

## **PENYELESAIAN MASALAH PENJADUALAN DINAMIK DENGAN MENGGUNAKAN PENDEKATAN TAAKULAN BERASASKAN KES**

ZALMIYAH ZAKARIA<sup>1</sup> & SAFAAI DERIS<sup>2</sup>

**Abstrak.** Masalah penjadualan waktu universiti telah menjadi satu masalah yang membebankan pihak pengurusan akhir-akhir ini. Persekitaran aktiviti akademik yang sentiasa berubah menyebabkan berlakunya perubahan kepada jadual semasa. Kebanyakan sistem penjadualan yang sedia ada tidak dapat menyelesaikan masalah ini. Oleh yang demikian, taakulan berasaskan kes (CBR) telah dipilih untuk bertindakbalas terhadap sebarang perubahan di dalam sistem penjadualan untuk memastikan penyelesaiannya sentiasa selari dengan perkembangan persekitaran yang dinamik.

*Keywords.* Taakulan berasaskan kes, taakulan berasaskan kekangan, masalah pemuasan kekangan, penjadualan reaktif, penjadualan universiti

### **1.0 PENGENALAN**

Penjadualan ditakrifkan sebagai satu proses pengumpulan satu set sumber yang terhad kepada tempoh masa tertentu untuk melaksanakan satu koleksi aktiviti atau tugas yang dapat memenuhi semua kekangan [1]. Penjadualan dikelaskan sebagai masalah '*NP-complete*' dan penyelesaiannya pula dikatakan bukan polinomial [2] kerana setakat ini tiada satu kaedah penyelesaian optimum telah dilaporkan disebabkan oleh fenomena '*combinatorial explosion*'.

Penjadualan akademik universiti pula adalah salah satu contoh penjadualan. Penjadualan akademik universiti menempatkan satu set pertemuan di antara pensyarah dengan pelajar kepada sumber-sumber tertentu seperti slot masa, bilik kuliah dan pensyarah dengan mengambilkira kekangan-kekangan tertentu seperti kekangan pertembungan masa pensyarah dan kekangan pertembungan masa bilik kuliah [2, 3].

Dalam suasana sebenar, jadual asal yang dihasilkan sering tidak menepati kehendak pengguna oleh kerana keperluannya sering berubah (dinamik). Lazimnya penjadualan semula dilaksanakan untuk menangani perubahan yang berlaku. Namun pendekatan ini masih tidak berupaya mendapatkan jadual baru yang menepati kehendak pengguna, di samping itu tiada kesinambungan di antara jadual asal dengan jadual

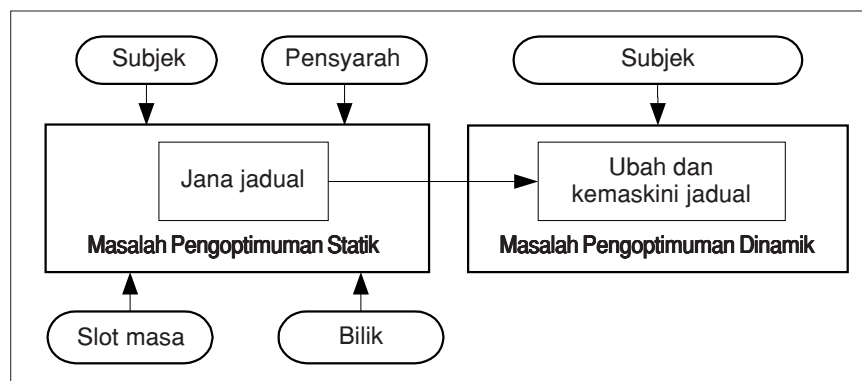
<sup>1,2</sup> Jabatan Kejuruteraan Perisian, Fakulti Sains Komputer dan Sistem Maklumat, Universiti Teknologi Malaysia, Tel: 07-5576160 ext: 1 3857, 2 3843 Fax: 07 5565044 e-mail: <sup>1</sup>zalmiyah@fsksm.utm.my, <sup>2</sup>safaai@fsksm.utm.my

terkini. Walau bagaimanapun, satu pendekatan telah didapati berpotensi mengatasi masalah keperluan dinamik di dalam masalah penjadualan universiti iaitu dengan menggunakan taakulan berasaskan kes atau '*case-based reasoning*' (CBR) yang telah digunakan untuk menangani masalah dinamik penjadualan mesin di kilang pembuatan besi di Kapfenberg, Jerman [4].

