Jurnal Teknologi, 49(D) Dis. 2008: 61–85
© Universiti Teknologi Malaysia

ANALISIS CAPAIAN WEB CRAWLER DENGAN MENGGUNAKAN ALGORITMA GENETIK

ALI SELAMAT[1], LIM YEE WAY[2] & SITI NURKHADIJAH AISHAH IBRAHIM[3]

Abstrak. Dengan perkembangan teknologi maklumat yang kian mendadak, proses carian maklumat di internet menjadi satu kerja yang sukar dan memakan masa. Enjin carian seperti *Google, LookSmart, Altavista* dan *Yahoo* diperkenalkan dan digunakan untuk memudahkan proses carian maklumat di internet. *Web crawler* adalah komponen utama enjin carian yang digunakan untuk mencari sumber maklumat di internet. Semasa proses pencarian, *web crawler* menghadapi beberapa masalah seperti keputusan carian kurang tepat, tidak terkini, tidak mempunyai perhubungan secara langsung antara pengguna dengan *web crawler* dan lain-lain lagi. Satu *web crawler* pintar yang bernama *UtmCrawler* dibangunkan untuk mengatasi masalah keputusan carian kurang tepat dan masalah tidak mempunyai perhubungan secara langsung antara pengguna dengan *web crawler*. Metodologi yang digunakan untuk membangunkan *UtmCrawler* terdiri daripada beberapa fasa. Pada fasa pemprosesan, *UtmCrawler* akan mengembangkan kata kunci carian yang dimasukkan oleh pengguna dengan menggunakan teknik algoritma genetik (GA). Daripada hasil kajian, keputusan carian yang dijanakan dengan menggunakan GA mempunyai nilai ketepatan yang lebih tinggi, iaitu 95.19% berbanding dengan keputusan carian tanpa menggunakan GA, iaitu 89.07%. Kesimpulannya, *UtmCrawler* yang dibangunkan dapat membekalkan keputusan carian yang lebih tepat.

Kata kunci: *Web crawler*; algoritma genetik; enjin carian; agen; ketepatan

Abstract. With the tremendous growth of World Wide Web (WWW), finding relevant source through this boundless world becomes a challenging task. In order to make web user easier to seek for their desire information, several famous search engines such as Google, LookSmart, Altavista and Yahoo have been introduced to WWW in these recent years. One of the most crucial components in search engine is web crawler. Web crawler also name as web ant or web robot which uses to crawl all resources or information in the WWW. As the current design of search engines do not have the communication capabilities between the web crawler and the users who dispatched the crawler which cause the imprecise phenomena. Almost the result of finding is outdated or incorrect. Therefore, an intelligent web crawler which namely UtmCrawler has been designed to solve the imprecise phenomena. The methodology of UtmCrawler is consisting of several phases such as literature review, crawling, preprocessing, processing, testing and documentation phase. During the processing phase, genetic algorithm (GA) works as keyword optimization where it expends initial keywords to certain appropriate threshold. The experimental results has shown that a web crawler with GA design has achieved higher precision (95.19%) than the usual crawler which without GA (85.07%). As conclusion, UtmCrawler could provide a better search result for current web user.

Keywords: Web crawler; genetic algorithms; search engine; agent; precision

[1,2&3] Faculty of Computer Science and Information Systems (FSKSM), Universiti Teknologi Malaysia, 81310 UTM Skudai, Johor Bahru, Malaysia
Tel: +607-55 32099/32211/32212 (ext.); Fax: +607-5532210. Email: aselamat@utm.my, yee_way@yahoo.com, echoas1306@yahoo.com

1.0 PENGENALAN

Dengan perkembangan teknologi maklumat yang kian mendadak, dunia semakin menuju era ledakan maklumat. Pelbagai maklumat dari seluruh pelusuk dunia kini hanya di hujung jari dan dapat dicapai dengan mudah, pantas dan terkini. Walau bagaimanapun, bilangan laman web yang dijumpai di internet kini berjumlah satu billion dengan pertambahan sebanyak 1.5 juta laman web setiap hari [1]. Ini menyebabkan proses carian maklumat di internet memakan masa dan menyukarkan [2].

Enjin carian (*search engine*) seperti *Google* [3], *Altavista* [4], *Yahoo!* [5] dan sebagainya telah diperkenalkan dan digunakan bagi memudahkan proses carian maklumat dalam internet. *Web crawler* merupakan komponen yang sangat penting dalam enjin carian [6] telah digunakan sebagai agen untuk mencari dan mengumpul sumber maklumat dalam internet berdasarkan kepada kata kunci tertentu. Namun, proses carian menggunakan *web crawler* menghadapi masalah limpahan [7 – 8] maklumat apabila disimpan dalam pangkalan data [9] enjin carian. Selain itu, keputusan carian yang dijanakan oleh *web crawler* juga kurang tepat [10]. Oleh sebab itu, *UtmCrawler* diperkenalkan dalam kertas kerja ini bagi mengatasi masalah tersebut. Kertas kerja ini dibahagikan kepada beberapa bahagian seperti di bawah.

Bahagian 2 merupakan kajian literatur tentang *web crawler*, teknik algoritma genetik (GA) dan topik yang berkaitan dengannya. Bahagian 3 membincangkan seni bina *UtmCrawler* yang dibangunkan dan bahagian 4 menjelaskan metodologi yang digunakan untuk membangunkan prototaip *UtmCrawler*. Perbincangan dan keputusan pula dibincangkan dalam bahagian 5.

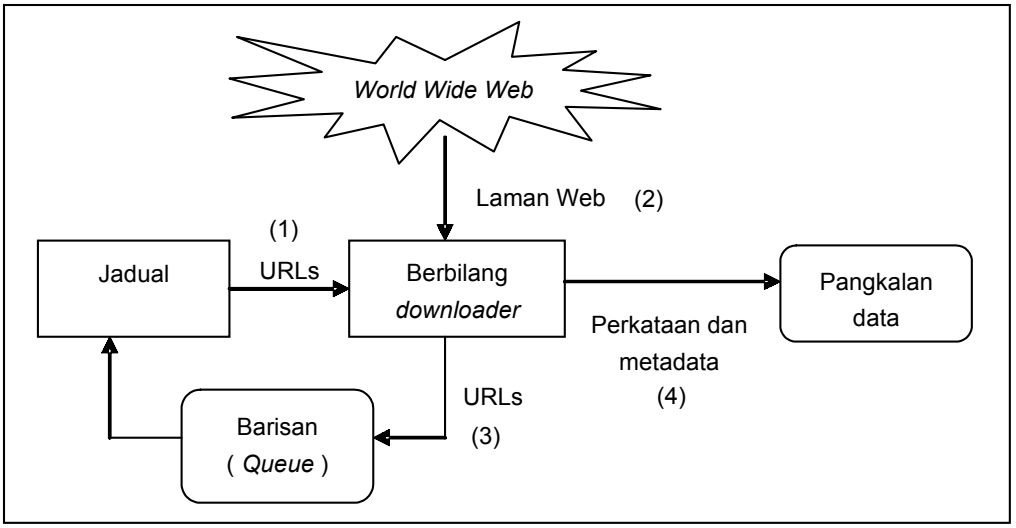
2.0 KAJIAN LITERATUR

2.1 Pengenalan kepada *Web Crawler*

Web crawler dikenali juga sebagai *web spider, web robot* atau *worm* merupakan komponen yang digunakan oleh enjin carian untuk menguruskan dan mengumpul maklumat di pangkalan data sesuatu enjin carian. Seperti dalam Rajah 1, *web crawler* akan menjelajah dari satu pelayan ke pelayan lain untuk mendapatkan sumber maklumat yang diperlukan. Proses carian *web crawler* mengikut turutan (1) hingga (4) seperti ditunjukkan dalam Rajah 1.

Langkah-langkah kerja *web crawler* pada Rajah 1 adalah seperti berikut:

Langkah (1): URL yang dimasukkan oleh pengguna akan disimpan dalam jadual. Satu URL daripada senarai jadual diambil oleh agen untuk menjalankan proses penjelajahan.
Langkah (2): Agen menjalankan proses pencarian dalam WWW.
Langkah (3): Isi kandungan URL diperolehi oleh agen.



Rajah 1 Seni bina *web crawler* berperingkat tinggi [11]

Langkah (4): URL yang baru diteroka oleh *web crawler* akan disimpan ke dalam senarai jadual.

2.2 Kajian Terhadap *Web Crawler* Sedia Ada

Sehingga kini, terdapat pelbagai *web crawler* yang digunakan bagi melaksanakan proses carian. Antaranya, GoogleBot [11] yang merupakan *crawler* bagi enjin carian Google. GoogleBot mempunyai saiz dan skop carian yang terbesar dan meliputi pelbagai format fail seperti *.pdf, *.doc dan *.ps. Carian yang dilaksanakan oleh GoogleBot adalah berdasarkan hubungan antara laman (*sites linkages*) dan jenis keutamaan (*authority*) laman web. Ia mempunyai pangkalan data tambahan seperti kumpulan Google (*Google Groups*), berita (*News*) dan direktori (*Directory*). Namun begitu, GoogleBot juga mempunyai kelemahan dari segi ciri-ciri carian yang terhad seperti tidak bersarang (*no nesting*), tiada pemangkasan (*no truncation*) dan tidak menyokong carian menggunakan teknik Boolean. Selain itu, GoogleBot juga hanya akan mengindekskan 101 KB pertama bagi saiz sesebuah laman web dan 120 KB pertama bagi saiz format fail PDF. Carian secara majmuk atau individu, dan sinonim dengan menggunakan pelbagai tatabahasa juga dijalankan oleh GoogleBot tanpa pengetahuan pengguna.

