesimpulan

Satu metodologi pembangunan perisian sistem terbenam masa nyata telah dibincangkan dalam kertas kerja ini. Metodologi ini telah digunakan dalam satu kajian kes perisian kawalan robot memanjat dinding, dan didapati metodologi tersebut boleh menangani isu-isu dalam pembangunan perisian sistem terbenam masa nyata. Prinsip kejuruteraan perisian yang merangkumi kaedah, alatan, strategi, teknik dan kitaran hayat yang digunakan dalam pembangunan perisian robot ini, mampu menyokong pembangunan perisian kawalan robot tersebut. Di dalam hal ini, yang paling penting adalah spesifikasi perisian robot tersebut dapat disahkan dan ditentusahkan oleh pengguna dan pembangun pada awal proses pembangunan sistem robot tersebut, walaupun perkakasan robot tersebut belum disiapkan lagi.

#### 5.0 Rujukan

- Bennett S. (1994). "Real-Time Computer Control: An Introduction", 2<sup>nd</sup> edition, UK: Prentice Hall.
- Burns A., Wellings A. J. (1995). HRT-HOOD: A Structured Design Method for Hard Real-time System, Volume 3, Elsevier.
- Calvez J. P. (1997), "Embedded Real-time Systems A specification and design Methodology", Report no: BP 60601-44306, IRESTE – University of Nantes, France.
- Douglass B. P. (1998). Real-time UML Developing Efficient Object for Embedded Systems, USA: Addison Wesley.
- Ghenzzi C., Mehdi J. dan Dino M. (1997), "Fundamentals of Software Engineering", Singapore: Prentice Hall.
- Mazza *et al.*, (1996). "Software Engineering Guides". Hertfordshire, U.K.: Prentice Hall.
- Pressman, Roger S. (1997). "Software Engineering A Practitioner 's Approach". Forth Edision. New York, U.S.A.: McGraw-Hill.
- Vetterli C. (1999). "Success with the Power of Simulation", *Embedded Systems Conference*, Fall 99, Lecture no. 506.
- Ward, P. T. and Mellor, S. J. (1985). "Structured Development for Real-Time Systems", Volume 1-3, New York: Yourdon Press.