## 2.0 MASALAH PENJADUALAN

Penjadualan adalah satu proses yang berterusan. Pelbagai teknik dan pendekatan telah digunakan untuk meningkatkan tahap pencapaian penyelesaian masalah penjadualan [3, 5].

Terdapat dua jenis masalah penjadualan, sebagaimana ditunjukkan di dalam Rajah 1. Pertama adalah masalah pengoptimuman statik yang melibatkan kombinasi masalah tabii, di mana penyelesaian optimum sukar diperolehi disebabkan oleh ketidak-upayaan sistem untuk mempertimbangkan semua nod di dalam ruang carian yang besar. Kebanyakan pendekatan yang telah digunakan untuk menyelesaikan masalah ini hanya dapat menghasilkan penyelesaian hampir-optimum atau sekurang-kurangnya penyelesaian tersaur. Manakala masalah kedua adalah masalah pengoptimuman dinamik yang berkait dengan persekitaran dinamik, di mana pembolehubah dan kekangan sentiasa berubah disebabkan oleh pertumbuhan sesebuah organisasi atau terjadinya peristiwa di luar jangkaan [6].



**Rajah 1** Masalah Penjadualan

Masalah pengoptimuman statik telah mendapat perhatian dari para penyelidik sejak lebih kurang tiga puluh tahun yang lalu. Pelbagai teknik dan pendekatan telah digunakan untuk menyelesaikan masalah tersebut, seperti pewarnaan graf [1], peraturan pengendalian kekangan [7], carian tabu [5] dan masalah pemuasan kekangan [6, 8]. Literatur tentang penyelesaian masalah dinamik pula, masih lagi terhad dan percubaan ke arah menyelesaikannya masih lagi di tahap permulaan [4].

Justeru itu, banyak lagi usaha-usaha yang perlu dilakukan untuk menyelesaikan masalah tersebut.

Di dalam kajian ini, masalah pengoptimuman statik telah diselesaikan dengan menggunakan algoritma hibrid [6] di mana prosidur pemproses kekangan daripada taakulan berasaskan kekangan atau '*constraint-based reasoning*' (CoBR) diserapkan ke dalam algoritma genetik atau "*genetic algorithm*" (GA) untuk menghasilkan satu penyelesaian hampir-optimum. Seterusnya, masalah pengoptimuman dinamik pula diselesaikan dengan pendekatan taakulan berasaskan kes (CBR) di mana pengalaman lepas digunakan untuk menyelesaikan perubahan yang berlaku.

### 3.0 PENJADUALAN REAKTIF

Penjadualan reaktif adalah satu proses mengubahsuai jadual yang sedia ada disebabkan berlakunya peristiwa di luar jangkaan. Proses ini juga dianggap sebagai masalah penjadualan semula satu set tugas di bawah spesifikasi baru atau kekangan yang telah berubah. Sesuatu sistem itu dianggap reaktif apabila ia boleh bertindak balas terhadap perubahan yang berlaku di dalam persekitarannya [8].

Di antara alternatif utama untuk mengemaskini atau membaiki jadual adalah sama ada dengan menjana semula jadual daripada awal berdasarkan kepada keperluan dan spesifikasi baru atau dengan mengubah dan mengemaskini jadual sedia ada bersesuaian dengan situasi baru dengan menggunakan kaedah-kaedah tertentu [4].