Netsifter [22], juga merupakan salah satu *crawler* yang dikenali sebagai *crawler* penyimpan laman web atau *web repository crawler*. Ia menyimpan laman web dan datanya kepada pengguna. Kaedah yang digunakan dalam Netsifter berupaya untuk menilai laman web dengan menggunakan gabungan analisis aras laman (*page level*) dan heuristik. Ia juga boleh digunakan untuk menentukan kepopularan sesebuah

laman web dengan memeriksa hubungan luar (*outlink*) pada laman web tersebut. Walau bagaimanapun, pengumpulan laman web adalah secara tidak menyeluruh kerana laman web yang berkualiti rendah akan diabaikan.

Selain GoogleBot dan Netsifter, WebBase [14] juga merupakan penyimpan web yang masih dalam peringkat percubaan (*experimental web repository*) atau *crawler* yang merupakan versi akademik yang dibina di Stanford University. Ia terdiri daripada modul teragih yang membenarkan para penyelidik memperolehi maklumat tertentu sebelum proses penyelidikan. WebBase digunakan oleh penyelidik untuk menyelidik topologi web, menganalisis bahasa kandungan laman web, menganalisis laluan (*link*) teragih berskala besar, menyimpan laman web kerajaan, projek pembelajaran tentang teknologi carian web dan lain-lain. Kelemahan bagi WebBase pula ialah ianya memerlukan ruang cakera yang besar untuk memuatkan data yang besar dan ia hanya akan menjelajah laman web yang mengandungi teks HTML sahaja.

Visual Web Spider [18] ialah *crawler* bagi laman web persendirian (*personal web crawler*) yang mencari maklumat dalam internet. Ia menyimpan laman web berjenis HTML ke dalam fail dan merungkaikan URL yang tidak boleh dijelajah secara langsung. Selain itu, ia juga mempunyai rajah hubungan laluan yang membolehkan pengguna mengikut jejak proses carian dan mempunyai banyak tapisan sebagai penghad sesi. Antara kelemahannya adalah tiada mekanisme pengemaskini maklumat atau isi kandungan laman web secara automatik. Visual Web Spider juga hanya menyimpan ID, URL, tajuk, kata kunci, penerangan dan kandungan teks yang kosong ke dalam pangkalan data atau Microsoft Access.

Web Spider [19] juga merupakan *Personal Web Crawler* yang mencari URL di internet. Ia mempunyai rajah hubungan laluan yang membolehkan pengguna mengikut jejak proses carian dan merupakan *Personal Crawler* yang ringkas dan mudah digunakan oleh pengguna. Ia juga menganalisis bilangan perkataan, laluan dan gambar rajah setiap laman web. Namun begitu, ia tidak menyimpan laman web dalam semua jenis format. Ia mempunyai ciri-ciri dan parameter yang terhad serta tidak mempunyai tapisan carian yang banyak.

Personal Web Crawler lain adalah seperti Win Web Crawler [20] juga turut mencari maklumat di internet dan ianya menyimpan laman web berbentuk HTML ke dalam fail secara automatik serta mempunyai banyak tapisan sebagai penghad sesi. Tetapi, Win Web Crawler tidak mempunyai mekanisme pengemaskini maklumat atau isi kandungan secara automatik serta tidak mempunyai rajah hubungan laluan yang membolehkan pengguna mengikut jejak proses carian.

Yahoo! Slurp [21] atau *crawler* bagi enjin carian Yahoo! mempunyai pangkalan data enjin carian yang besar dan baru serta mempunyai perhubungan dengan Yahoo! *Directory*. Yahoo! Slurp juga menyokong carian Boolean penuh. Tetapi ia kekurangan ciri-ciri carian seperti pemangkasan (*truncation*) dan hanya mengindeks 500 KB pertama sesebuah laman web.

2.3 Strategi *Crawling*

Kecekapan *web crawler* untuk mencari sumber maklumat di internet bergantung kepada penggunaan algoritma carian yang sesuai. Kebanyakan penyelidik menggunakan algoritma carian yang berjenis graf dengan mempertimbangkan laman web sebagai nod (*node*) dan perhubungannya sebagai sisi (*edge*). Secara umumnya, terdapat dua jenis algoritma carian berjenis graf, iaitu carian *uninformed search* dan carian *informed search* [12].

Carian menggunakan teknik *uniformed search* merupakan algoritma yang mudah dan juga dikenali sebagai carian buta kerana ia melakukan carian tanpa melibatkan sebarang maklumat mengenai tentang keadaan semasa untuk bergerak dari satu nod ke satu nod seterusnya. Contoh carian jenis *uniformed search* adalah seperti teknik *breadth-first* dan teknik *depth-first* [12]. Teknik *breadth-first* merupakan algoritma yang biasa digunakan oleh *web crawler* untuk mengumpul semua laman web pada peringkat tertentu sebelum kepada peringkat seterusnya.

Carian berpanduan (*informed search*) merupakan carian yang menyimpan maklumat mengenai setiap sasaran nod yang dicapai seterusnya dan jarak nod terhadap capaian sasaran boleh dijangkakan. Maklumat tersebut seperti jumlah pautan yang terdapat di dalam sesebuah laman web (*number of in-link*), skor kiraan *pagerank*, frekuensi kata kunci dan *query* carian digunakan sebagai heuristik untuk menentukan nod yang mana dicapai dahulu. Contoh carian berpanduan ialah seperti teknik *best-first, hill-climbling* dan A* [2].

Selain daripada *breath-first* dan teknik *best-first*, algoritma carian lain yang biasa digunakan dalam proses *crawling* adalah strategi *crawling* dengan menggunakan teknik *backlink-count* [14], *batch-pagerank* [14], *partial-pagerank* [14] dan *on-line page important computation* (OPIC) [15].

2.3.1 Teknik Breadth-first

Teknik *breadth-first* menggunakan teknik *first-in-first-out* (FIFO), iaitu nod pertama yang dimasukkan akan dikeluarkan dahulu [12]. Kebanyakan *web crawler* menggunakan teknik melebar dahulu untuk proses *crawling* kerana algoritma carian ini bersifat mudah dan tidak kompleks. Semua URL pada aras kedalaman pertama akan dijelajah dahulu sebelum menjelajah kepada aras kedalaman seterusnya. Nilai heuristik tidak digunakan untuk menentukan URL mana yang perlu dijelajah seterusnya.

2.3.2 Teknik Best-first

Teknik *best-first* adalah algoritma carian ruang yang menggunakan heuristik untuk menyusun URL dan menentukan nod yang mana dicapai dahulu [12]. URL yang berkaitan akan disusun pada bahagian depan sesuatu *query*, manakala URL yang

kurang berkaitan akan disusun pada bahagian belakang sesuatu *query* untuk dilayari. Taraf URL ini ditentukan oleh fungsi darjat (*rank function*). URL yang baru dimasukkan ke dalam senarai *query* akan kehilangan peluang untuk dijelajah sekiranya ia mempunyai nilai had tertinggi (*ceiling value*).

2.3.3 Teknik Backlink-count

Teknik *backlink-count* adalah teknik di mana laman web yang mempunyai nilai bilangan hubungan yang terbanyak berhubung dengannya akan dilayari terlebih dahulu [14]. Oleh sebab itu, laman web yang seterusnya dilayari adalah laman web yang paling berkaitan dengan laman web yang telah dimuat turunkan.

2.3.4 Teknik Batch-pagerank

Web crawler akan memuat turun sejumlah *K* laman web dan semua laman web tersebut akan dinilaikan dengan menggunakan kiraan *pagerank*. Sejumlah *K* laman web yang akan dimuat turun seterusnya adalah laman web yang mempunyai nilai *pagerank* yang tertinggi. Nilai *pagerank* ini akan dikira setiap kali terdapat URL yang baru dimuat turun. Teknik *batch-pagerank* adalah lebih baik jika berbanding dengan teknik *backlink-count* [14].

2.3.5 Teknik Partial-pagerank

Teknik *partial-pagerank* adalah sama seperti teknik *batch-pagerank*, tetapi semasa proses mengira semula nilai *pagerank*, satu nilai *pagerank* sementara disetkan kepada laman web yang baru dimuat turun [14]. Nilai *pagerank* sementara ini adalah sama dengan jumlah tahap pernormalan (*normalized rankings*) sesuatu laman web yang berhubung dengannya. Dengan itu, laman web yang terkini dijumpai dapat dijelajah oleh *web crawler* dengan secepat mungkin. Laman web yang dimuat turun adalah laman web yang mempunyai nilai *partial pagerank* yang tertinggi.

2.3.6 Teknik On-line Page Important Computation (OPIC)

Dalam teknik OPIC, semua laman web permulaan mempunyai nilai *cash* yang sama. Setiap kali laman web tertentu dijelajah oleh *web crawler*, nilai *cash*nya akan dibahagikan kepada kalangan laman web yang berhubung dengannya [15]. Laman web yang belum dijelajahi oleh *web crawler* akan mempunyai jumlah nilai *cash* daripada laman web yang berhubung dengannya. Strategi ini hampir sama dengan *pagerank*, tetapi pengiraannya tidak diulangi. Oleh sebab itu, teknik yang menggunakan strategi ini adalah lebih pantas.

2.4 Masalah *Web Crawler*

Kebanyakan *web crawler* hanya menggunakan bahagian tajuk laman web sahaja untuk proses pengindeksan. Bahagian tajuk adalah bahagian yang ditulis atau tidak ditulis oleh pengaturcara web dalam dokumen tersebut. Sebanyak 20% dokumen yang dijelajah oleh *web crawler* tertentu tidak mempunyai bahagian tajuk dalam dokumen tersebut [10]. Proses pengindeksan yang menggunakan tajuk dokumen tersebut akan menyebabkan bahagian penting seperti isi kandungan dokumen tidak diambil kira semasa proses tersebut dilakukan. Tambahan pula, maklumat pada bahagian tajuk tidak dapat mengambarkan isi kandungan dokumen dengan sepenuhnya. Oleh sebab itu, keputusan carian yang dijanakan oleh *web crawler* adalah kurang tepat.