Pendekatan pertama mungkin sesuai dilaksanakan untuk memantau penyelesaian yang optimum, tetapi kurang sesuai untuk dilaksanakan secara kerap kerana kekerapan penjanaan semula jadual boleh menyebabkan ketidak-stabilan dan ketidaksinambungan pelaksanaan dengan perancangan sebenar. Masa yang diambil untuk menjana semula jadual juga agak lama. Justeru itu, kebanyakan pendekatan penjadualan reaktif hari ini adalah untuk mengelakkan kekerapan penjanaan semula jadual bagi mengekalkan kesinambungan di antara jadual baru dan jadual asal.

Di antara pendekatan yang telah digunakan untuk membangunkan penjadualan reaktif adalah pendekatan berpandukan-kekangan [7], gabungan CBR, taakulan kabur dan carian tabu [4], pengaturcaraan logik kekangan (CLP) [9] dan teknik carian heuristik di bawah kawalan pengaturcaraan meta [10]. Di dalam kajian ini taakulan berasaskan kes digunakan untuk membangunkan penjadualan reaktif. Manakala algoritma hibrid [2, 6] digunakan untuk menjana jadual awal.

### 3.1 Algoritma Hibrid

Algoritma hibrid (AH) [2, 6] adalah gabungan GA dan CoBR. Penyelesaian ini menggunakan rambatan kekangan melalui algoritma '*arc-consistency*' untuk mengurangkan ruang carian bagi masalah kombinatorial dengan mencantas semua nilai-nilai yang melanggar kekangan. Kekangan binari diwakilkan sebagai satu graf

kekangan, di mana nod mewakili pembolehubah di dalam rangkaian kekangan dan cabang atau 'arc' mewakili kekangan di antara pembolehubah. Oleh itu, pengumpulan nilai 'arc-consistency' bagi setiap pasangan nod dihubungkan oleh kekangan binari dan nilai-nilai yang tidak menepati kekangan dihapuskan daripada domain tersebut. Kemudian GA digunakan untuk menjana penyelesaian yang berpotensi dengan mewakili kekangan dalam bentuk kromosom dan melaksanakan proses 'normal crossover', 'mutation operator' dan penilaian fungsi 'fitness'.

Prosedur pemproses-kekangan digunakan untuk menyemak supaya setiap nilai konsisten dengan kekangan. Nilai yang tidak konsisten akan dihapuskan daripada domain. Oleh kerana setiap nilai domain bagi satu pembolehubah mewakili satu nod di dalam ruang carian, maka penghapusan nilai daripada domain dapat mengurangkan ruang carian dan menyebabkan carian ini dapat dilaksanakan dengan lebih cepat.

### 3.2 Taakulan Berasaskan Kekangan (CoBR)

CoBR adalah satu teknik penyelesaian masalah yang digunakan untuk menyelesaikan masalah pemuasan kekangan atau "constraint satisfaction problems" (CSP). Penyelesaian CSP pula adalah satu proses pengumpulan nilai-nilai kepada pembolehubah dengan memenuhi kekangan-kekangan tertentu [9, 11]. CSP terdiri daripada satu set  $n$  pembolehubah  $X_i$  yang mempunyai satu kesatuan domain  $D_i$  bagi nilai-nilai kemungkinan, di mana  $i = 1, 2, 3, \dots, n$ . Terdapat juga satu set kekangan  $C_k$  yang menghuraikan hubungan di antara pembolehubah, di mana  $k = 1, 2, 3, \dots, q$ . Penyelesaian tersaur adalah satu gabungan pembolehubah yang memenuhi semua kekangan.