Penyelesaian bagi masalah ini adalah dengan menjalankan proses pengindeksan menggunakan maklumat pada bahagian tajuk dan isi kandungan dokumen tersebut (*full text indexing*). Proses pengindeksan yang menggunakan tajuk dokumen dan isi kandungan dokumen dapat menggambarkan dokumen tersebut dengan lebih menyeluruh. Dengan itu, *web crawler* dapat membekalkan maklumat yang lebih tepat kepada pengguna. Selain itu, penggunaan teknik GA juga boleh digunakan untuk menjanakan keputusan carian yang lebih tepat. GA akan mengembangkan kata kunci yang dimasukkan oleh pengguna supaya hasil carian mempunyai nilai ketepatan yang tinggi [16].

Proses carian sumber maklumat bermula dengan URL permulaan yang dimasukkan oleh pengguna. *Web crawler* akan meneroka semua laman web yang berhubung dengan laman web permulaan. Kemudian, bilangan *web crawler* akan bertambah berdasarkan bilangan laman web yang baru diteroka dan seterusnya. *Web crawler* akan menjelajah kesemua laman web pada setiap kedalaman. Oleh sebab itu, bilangan *web crawler* yang menjalani fungsi carian semakin bertambah banyak. Maklumat yang dikumpul dan diperolehi oleh *web crawler* juga akan bertambah dan seterusnya disimpan ke dalam pangkalan data komputer pengguna [9]. Tetapi hal ini akan membebankan pangkalan data komputer pengguna.

Salah satu cara untuk menyelesaikan masalah limpahan maklumat adalah penggunaan *robots exclusion protocol*. *Robots exclusion protocol* yang juga dikenali sebagai *robots.txt protocol* boleh digunakan oleh pentadbir web untuk menentukan bahagian-bahagian pelayan (*web server*) yang menyimpan laman web mana yang tidak akan dicapai oleh *web crawler*. *Robots.txt protocol* juga boleh digunakan oleh pentadbir web untuk menentukan bahagian-bahagian dokumen yang berguna supaya dapat dicapai oleh *web crawler*. Fungsi ini dapat mengelakkan *web crawler* daripada memperoleh maklumat yang tidak berguna demi menjamin kualiti proses pengindeksan.

Sela masa permintaan antara pengguna dan sistem juga pelu dipertimbangkan supaya proses carian dapat dijalankan dengan lebih lancar. Sela masa tersebut perlu

dilaksanakan dalam satu tempoh masa menunggu yang singkat semasa proses *crawling* dilakukan. Contohnya, kajian Cho, J. *et al.* [7] telah meletakkan jumlah tetap 10 saat sebagai sela masa yang perlu ada pada *web crawler* bagi mencapai dokumen tertentu. Selain itu, cara pengumpulan maklumat yang sistematik juga diperlukan. Maklumat yang diperolehi oleh *web crawler* pada aras tertentu akan disimpan dalam satu fail. Kemudian, fail-fail pada aras tersebut akan dikumpul dan disimpan ke dalam satu fail dan seterusnya. Dengan demikian, hasil hantaran akhir *web crawler* kepada pangkalan data hanya satu fail yang bermaklumat sistematik demi mengatasi masalah limpahan dalam penyimpanan maklumat ke dalam pangkalan data.

UtmCrawler yang dibangunkan dengan menggunakan teknik algoritma genetik (GA) untuk mengatasi masalah hasil carian yang kurang tepat. Satu cara pengumpulan maklumat yang sistematik dan berstruktur perlu digunakan untuk mengatasi masalah limpahan dalam penyimpanan maklumat ke dalam pangkalan data.

2.5 Pengenalan kepada Algoritma Genetik

Algoritma Genetik (GA) adalah teknik pencarian tegap yang terbukti sebagai satu kaedah yang baik untuk menyelesaikan masalah pengoptimuman yang kompleks. Ia mula diperkenalkan oleh Holland pada tahun 1975 [4]. Nama algoritma genetik berasal dari analogi di antara struktur kompleks melalui vektor komponen dan struktur genetik pada kromosom.

Prinsip utama dalam mengambil GA adalah ciri-ciri penurunan dan evolusi dalam genetik kromosom. Dalam proses GA, setiap individu ditakrifkan sebagai berpotensi untuk menjadi salah satu cara penyelesaian masalah. Kualiti sesuatu individu dinilaikan dengan menggunakan satu fungsi yang dikenali sebagai *fitness function*. Setiap generasi akan menghasilkan generasi yang baru melalui proses pemilihan, proses penyilangan dan proses mutasi [12]. Setiap generasi baru yang dihasilkan merupakan individu yang lebih berkualiti berbanding dengan generasi sebelumnya.

Di samping itu, GA juga digunakan untuk mencari cara penyelesaian yang paling optimum bagi masalah tertentu. Contohnya, penggunaan GA dalam mencari laluan terpendek perjalanan bas, penggunaan teknik GA untuk menyelesaikan masalah penjadualan dan lain-lain [13].

2.6 Fitness Function

Fitness function adalah satu fungsi GA yang digunakan untuk menentukan pelaksanaan atau *fitness* setiap individu dalam populasi. Individu yang mempunyai nilai *fitness* yang tinggi akan mempunyai kebarangkalian tinggi dipilih untuk proses pemilihan. Contoh-contoh *fitness function* adalah seperti pekali *Jaccard, cosine similarity* dan *Haming distance* [4].

$$fitness\left(d_j\right) = \frac{1}{n} \bullet \sum_{k=1}^{n} \left[\frac{\left| d_j \cap d_q \right|}{\left| d_j \cup d_q \right|} \right] \quad (1)$$

dengan *dj* adalah dokumen *j*, *n* adalah bilangan kata kunci yang terdapat dalam dokumen *j*, *k* adalah satu perwakilan nombor dan d_q adalah dokumen *q*.

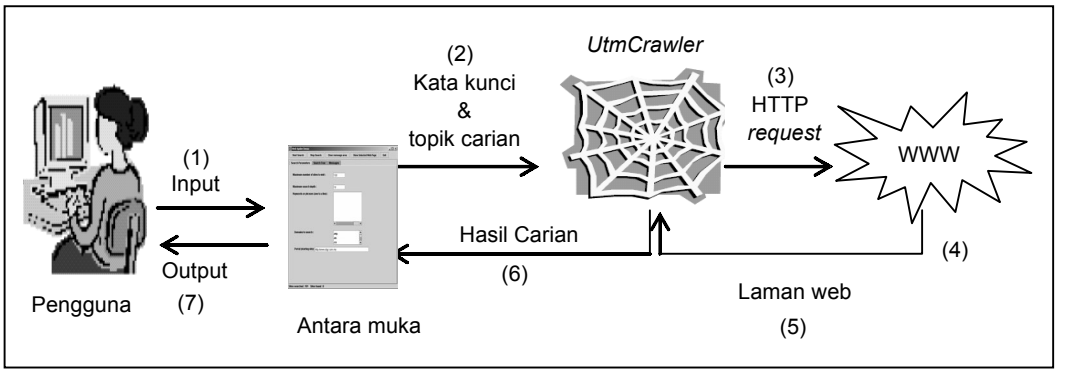
Cosine similarity adalah satu ukuran yang mengukur kesamaan dua vektor yang berdimensi *n* dengan mencari sudut di antaranya. Diberi dua vektor A dan B, *Cosine similarity* (θ) dikira seperti yang ditunjukkan dalam rumus (2) dengan menggunakan hasil *dot product* dan *magnitude* seperti di bawah:

$$\theta = \arccos \frac{A * B}{|A| * |B|} \quad (2)$$

Hamming distance antara dua string yang mempunyai kepanjangan yang sama adalah jumlah perbezaan simbol antara dua string tersebut. Contohnya, *Hamming distance* antara 1011101 dan 1001001 adalah 2, *Hamming distance* antara 2143896 dan 2233796 adalah 3.

3.0 SENI BINA UTMCRAWLER

Seperti yang ditunjukkan pada Rajah 2, pengguna akan memasukkan input kepada *UtmCrawler* melalui antara muka yang dibangunkan dengan menggunakan GUI dan Java. Kemudian, *UtmCrawler* akan mula menjalankan proses carian dengan menghantar *HTTP* request kepada WWW yang dinamik. Laman web yang diperoleh oleh *UtmCrawler* akan diproses dan disimpan dalam pangkalan data pengguna. Hasil keputusan yang mempunyai nilai ketepatan yang tinggi akan dihantar dan dipaparkan kepada pengguna melalui antara muka *UtmCrawler*. Rajah 2 menunjukkan seni bina dan proses yang dilakukan oleh *UtmCrawler* secara teliti



Rajah 2 Gambar rajah peringkat tinggi *UtmCrawler*

dari proses (1) hingga proses (7) bagi mencapai isi kandungan laman web yang dikehendaki oleh pengguna.

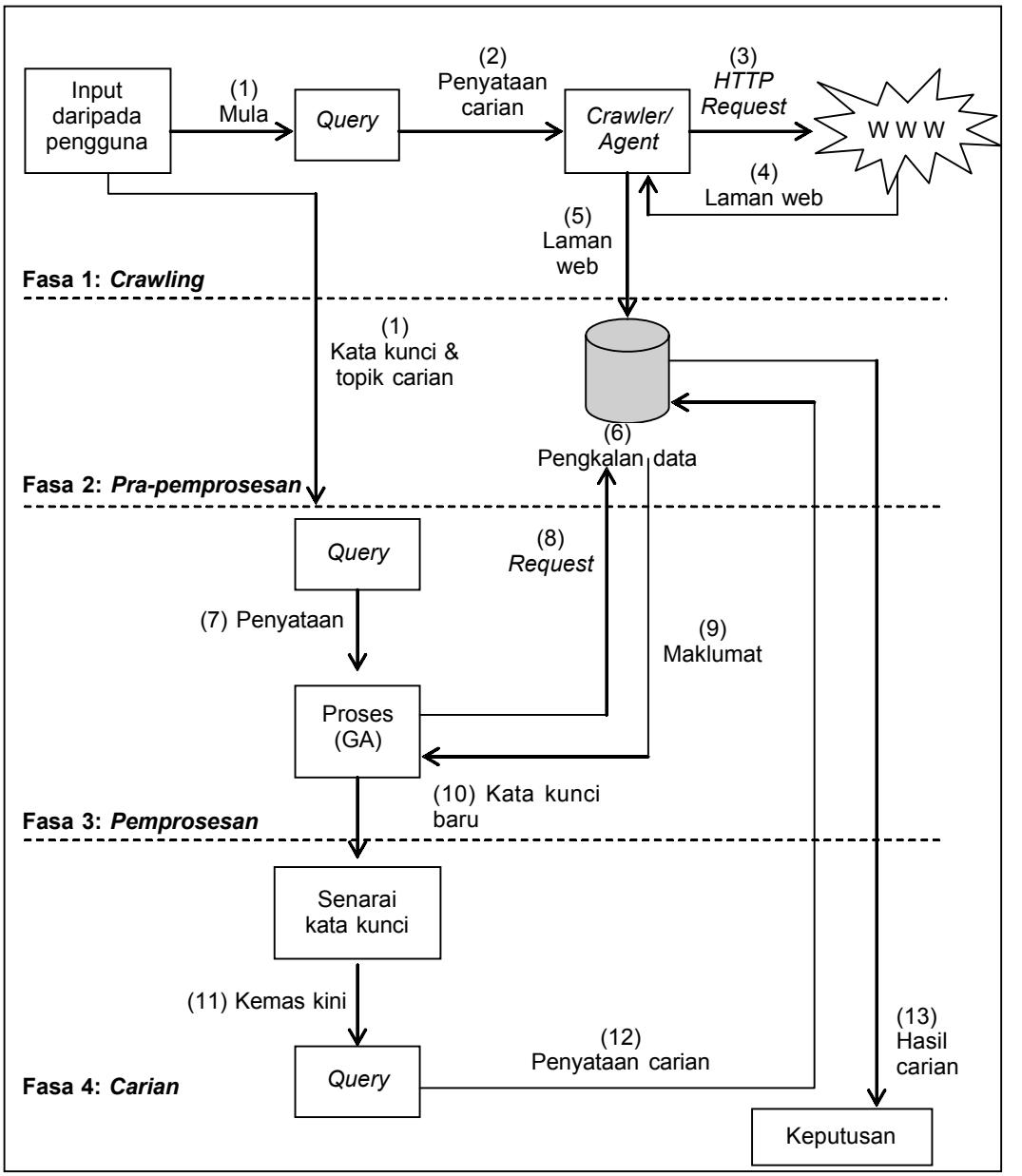
Rajah 2 menunjukkan seni bina *UtmCrawler* pada peringkat tinggi dan aliran kerjanya adalah seperti berikut:

Langkah (1): Pengguna memasukkan input kepada *UtmCrawler* melalui antara muka *UtmCrawler*.
Langkah (2): Kata kunci dan topik carian dihantar kepada *UtmCrawler*.
Langkah (3): *UtmCrawler* menghantar *HTTP request* kepada web.
Langkah (4): Proses carian dalam WWW.
Langkah (5): Laman web yang diteroka dihantar balik kepada *UtmCrawler*.
Langkah (6): Hasil carian dihantar kepada antara muka *UtmCrawler* selepas diproses oleh *UtmCrawler*.
Langkah (7): Hasil carian dipaparkan sebagai output kepada pengguna.

Setiap proses yang dirujuk dalam Rajah 2 dapat dijelaskan secara terperinci dalam Rajah 3 dengan fasa *crawling* adalah proses pencarian maklumat oleh agen dalam WWW. Fasa pra-pemprosesan (*preprocessing*) adalah penyusunan laman web mengikut nombor rujukan dan penyimpanan maklumat laman web dengan sistematik (rujuk nombor (1) hingga (13) dalam Rajah 3 untuk melihat proses yang terlibat). Proses GA dilaksanakan pada fasa pemprosesan (*processing*) dan hasil modul GA adalah senarai kata kunci yang baru dan berkaitan dengan kata kunci permulaan. Pada fasa carian (*searching*), senarai kata kunci baru akan mengemaskinikan penyataan carian dan pencarian maklumat dalam pangkalan data. Hasil carian dipaparkan sebagai output kepada pengguna.

Dengan merujuk kepada Rajah 3, langkah-langkah yang terlibat dalam proses carian *UtmCrawler* adalah seperti berikut:

Langkah (1): Pengguna memulakan proses carian. Kata kunci serta topik carian dihantar ke dalam modul *query*.
Langkah (2): Modul *query* akan menghantar penyataan carian untuk mencari semua maklumat dalam internet.
Langkah (3): *UtmCrawler* menghantar *HTTP request* kepada web.
Langkah (4): Proses carian dalam WWW dan laman web yang diteroka dihantar balik kepada *UtmCrawler*.
Langkah (5): Laman web yang dijelajahi oleh agen disimpan dalam pangkalan data secara sistematik.
Langkah (6): Proses penyimpanan maklumat laman web dengan sistematik dilaksanakan.
Langkah (7): Modul *query* akan menghantar penyataan carian (kata kunci) untuk menjalankan proses carian dalam pangkalan data.



Rajah 3 Gambar rajah seni bina *UtmCrawler*

Langkah (8): Hasil carian dipaparkan kepada pengguna.
Langkah (9): Senarai URL yang dipilih oleh pengguna dihantar ke Modul GA.
Langkah (10): Modul GA menghantar *request* kepada pangkalan data untuk mendapatkan maklumat URL yang dipilih oleh pengguna sebagai input kepada proses GA.

Langkah (11): Pangkalan data menghantar maklumat tersebut kepada modul GA.
Langkah (12): Proses GA dilaksanakan dan hasil modul GA adalah satu kata kunci yang baru dan berkaitan dengan kata kunci permulaan.
Langkah (13): Modul *query* dikemaskinikan, iaitu penambahan kata kunci baru terhadap kata kunci lama. Kemudian, modul *query* menghantar penyataan carian (kata kunci + kata kunci baru) untuk menjalankan proses carian dalam pangkalan data.
Langkah (14): Hasil carian dipaparkan kepada pengguna.

4.0 METODOLOGI

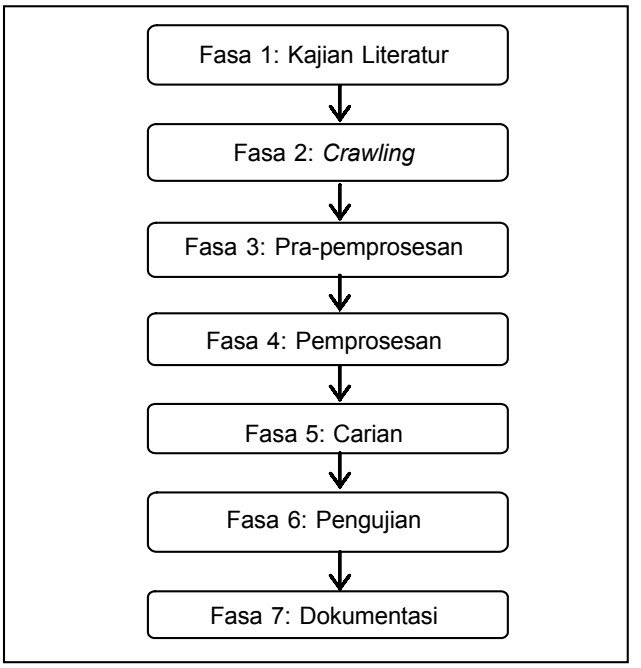
Rajah 4 menunjukkan metodologi yang digunakan dalam pembangunan *UtmCrawler*. Ia terdiri daripada tujuh fasa utama, iaitu fasa kajian literatur, fasa *crawling*, fasa pra-pemprosesan, fasa pemprosesan, fasa carian, fasa dokumentasi dan fasa pengujian.

4.1 Fasa Kajian Literatur

Fasa ini melibatkan tiga tugas utama. Tugas pertama ialah merancang projek awalan. Matlamat projek, objektif projek, skop projek, latar belakang masalah dan kepentingan projek dikaji dan dikenal pasti. Tugas kedua pula adalah mengkaji dan menganalisis topik yang berkaitan dengan *web crawler* dengan teliti. Kajian tersebut merangkumi penelitian terhadap *web crawler*, strategi *crawling*, teknik algoritma genetik (GA) dan topik yang berkaitan dengannya. Tugas ketiga adalah mereka bentuk *UtmCrawler*, iaitu mereka bentuk seni bina *UtmCrawler*, mereka bentuk antara muka yang menghubungkan *UtmCrawler* dengan pengguna, mereka bentuk aliran proses dan mereka bentuk pangkalan data.

4.2 Fasa *Crawling*

Fasa *crawling* yang dilaksanakan oleh *UtmCrawler* digunakan untuk mencari semua maklumat di internet. Teknik *breadth-first* dipilih sebagai strategi *crawling* kerana teknik tersebut bersifat mudah dan menjelajah semua laman web di internet. *UtmCrawler* akan menjelajah dari satu pelayan ke pelayan yang lain untuk mendapatkan sumber maklumat yang diperlukan. Rajah 5 menunjukkan aliran kerja modul *crawling* di mana *fetcher* dan *parser* adalah sejenis agen yang mempunyai fungsi tertentu. Rajah 6 menunjukkan antara muka bagi proses *crawling* di mana pengguna perlu memasukkan URL untuk memulakan proses tersebut. Setelah proses *crawling* tamat, proses pra-pemprosesan akan bermula. Manakala Rajah 7 pula menunjukkan contoh hasil keputusan daripada proses *crawling*.