Di antara kekangan  $C_k$  yang terlibat di dalam masalah penjadualan universiti ini adalah:

- $C_1$  : Kekangan pertembungan masa pensyarah – subjek-subjek yang diajar oleh pensyarah yang sama tidak boleh diumpukkan kepada slot masa yang sama.
- $C_2$  : Kekangan pertembungan masa kumpulan subjek – subjek-subjek daripada kumpulan yang sama tidak boleh diumpukkan kepada slot masa yang sama.
- $C_3$  : Kekangan pertembungan bilik – satu bilik tidak boleh diumpukkan kepada lebih daripada satu subjek dalam satu slot masa yang sama.
- $C_4$  : Kekangan kapasiti bilik – jumlah pelajar bagi satu subjek yang diumpukkan kepada satu bilik mestilah kurang atau sama dengan kapasiti bilik tersebut.
- $C_5$  : Kekangan bilik dan domain-masa – bilik atau slot masa yang diumpukkan kepada satu subjek mestilah di dalam had domain.

- $C_6$ : Kekangan slot masa – slot masa yang diperlukan untuk aktiviti tertentu seperti ko-kurikulum dan sebagainya, tidak boleh diberikan kepada mana-mana subjek.
- $C_7$ : Kekangan taburan kelas – masa bagi satu subjek perlu ditaburkan ke sepanjang minggu supaya dapat mewujudkan satu persekitaran kelas yang kondusif.
- $C_8$ : Sesetengah pensyarah dan pelajar lebih suka kepada slot masa tertentu seperti masa antara 9.00 hingga 11.00 pagi berbanding waktu tengahari dan petang.
- $C_9$ : Sesetengah pensyarah dan pelajar juga lebih suka kepada bilik tertentu yang mempunyai lebih banyak kelengkapan dan keselesaan.
- $C_{10}$ : Sesetengah pensyarah lebih suka mengajar subjek yang berkaitan dengannya.

Kekangan  $C_7$ ,  $C_8$ ,  $C_9$ ,  $C_{10}$  adalah kekangan-kekangan ringan. Selainnya adalah kekangan berat. Secara umumnya, kekangan-kekangan berat ini bolehlah dikategorikan sebagai:

- Kekangan pertembungan masa pensyarah ( $L$ ), kumpulan subjek ( $G$ ) dan bilik ( $R$ ) yang boleh diwakilkan sebagai:

$$T(S_i) \neq T(S_j) \text{ sekiranya } L(S_i) = L(S_j)$$

$$T(S_i) \neq T(S_j) \text{ sekiranya } G(S_i) = G(S_j)$$

$$T(S_i) \neq T(S_j) \text{ sekiranya } R(S_i) = R(S_j)$$

Di mana  $T(S_i)$  dan  $T(S_j)$  adalah slot masa bagi subjek  $S_i$  dan  $S_j$ ;  $i, j = 1, 2, \dots, n$ . Persamaan ini menunjukkan yang slot masa bagi subjek  $S_i$ ,  $T(S_i)$  mestilah tidak sama dengan slot masa bagi subjek  $S_j$ ,  $T(S_j)$  sekiranya pensyarah  $L$  atau kumpulan subjek  $G$  atau bilik  $R$  bagi kedua subjek itu adalah sama.

- Manakala kekangan kapasiti bilik pula dinyatakan sebagai:  $Z(R(S_i)) \geq N(S_i)$  di mana  $i = 1, 2, \dots, p$ .  $Z(R(S_i))$  adalah kapasiti bilik bagi subjek  $S_i$ , dan  $N(S_i)$  adalah jumlah pelajar bagi subjek  $S_i$ .

### 3.3 Taakulan Berasaskan Kes (CBR)

CBR adalah satu usaha ke arah pembangunan mesin yang boleh meniru cara bagaimana manusia berfikir dan membuat keputusan secara efektif [12]. Modelnya adalah manusia, yang bukan sahaja boleh bercakap tetapi pada masa yang sama boleh berfikir, menaakul, mencapai pengetahuan lepas daripada memori, mencipta idea-idea baru, memahami situasi atau keadaan yang belum pernah ditemuinya dan membuat keputusan berdasarkan kepada pengalaman lepas. Sekiranya pengalaman

lepas mengatakan penyelesaian 'A' adalah terbaik, maka penyelesaian 'A' akan dipilih, sebaliknya jika pengalaman lepas menunjukkan bahawa penyelesaian 'A' tidak baik, manusia yang pintar itu tidak akan membuat kesilapan yang sama dua kali. Maka di sinilah timbulnya motivasi untuk membangunkan CBR untuk membantu menyelesaikan masalah penjadualan. Sebagaimana pengetahuan yang merujuk kepada koleksi pengalaman yang dimiliki oleh manusia, maka kes pula merupakan pengalaman yang disimpan di dalam memori komputer yang dikenali sebagai pangkalan kes. CBR boleh juga dianggap sebagai taakulan berasaskan pengalaman [12].