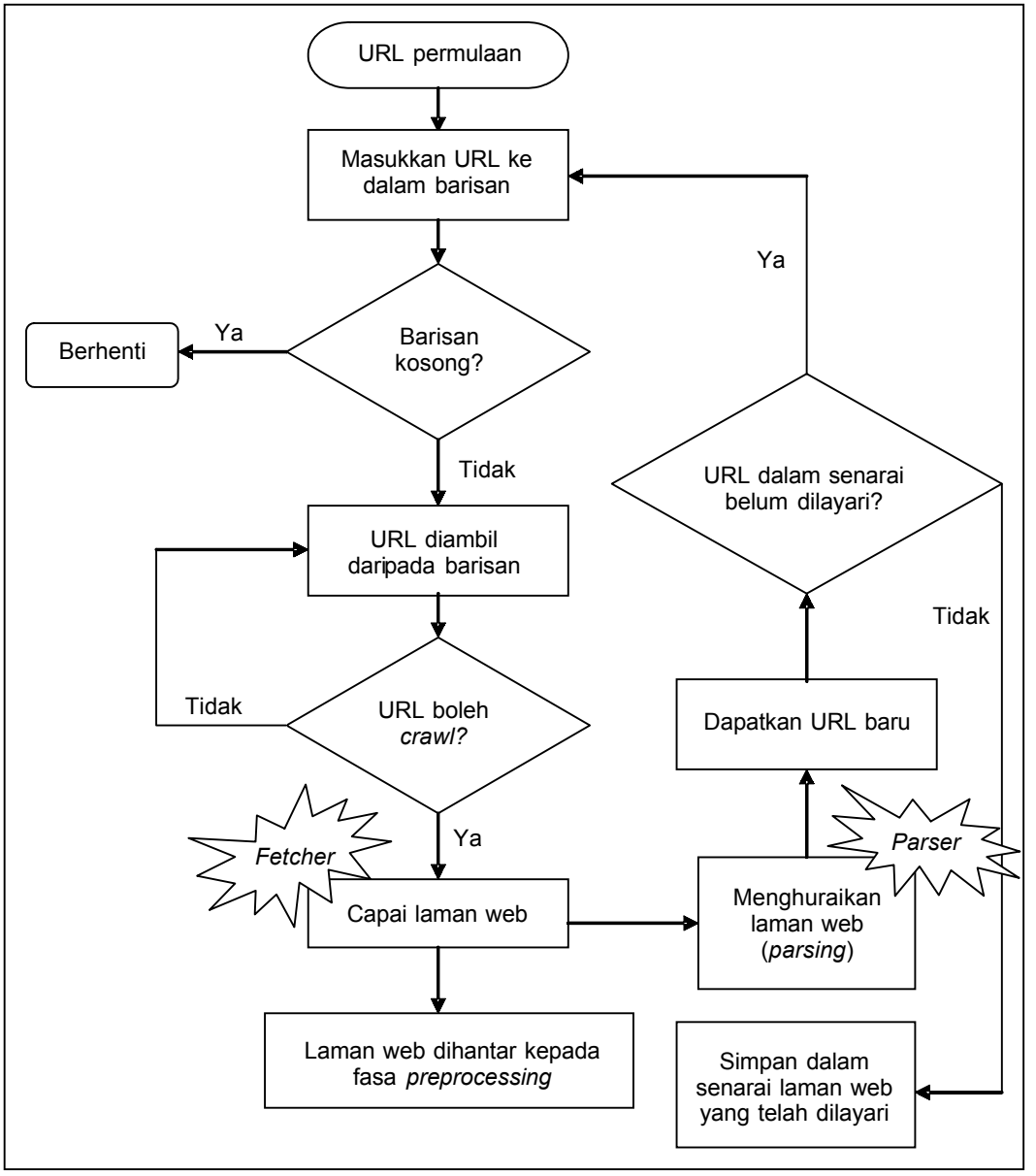
Rajah 4 Metodologi pembangunan *UtmCrawler*

4.3 Fasa Pra-pemprosesan

Dalam fasa pra-pemprosesan, proses yang terlibat adalah proses menyusun laman web mengikut Idnya dan proses penyimpanan maklumat laman web dengan sistematik. Satu laman web mempunyai nombor rujukan (ID) yang unik, isi kandungan yang sama tetapi boleh memiliki nama URL yang berbeza. Dalam fasa ini, laman web disusun berdasarkan Idnya supaya isi kandungan URL dan ID yang sama tidak berulangan dalam senarai dan pangkalan data. Proses penyimpanan maklumat carian ke dalam pangkalan data secara sistematik dapat mengurangkan beban pelayan dan pangkalan data. Ini memudahkan proses penjanaan rajah pokok carian. Rajah 8 menunjukkan aliran kerja modul pra-pemprosesan.

4.4 Fasa Pemprosesan

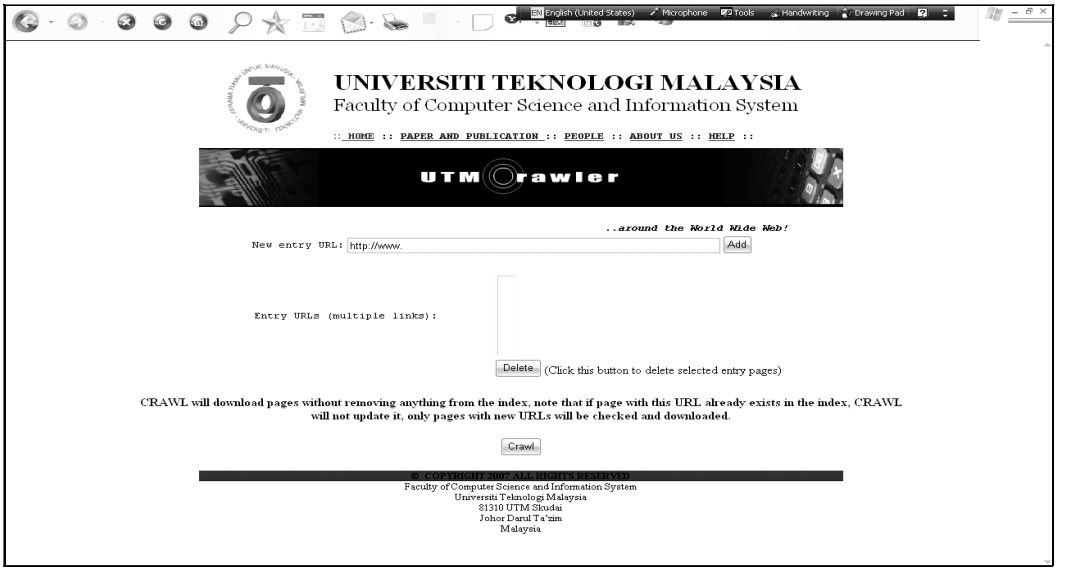
Dalam fasa pemprosesan, teknik GA digunakan untuk mengembangkan kata kunci carian permulaan. Contohnya, satu kata kunci baru, iaitu *British* akan ditambahkan kepada kata kunci permulaan yang terdiri daripada *Iraq, Iraqi, US* dan *War* selepas melalui proses GA. Ini kerana banyak laman web yang mencatatkan berita pasukan tentera *British* dalam negara *Iraq*. Proses perkembangan kata kunci carian permulaan ini menjana keputusan carian yang mempunyai nilai ketepatan yang tinggi. Nilai ketepatan digunakan untuk mengukur darjah ketepatan dokumen yang diperoleh



Rajah 5 Aliran kerja proses *UtmCrawler*

oleh *UtmCrawler* adalah sama dengan *query* carian. Rajah 9 menunjukkan aliran kerja modul pemprosesan.

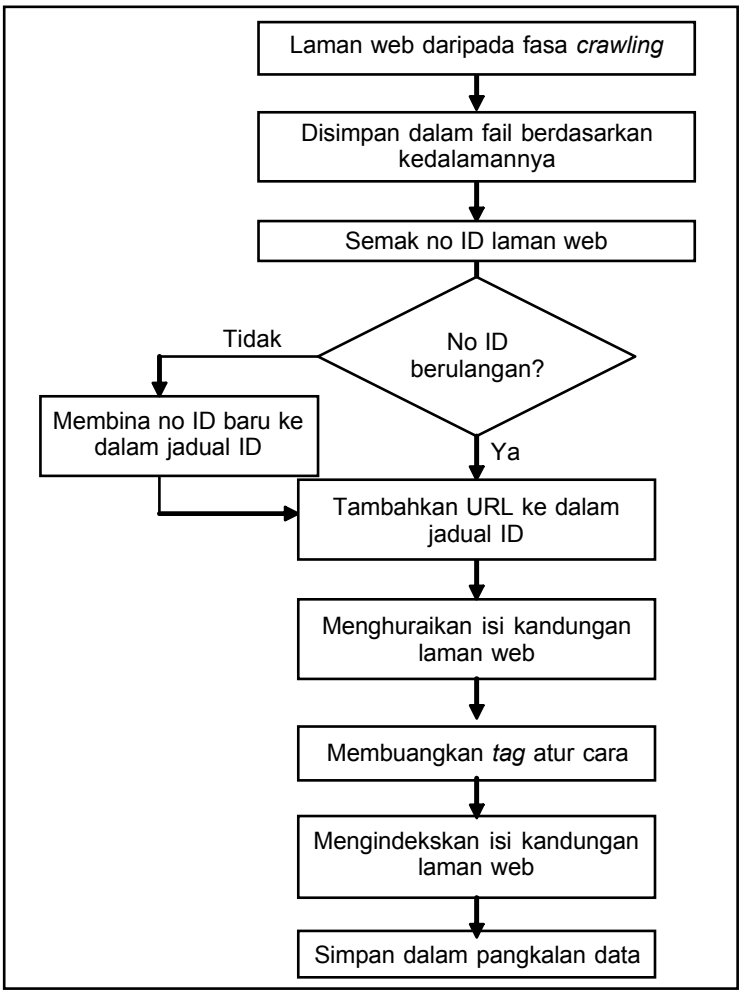
Contohnya, katakan kata kunci permulaan adalah *information* dan *retrieval*. Permintaan bagi kata kunci tersebut akan dihantar kepada pangkalan data supaya memperoleh dokumen-dokumen untuk diproses dan proses pengiraan nilai pemberat terhadap semua dokumen-dokumen dapat dilaksanakan. Lima dokumen yang