Apabila berdepan dengan satu masalah baru, CBR akan membandingkan masalah semasa dengan kes-kes sebelumnya untuk menentukan sekiranya ada pengalaman lepas yang boleh menyediakan penyelesaian segera. Sekiranya sistem mendapati ada kes yang sama, tetapi tidak semua atributnya sesuai dengan kes semasa, CBR akan mencipta satu penyelesaian baru dengan mengubahsuai kes yang sedia ada. Kes baru ini akan ditambah ke dalam pangkalan kes untuk kegunaan di masa hadapan. Konsep penting CBR adalah kemampuannya untuk belajar secara berterusan kerana setiap pengalaman yang baru ditemui akan disimpan, setiap kali masalah baru dapat diselesaikan [13]. Selain daripada itu, pengalaman ini juga boleh diguna semula untuk menyelesaikan masalah lain yang bakal muncul.

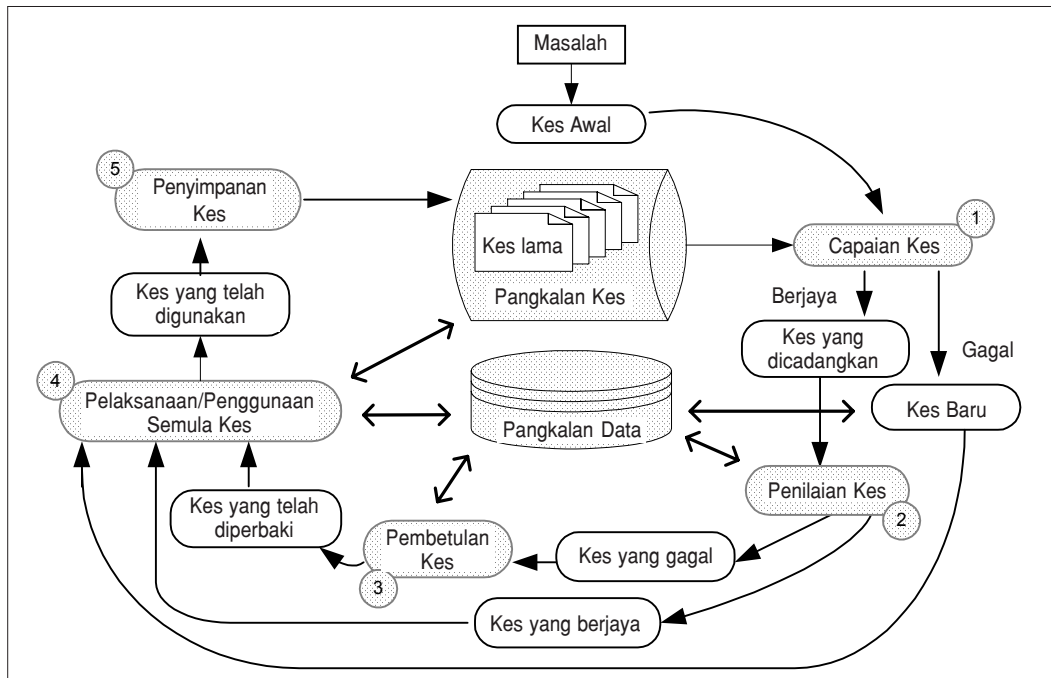
#### **4.0 FAKTOR-FAKTOR DINAMIK DI DALAM PENJADUALAN UNIVERSITI**

Di antara faktor-faktor yang menyebabkan perubahan berlaku ke atas jadual semasa ialah:

- (i) Pertukaran staf akademik
  - Pertambahan staf – sebagai contoh; pensyarah tamat cuti belajar dan pengambilan staf baru.
  - Ketidak-hadiran staf – seperti pensyarah memulakan cuti belajar atau bercuti sakit.
- (ii) Penawaran kursus yang dinamik
  - Pembatalan seksyen atau subjek – contohnya; kekurangan pelajar dan penggabungan seksyen.
  - Penambahan seksyen atau subjek baru – sebagai contoh; jumlah pelajar berlebihan.
  - Pertambahan atau pembatalan aktiviti (syarahan, tutorial atau seminar).
  - Pertukaran pensyarah (pertukaran pensyarah di antara dua subjek).
  - Penggabungan aktiviti pengajaran (dua seksyen atau kelas tutorial digabungkan).
- (iii) Permintaan untuk perubahan slot masa atau bilik – permintaan pensyarah.

## 5.0 PERMODELAN DAN PEMBANGUNAN ALGORITMA CBR

Rajah 2 menunjukkan satu kitaran lengkap CBR yang telah dibangunkan. Terdapat 5 fasa utama di dalam kitaran ini iaitu capaian, penilaian, pembetulan, pelaksanaan atau penggunaan semula dan penyimpanan kes yang telah diubahsuai daripada kitaran CBR Aamodt dan Plaza [13].



**Rajah 2** Kitaran Taakulan Berasaskan Kes (CBR) (diubahsuai daripada kitaran CBR Aamodt & Plaza [13])

Apabila ada masalah baru, sistem menggunakan masalah tersebut untuk mencipta nama kes dan menggunakannya sebagai kata-kunci untuk membuat capaian kepada pangkalan kes (Langkah 1). Sekiranya ada kes lepas yang serupa dengan masalah tersebut, ianya akan dicadangkan kepada sistem. Sebelum itu, kes tersebut perlu dinilai untuk memastikan setiap penyelesaian yang dicadangkan tidak melanggar kekangan semasa. Penilaian dibuat sama ada dengan menguji penyelesaian kepada situasi semasa atau dengan mendapatkan maklumbalas daripada pengguna (Langkah 2). Kes yang berjaya akan digunakan untuk menyelesaikan masalah tersebut (Langkah 4). Sebaliknya kes yang gagal akan dibaiki dan diadaptasi sebelum digunakan untuk menyelesaikan masalah semasa dan disimpan semula ke dalam pangkalan kes (Langkah 3). Sebaliknya, jika capaian tidak menemui sebarang kes yang sama, sistem akan mencipta satu kes baru hasil daripada pelaksanaan penyelesaian masalah tersebut. Kes baru akan disimpan ke dalam pangkalan kes, setiap kali masalah baru dapat diselesaikan (Langkah 5). Proses pembelajaran berlaku

apabila ada kes baru yang mewakili pengalaman disimpan ke dalam pangkalan kes yang menyebabkan penambahan pengetahuan.

### 5.1 Perwakilan Kes

Kecekapan CBR adalah bergantung kepada struktur dan kandungan kes yang terdapat di dalam memori kes atau pangkalan kes. Kes adalah stuktur data yang menyimpan maklumat atau pengetahuan tertentu yang terikat dengan domain tertentu. Struktur kes ini berbeza berdasarkan kepada matlamat ia dibangunkan.