Rajah 6 Antara muka *UTMCrawler* yang dibangunkan

No	Id	Ranking	Title	Url	Selected
22	255.txt	21.50%	Universiti Teknologi Malaysia	http://web.utm.my/today/index.php?option=co...	✔
23	90.txt	21.14%	Universiti Teknologi Malaysia	http://web.utm.my/about/index.php?option=co...	✔
24	244.txt	21.09%	Universiti Teknologi Malaysia	http://web.utm.my/today/index.php?option=co...	✔
25	237.txt	20.44%	Campus Maps	http://web.utm.my/campus_map/	✔
26	56.txt	19.88%	Perpustakaan Sultanah Zana...	http://web.utm.my/psz	✔
27	50.txt	19.87%	Universiti Teknologi Malaysia	http://web.utm.my/about/index.php?option=co...	✔
28	257.txt	19.70%	Universiti Teknologi Malaysia	http://web.utm.my/today/index.php?option=co...	✔
29	245.txt	19.60%	Universiti Teknologi Malaysia	http://web.utm.my/today/index.php?option=co...	✔
30	248.txt	18.56%	Universiti Teknologi Malaysia	http://web.utm.my/about/index.php?option=co...	✔
31	236.txt	17.98%	Universiti Teknologi Malaysia ...	http://www.utm.my/	✔
32	51.txt	16.62%	Admission	http://web.utm.my/admission	✔
33	240.txt	15.57%	Universiti Teknologi Malaysia	http://web.utm.my/about	✔
34	274.txt	14.73%	Universiti Teknologi Malaysia	http://web.utm.my/today/index.php?option=co...	✔
35	52.txt	14.32%	Office of Student Affair	http://web.utm.my/hep	✔
36	270.txt	13.62%	Office of Deputy Vice Chancell...	http://web.utm.my/tncap/	✔
37	53.txt	12.96%	Office of International Affairs	http://web.utm.my/oia	✔
38	247.txt	12.44%	Office of Deputy Vice Chancell...	http://web.utm.my/tncap	✔
39	249.txt	12.08%	Universiti Teknologi Malaysia	http://web.utm.my/about/index.php?option=co...	✔
40	282.txt	11.87%	Students Representative Cou...	http://web.utm.my/mpp	✔
41	94.txt	11.42%	Students Representative Cou...	http://web.utm.my/mpp/	✔
42	250.txt	10.92%	Calendar of Events | FKSG W...	http://www.fksg.utm.my/	✔
43	251.txt	10.44%	Office of Deputy Vice Chancell...	http://web.utm.my/tncap/index.php?option=co...	✔
44	95.txt	9.75%	Skudai Post Online :: Universi...	http://web.utm.my/skpost/	✔
45	54.txt	9.61%	Centre for Teaching & Learni...	http://www.ctl.utm.my	✔
46	259.txt	9.28%	Welcome to FKE	http://www.fke.utm.my/	✔
47	165.txt	9.25%	Welcome to Universiti Malaysi...	http://www.ums.edu.my/usefullink.php	
48	261.txt	7.97%	Main Page - Faculty of Educati...	http://www.fp.utm.my/	✔
49	241.txt	7.84%	Unit Perhubungan Alumni	http://alumni.utm.my/alumni3/index.jsp	✔
50	105.txt	7.35%	UTMonline - Intranet UTM	http://utmonline.utm.my	✔

Rajah 7 Keputusan yang diperoleh oleh *UTMCrawler*

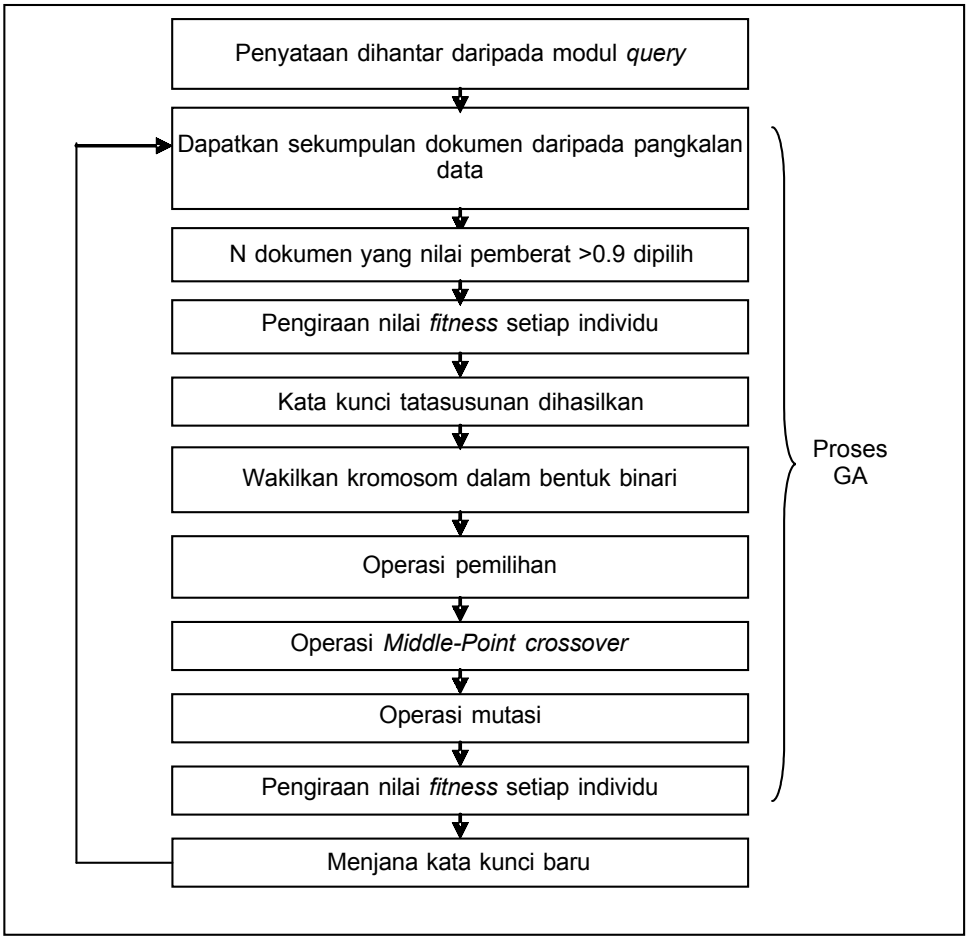


Rajah 8 Aliran kerja fasa pra-pemprosesan

mempunyai nilai pemberat lebih daripada 0.9 akan dipilih dan kata kunci bagi setiap dokumen disusun atur dalam bentuk tatasusunan seperti yang ditunjukkan pada Rajah 10. Pengiraan pemberat adalah berdasarkan skema pemberat *Lnu.Itu* [17]. Contoh isi kandungan lima dokumen adalah seperti Jadual 2.

Jadual 2 Isi kandungan lima dokumen dan kata kunci masing-masing

Dokumen	**Isi kandungan laman web (dalam bentuk *TF*)**	**Kata kunci**
Doc 1	*Data – 8, Retrieval-4, Information-2, Queries-1, Database-2*	*Data*
Doc 2	*Data – 2, Retrieval-2, Information-4, Computer-1*	*Information*
Doc 3	*Indexing-3, System-1, Retrieval-2, Information-5, IR-6*	*IR*
Doc 4	*Query-9, Information-3, Data-5*	*Query*
Doc 5	*Retrieval-5, Information-4, Data-2*	*Retrieval*



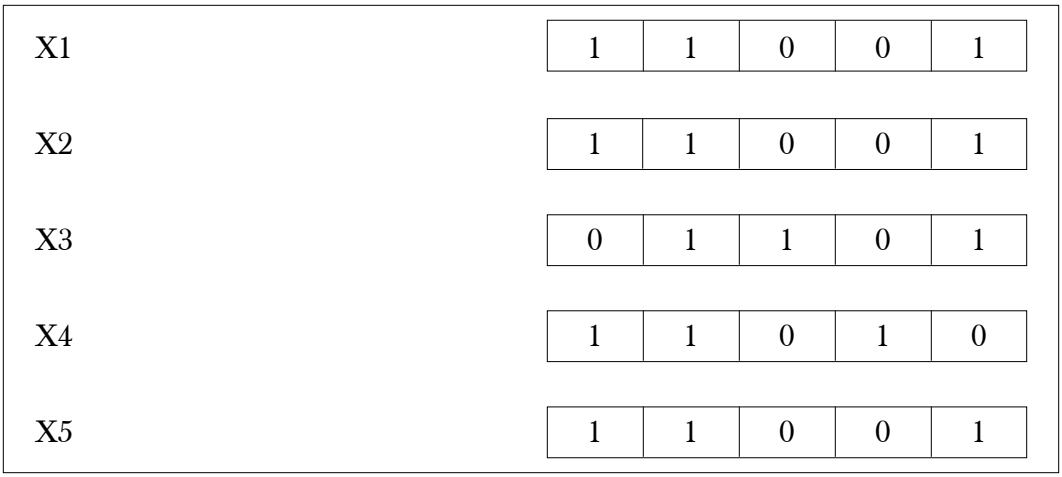
Rajah 9 Aliran kerja fasa pemprosesan

Data	*Information*	*IR*	*Query*	*Retrieval*

Rajah 10 Kata kunci dalam tatasusunan

Nilai binari tatasusunan bersaiz 5 diberi kepada setiap individu. Nilai gen disetkan kepada 1 sekiranya individu tersebut mempunyai kata kunci (*term*) yang sama dengan kata kunci tatasusunan dan sebaliknya. Rajah 11 pula menunjukkan perwakilan 5 dokumen tersebut dalam bentuk kromosom.