Di dalam kajian ini, kes diwakilkan di dalam bentuk bingkai dan slotnya terdiri daripada

- Matlamat atau nama masalah yang diwakilkan sebagai *p*.
- Satu set penerangan masalah iaitu *id* subjek yang diwakilkan sebagai *sid* dan *id* seksyen yang diwakilkan sebagai *secid*.
- Maklumat jadual semasa yang diwakilkan sebagai *tt*: *tt* menyimpan nilai-nilai bagi *id* pertemuan yang diwakilkan sebagai *lessid*, *id* slot masa yang diwakilkan sebagai *tsid*, *id* bilik yang diwakilkan sebagai *rid*, *id* subjek yang diwakilkan sebagai *sid*, *id* pensyarah yang diwakilkan sebagai *lid*, jumlah pelajar yang diwakilkan sebagai *nos*, *id* seksyen yang diwakilkan sebagai *secid*, jumlah pertemuan yang diwakilkan sebagai *nol* dan *id* kumpulan-subjek yang diwakilkan sebagai *sgid*.
- Perincian penyelesaian semasa yang diwakilkan sebagai *s*.
- Langkah-langkah yang dilaksanakan untuk menyelesaikan masalah diwakilkan sebagai *steps*.
- Maklumat jadual setelah dikemaskini atau setelah penyelesaian terlaksana diwakilkan sebagai *utt*. Ia juga menyimpan nilai-nilai yang sama sebagaimana maklumat jadual semasa *tt*.

Penerangan kes yang dimaksudkan adalah seperti di dalam Jadual 1:

**Jadual 1** Perwakilan Attribut dalam Bingkai serta Contoh Nilai

Slot	Nilai	Penerangan
lessid	7416	Pertemuan pertama bagi subjek 74
tsid	35	Jumaat : 10.00 – 10.50 pagi
rid	4	Bilik Kuliah 6, N24
sid	74	Projek 1
lid	24	Pensyarah : Muhammad bin Abdullah
nos	90	Pelajar 90 orang
secid	6	Seksyen 06
nol	2	2 jam pertemuan seminggu
sgid	12	Kumpulan Teras Fakulti



Manakala struktur kes yang disimpan di dalam pangkalan kes adalah seperti Rajah 3.

```

((CHANGE-TT 74 6) (SUBJECT 74) (SECTION 6)
  (CURRENT-TT ((7416 35 4 74 24 90 6 2 12)
                (7426 8 5 74 24 90 6 2 12))))
  ((LESSON 7416) (TIMESLOT 4) (ROOM 4))
  ((LESSON 7426) (TIMESLOT 35) (ROOM 4)))
(STEPS (CHECK-ROOM-CONSTRAINT RID NOS)
  (CHECK-LECTURER-CONSTRAINT TS LEC)
  (CHECK-SGROUP-CONSTRAINT SID TS)
  (UPDATE-TTL TS RID LESSID))
(UPDATED-TT ((7416 4 4 74 24 90 6 2 12)
              (7426 35 4 74 24 90 6 2 12))))

```

**Rajah 3** Struktur Kes

Kes ini mewakili penyelesaian untuk pertukaran jadual bagi subjek id 74 dan seksyen 6, di mana nama kes adalah (CHANGE-TT 74 6). Nama ini digunakan sebagai kata kunci untuk membuat capaian kepada kes lama. (CURRENT-TT) menyimpan maklumat penyelesaian jadual semasa bagi subjek 74. Ia adalah merupakan penyelesaian terkini selepas jadual tersebut diubah. Manakala senarai seterusnya adalah perincian tentang penyelesaian yang dicadangkan oleh kes. (STEPS) pula menyimpan senarai tindakan yang dilaksanakan oleh sistem untuk menyelesaikan masalah yang berkenaan. Senarai tindakan tersebut adalah terdiri daripada (CHECK-ROOM-CONSTRAINT) yang berfungsi untuk memastikan nilai yang dicadangkan tidak melanggar kekangan bilik, (CHECK-LECTURER-CONSTRAINT) berfungsi untuk memastikan nilai yang dicadangkan tidak melanggar kekangan pensyarah, (CHECK-SGROUP-CONSTRAINT) berfungsi untuk memastikan nilai yang dicadangkan tidak melanggar kekangan kumpulan-subjek, dan (UPDATED-TTL) – berperanan untuk mengemaskini maklumat di dalam pangkalan data. Manakala (UPDATED-TT) pula menyimpan maklumat penyelesaian yang telah dikemaskini.