Nilai *fitness* bagi setiap individu dalam populasi akan dikira. *Fitness function* yang digunakan adalah daripada pengiraan Jaccard seperti yang ditunjukkan pada persamaan (1). Nilai *fitness* setiap individu seperti dalam Jadual 3.



Rajah 11 Perwakilan dokumen dalam kromosom

Jadual 3 Nilai *fitness* bagi populasi awalan

Dokumen	Kromosom	Nilai fitness	Fitness/total fitness *100%
X1	11001	0.8	22.79%
X2	11001	0.75	21.37%
X3	01101	0.54	15.38%
X4	11010	0.52	14.81%
X5	11001	0.9	25.64%

Kromosom atau individu yang mempunyai nilai fitness yang tinggi akan dipilih dalam operasi pemilihan. Dua individu yang mempunyai nilai *fitness* yang tinggi dipilih untuk melaksanakan proses penyilangan. Teknik penyilangan yang digunakan adalah *middle-point crossover*. Operasi mutasi pula dilaksanakan setelah operasi penyilangan selesai. Kemudian, nilai *fitness* untuk setiap individu dalam populasi baru akan dikira. Jadual 4 menunjukkan nilai fitness untuk populasi baru.

Jadual 4 Nilai *fitness* populasi baru

Dokumen	Kromosom	Nilai fitness
X1	11001	0.95
X2	11001	0.95
X3	11001	0.95
X4	11001	0.95
X5	11011	0.8

4.5 Fasa Carian

Kata kunci baru yang dihasilkan daripada fasa pemprosesan digunakan untuk mengemas kini penyataan carian. Permintaan bagi kata kunci dihantar ke pangkalan data untuk memulakan proses carian. Proses carian dalam pangkalan data adalah berdasarkan penggabungan kata kunci permulaan dengan kata kunci baru. Hasil carian akan dihantar dan dipaparkan kepada pengguna melalui antara muka *UtmCrawler*.

4.6 Fasa Pengujian

Penanda aras yang dijalankan adalah berdasarkan *fitness function* yang digunakan dalam proses GA pada fasa pemprosesan. *Fitness function* yang terpilih adalah *Jaccard Coefficient* seperti dalam formula (1), *Cosine similarity* seperti dalam formula (2) dan *Haming distance*. Contoh *Hamming distance* adalah perbezaan antara 1011101 dan 1001001 adalah 2, *Hamming distanc*e antara 2143896 dan 2233796 adalah 3.

Keputusan perbandingan akan dinilai berdasarkan nilai ketepatan (*precision*), nilai carian relevan yang diperolehi (*recall*), dan nilai ketepatan (*F1*). Nilai ketepatan, *recall* dan F1 dinyatakan di bawah dengan nilai *a*, *b* dan *c* ditakrif pada Jadual 5 dan Jadual 6.

$$precision = \frac{a}{a+b} \tag{3}$$

$$recall = \frac{a}{a+c} \tag{4}$$

$$F1 = \frac{2}{\frac{1}{precision} + \frac{1}{recall}} \tag{5}$$

Jadual 5 Takrifan pemboleh ubah *a,b, c* dan *d*

Nilai	Penerangan
a	Bilangan laman web yang dipersetujui oleh *UtmCrawler* dan pengguna
b	Bilangan laman web yang tidak dipersetujui oleh *UtmCrawler* tetapi dipersetujui oleh pengguna
c	Bilangan laman web yang dipersetujui oleh *UtmCrawler* tetapi tidak dipersetujui oleh pengguna
d	Bilangan laman web yang tidak dipersetujui oleh *UtmCrawler* dan pengguna

Jadual 6 Hubungan antara *a, b, c* dan *d*

	Sistem (setuju)	**Sistem (tidak setuju)**
Pengguna (setuju)	*a*	*b*
Pengguna (tidak setuju)	*c*	*d*

5.0 HASIL PENGUJIAN

Daripada Jadual 4, X2 yang mempunyai nilai *fitness* yang paling tinggi dipilih untuk mengembangkan kata kunci permulaan. Kata kunci X2 mengandungi "*Data,*" "*Information*" dan "*Retrieval.*" Oleh sebab kata kunci *Information* dan *Retrieval* adalah kata kunci permulaan, maka kata kunci "*Data*" dipilih sebagai kata kunci baru pada generasi pertama proses GA dan proses GA diulang. Purata nilai *fitness* bertambah daripada 0.702 (Jadual 3) kepada 0.920 (Jadual 4).

Jadual 3 Nilai *fitness* bagi populasi awalan

Dokumen	**Kromosom**	**Nilai fitness**
X1	11001	0.8
X2	11001	0.75
X3	01101	0.54
X4	11010	0.52
X5	11001	0.90
Purata		**0.702**

Jadual 4 Nilai *fitness* populasi baru

Dokumen	**Kromosom**	**Nilai fitness**
X1	11001	0.95
X2	11001	0.95
X3	11001	0.95
X4	11001	0.95
X5	11011	0.80
Purata		**0.920**

5.1 PERBINCANGAN HASIL KEPUTUSAN

Keputusan carian dengan menggunakan teknik GA dan keputusan carian tanpa menggunakan teknik GA akan dibandingkan berdasarkan nilai ketepatan (*precision*), nilai carian relevan yang diperolehi (*recall*) dan F1. Seperti yang dinyatakan dalam skop projek, dataset projek ini terdiri daripada 3 bidang, iaitu bidang akademik, bank dan telekomunikasi. Jadual 7 menunjukkan bilangan laman web yang disimpan

Jadual 7 Jumlah laman web yang terdapat dalam bidang masing-masing

Set	Bidang	Sub-bidang	Bilangan laman web
1	Akademik	Universiti Teknologi Malaysia	72
		Universiti Malaya	45
		Universiti Malaysia Sarawak	21
		Universiti Malaysia Sabah	18
		Universiti Sains Malaysia	60
		Lain-lain	13
2	Bank	AmBank group	24
		Bank Rakyat	33
		OCBC Bank	26
		Associations of Bank	29
		Lain-lain	18
3	Telekomunikasi	Digi	19
		Celcom	10
		Maxis	14
		Lain-lain	5

dalam pangkalan data mengikut bidang. Setiap bidang pula mempunyai sub-bidang masing-masing. Contohnya, Universiti Teknologi Malaysia, Universiti Malaya, Universiti Malaysia Sarawak dan Universiti Malaysia Sabah adalah sub-bidang dalam bidang akademik. Keputusan pengujian boleh diringkaskan seperti dalam Jadual 8 dan Rajah 11. Apabila kata kunci di atas ditambah kepada kata kunci carian permulaan, keputusan carian yang dijana akan mempunyai nilai ketepatan yang lebih tinggi berbanding dengan keputusan carian menggunakan kata kunci carian permulaan. Ciri-ciri *UtmCrawler* yang dibangunkan pula adalah seperti yang ditunjukkan pada Jadual 9.

Daripada Rajah 12, keputusan carian yang dijanakan dengan menggunakan GA mempunyai nilai ketepatan yang lebih tinggi, iaitu 95.19% berbanding dengan keputusan carian yang tanpa menggunakan GA, iaitu 89.07%. Namun begitu, keputusan nilai carian relevan yang diperolehi dan F1 adalah kurang memuaskan. Nilai carian relevan yang diperolehi bagi proses carian tanpa GA adalah 80.06% dan

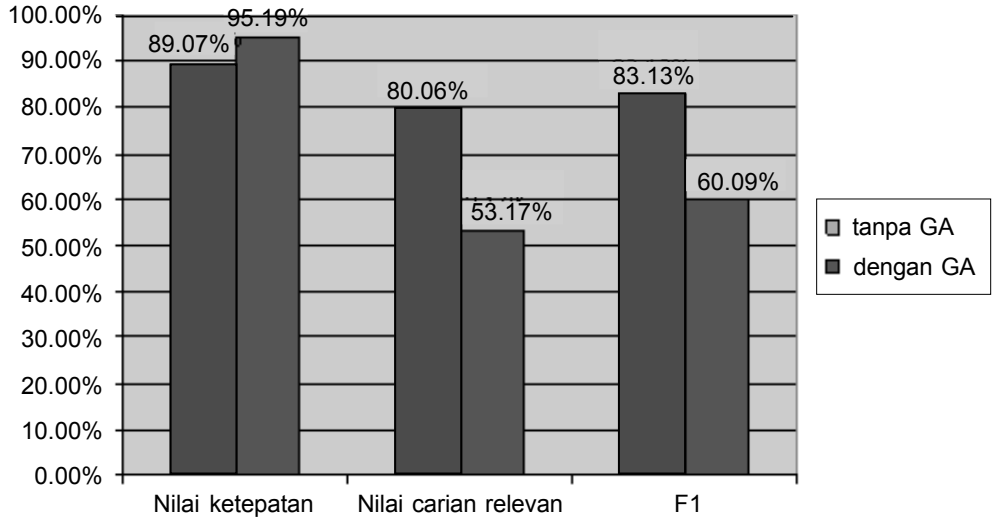
Jadual 8 Purata nilai *Precision, Recall* dan F1 bagi 3 set bidang pengujian

Set	Domain	Keputusan Carian tanpa GA			Keputusan Carian dengan GA		
		Precision	Recall	F1	Precision	Recall	F1
1	Akademik	73.84%	64.11%	67.03%	88.91%	44.90%	54.61%
2	Bank	96.70%	86.42%	89.61%	100%	44.25%	50.72%
3	Telekomunikasi	96.67%	89.65%	92.75%	96.67%	70.35%	74.93%
	Purata	89.07%	80.06%	83.13%	95.19%	53.17%	60.09%

Jadual 9 Perbandingan ciri-ciri antara *web crawler* sedia ada dengan *UtmCrawler*

Bil.	Ciri-ciri	Win Web Crawler [20]	Web Spider [19]	Visual Web Spider [18]	UTM Crawler
1.	Proses *crawling* bermula dengan senarai URL yang dimasukkan oleh pengguna.	✓	✓	✓	✓
2.	Proses *crawling* bermula dengan senarai URL yang telah dimasukkan oleh pengguna sebelumnya.			✓	
3.	Memaparkan laman web yang terpilih.	✓	✓	✓	✓
4.	Memperoleh URL, *meta tag* (tajuk, kata kunci) laman web tertentu.	✓	✓	✓	✓
5.	Memperolehi isi kandungan laman web tertentu.	✓	✓		✓
6.	Memperolehi saiz laman web, tarikh laman web dikemas kini dan lain-lain.	✓		✓	
7.	Menjana rajah hubungan laluan /rajah pokok carian.		✓	✓	
8.	Proses *crawling* bermula dengan kata kunci, bahasa, domain, negara atau kategori yang dimasukkan oleh pengguna.	✓	✓	✓	✓
9.	Memaparkan kesalahan dan status mesej carian.		✓	✓	✓
10.	Menganalisis bilangan perkataan, gambar rajah dan laluan laman web tertentu.		✓		
11.	Parameter carian (bilangan maksimum laman web yang dilayari, aras kedalaman sesuatu carian, senarai kata kunci, domain carian).	✓	✓	✓	✓
12.	Mempunyai banyak penapis (*filter*) sebagai pengehad sesi.	✓		✓	
13.	Membuang URL yang berulangan atau sintaks yang tidak sah secara automatik.			✓	✓
14.	Merungkai URL yang tidak langsung.			✓	✓
15.	Berbilang bebenang (*Multi-threaded*).	✓	✓	✓	✓
16.	Menyimpan data yang diperolehi ke pangkalan data, Microsoft Access atau fail Excel atau mySQL.			✓	✓
17.	Menyimpan data yang diperolehi ke dalam fail Text.	✓		✓	✓
18.	Menyimpan laman web yang dilayari ke dalam fail HTML.			✓	
19.	Mudah digunakan (*user friendly*).		✓	✓	✓
20.	Pengunaan GA untuk meningkatkan keberkesanan carian.				✓
21.	Pengemaskini penyataan carian secara automatik.				✓

menurun kepada 53.17% bagi proses carian dengan GA. Manakala nilai F1 bagi proses carian tanpa GA adalah 83.13% dan menurun kepada 60.09% bagi proses carian dengan GA. Penurunan nilai carian relevan yang diperolehi bagi proses carian dengan GA adalah disebabkan oleh semasa kata kunci carian bertambah, bilangan laman web yang memenuhi carian akan berkurang. Oleh itu, bilangan laman web yang tidak dipersetujui oleh *UtmCrawler* tetapi dipersetujui oleh pengguna akan bertambah. Merujuk kepada formula (4) bahawa semakin tinggi nilai *b* akan menyebabkan nilai carian relevan yang diperolehi semakin rendah. Dalam kajian ini, nilai ketepatan diutamakan kerana matlamatnya adalah untuk membangunkan prototaip *UtmCrawler* yang dapat mengembangkan kata kunci carian menggunakan teknik GA supaya hasil carian yang dijanakan oleh *UtmCrawler* mempunyai nilai ketepatan yang tinggi.



Rajah 12 Hasil pengujian

Jadual 9 pula menunjukkan perbandingan ciri-ciri antara *web crawler* sedia ada dengan *UTMCrawler* yang dibangunkan. Antara ciri *UTMCrawler* adalah proses pertama, iaitu proses *crawling* bermula dengan senarai URL dimasukkan oleh pengguna. Kemudian ia akan memaparkan laman web yang terpilih dengan mendapatkan URL, *meta tag* (tajuk, kata kunci) serta kandungan bagi laman web tersebut. Proses *crawling* oleh *UTMCrawler* bermula dengan kata kunci yang dimasukkan oleh pengguna dan ia akan memaparkan kesalahan dan status mesej carian sekiranya terdapat sebarang ralat ketika proses *crawling* dilaksanakan. *UTMCrawler* juga mempunyai parameter carian yang terdiri daripada bilangan

maksimum laman web yang dilayari, aras kedalaman carian, senarai kata kunci dan domain carian. Semasa proses *crawling*, URL yang berulangan dan sintaks yang tidak sah akan disingkirkan secara automatik dan merungkaikan URL yang tidak langsung. *UTMCrawler* turut mempunyai ciri berbilang bebenang (*multi threaded*) serta menyimpan data yang diperolehi ke pangkalan data Microsoft Access atau fail Excel atau MySQL.

Selain pangkalan data, data yang diperolehi turut disimpan dalam bentuk fail .txt. Antara kelebihan yang terdapat pada *UTMCrawler* adalah ia mudah digunakan kerana sifatnya yang ramah pengguna (*user-friendly*) dan ia melaksanakan teknik algoritma genetik untuk meningkatkan keberkesanan carian berbanding *web crawler* sedia ada yang tidak mengaplikasikan teknik tersebut dalam sistem mereka. Di samping itu, *UTMCrawler* juga mengemas kini penyataan secara automatik bagi memudahkan pengguna memperolehi data yang terkini.

6.0 KESIMPULAN

Kesimpulannya, *UtmCrawler* yang dibangunkan dengan menggunakan teknik algoritma genetik (GA) berfungsi untuk mengatasi masalah hasil carian yang kurang tepat. Keputusan carian yang dipaparkan mempunyai nilai ketepatan dan nilai *fitness* yang tinggi dan tidak memerlukan sebarang latihan serta dan *Relevance Feedback* (RF) dalam proses carian dengan menggunakan GA. Selain itu, satu cara pengumpulan maklumat yang sistematik dan berstruktur digunakan untuk mengatasi masalah maklumat berlebihan (*overload*) dalam penyimpanan maklumat ke dalam pangkalan data. Cara penyimpanan maklumat ke dalam pangkalan data yang sistematik dapat mengurangkan beban pelayan (*server*) dan pangkalan data. Tambahan pula, peristiwa URL yang berulangan dalam senarai hasil carian dan pangkalan data tidak akan berlaku.

PENGHARGAAN

Setinggi penghargaan kepada Pusat Pengurusan Penyelidikan (RMC), Universiti Teknologi Malaysia kerana membiayai projek ini di bawah Vot 79089.

RUJUKAN

[1] Pierre, J. M. 2000. Practical Issues for Automated Categorization of Web Pages. ECDL 2000 Workshop on the Semantic Web. Lisbon, Portugal.

[2] Pant, G. dan F. Menczer. 2002. MySpiders: Evolve Your Own Intelligent Web Crawlers. *Autonomous Agents and Multi-Agent Systems*. Kluwer Academic Publishers. Manufactured in The Netherlands. 5: 221-229.

[3] *www.google.com*, Google search engine, 2006.

[4] *www.altavista.com*, Altavista search engine, 2006.

[5] *www.yahoo.com*, Yahoo! search engine, 2006.

[6] Tsay, J-J., C-Y. Shih dan B-O. Wu. 2005. *AutoCrawler – An Integrated System for Automatic Topical Crawler*. Proceeding of the Fourth Annual ACIS International Conference on Computer and Information Science.

[7] Cho, J. dan H. Garcia-Molina. 2000. *Synchronizing a Database to Improve Freshness*. In Proceedings of ACM International Conference on Management of Data (SIGMOD). Dallas, Texas: USA.

[8] Selamat, A., S. Omatu dan H. Yanagimoto. 2003. *Web News Categorization Using Neural Networks*. IEEJ Transactions on Electrical and Information Systems. 123(5): 1020-1026.

[9] Koster, M. 1995. Robots in The Web: Threat or Treat? *ConneXions*. 9(4).

[10] Pinkerton, B. 1994. *Finding What People Want: Experiences with The WebCrawler*. In Proceedings of the First World Wide Web Conference, Geneva, Switzerland.

[11] Wikipedia, the free encyclopedia http://en.wikipedia.org/wiki. (accessed September 2, 2006).

[12] Luger, G. F. 2002. *Artificial Intelligence Structures and Strategies for Complex Problem Solving*. 4th ed. Addison-Wesley. 81-158.

[13] Negnevitsky, M. 2005. *Artificial Intelligence: A Guide to Intelligence Systems*. 2nd ed. Addison-Wesley. 219-258.

[14] Cho, J., H. Garcia-Molina dan L. Page. 1998. *Efficient Crawling through URL Ordering*. Proc. the 7th International World-Wide Web Conference. Brisbane, Australia, Apr. 1998.

[15] Abiteboul, S., M. Preda dan G. Cobena. 2003. *Adaptive On-line Page Importance Computation*. In Proceedings of the Twelfth International Conference on World Wide Web: 280-290.

[16] Shokouhi, M., P. Chubak dan Z. Raeesy. 2005. *Enchancing Focused Crawling with Genetic Algorithms*. Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'05).

[17] Mitra, M., A. Singhal dan C. Buckley. 1998. *Improving Automatic Query Expansion*. Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Melbourne, Australia.

[18] *www.Newprosoft.com*, 2006.

[19] *www.javaworld.com*, 2006.

[20] *www.WebExtractorSystem.com*, 2006.

[21] *help.yahoo.com,* Yahoo! Help, 2006.

[22] Marcus, A., Daniel N. M. and Ivan Gonzalez. 2006. *Effective Web-Scale Crawling Through Website Analysis*. In Proceedings of the 15th International Conference on World Wide Web. Edinburgh, Scotland: ACM. 1041-1042.