## 5.2 Capaian Kes

Capaian kes adalah satu isu penting di dalam CBR. Matlamatnya adalah untuk mendapatkan satu kes yang sepadan dengan situasi semasa supaya penyelesaian kes tersebut boleh digunakan untuk menyelesaikan masalah baru. Masalah baru dianggap sebagai satu matlamat yang perlu dicapai. Matlamat berkenaan diwakili oleh nama masalah  $p$ . Apabila satu masalah baru timbul, sistem akan menggunakan nama masalah tersebut sebagai kata-kunci untuk mencapai kes yang sepadan daripada pangkalan kes. Sekiranya kes yang sepadan ditemui, pencadang kes akan mencadangkan kes tersebut kepada sistem untuk dilaksanakan. Sebaliknya jika tiada sebarang kes yang serupa ditemui, sistem akan menghasilkan satu penyelesaian baru dengan bantuan CoBR. Penyelesaian yang dihasilkan akan digunakan untuk

mencipta satu kes baru. Kes baru berkenaan seterusnya akan disimpan ke dalam pangkalan kes supaya ianya boleh diguna semula pada masa hadapan. Kes-kes di dalam pangkalan kes ini diindeks secara jaringan diskriminasi iaitu berdasarkan nama kes. Pengindeksan jaringan diskriminasi dapat mempercepatkan proses carian kerana pepadanan tidak perlu dilakukan ke atas semua kes yang terdapat di dalam pangkalan kes. Sebaliknya ia hanya melibatkan beberapa kumpulan kes yang mempunyai persamaan dengan masalah semasa sahaja [14]. Capaian kes pula dibuat berdasarkan kepada perbandingan predikat iaitu dengan membandingkan nama masalah sasaran dengan setiap nama kes yang terdapat di dalam pangkalan kes.

Sebarang perubahan dilakukan dengan memasukkan maklumat yang berkaitan seperti jenis perubahan, id subjek dan seksyen bagi subjek tersebut. Rajah 4 menunjukkan hasil capaian kes yang diperolehi oleh sistem.

```
(CHANGE-TIMESLOT)
Enter Subject and Section : 90 1
CBR >> Retrieving case...
CBR >> Match !!!
CBR >> Case found...
CBR >> Retrieved case : ((CHANGE-TIMESLOT 90 1) (SUBJECT 90) (SECTION 1)
                        (CURRENT-TT ((9011 15 8 90 24 100 1 3 10)
                                      (9021 2 7 90 24 100 1 3 10)
                                      (9031 9 7 90 24 100 1 3 10)))
                        (((LESSON 9011) (TIMESLOT 15) (ROOM 8))
                         ((LESSON 9021) (TIMESLOT 2) (ROOM 7))
                         ((LESSON 9031) (TIMESLOT 9) (ROOM 7)))
                        (STEPS (CHECK-ROOM-CONSTRAINT RID NOS)
                              (CHECK-LECTURER-CONSTRAINT TS LEC)
                              (CHECK-SGROUP-CONSTRAINT SID TS)
                              (UPDATE-TTL TS RID LESSID))
                        (UPDATED-TT ((9011 31 8 90 24 100 1 3 10)
                                     (9021 15 7 90 24 100 1 3 10)
                                     (9031 2 7 90 24 100 1 3 10))))
CBR >> Proposed Solution : (TT ((9011 15 8) (9021 2 7) (9031 9 7)))
```

**Rajah 4** Kes yang telah Dicapai

### 5.3 Penilaian Kes

Penyelesaian yang dicadangkan mesti dinilai untuk memastikan ianya sah dan menepati semua kekangan dan keperluan masalah semasa.

```
CBR >> Evaluating case... for solution (9011 15 8)
      Check, either room 8 free or not at timeslot 15 ...
      Room 8 free at timeslot 15
      Check lecturer time constraint, lecturer 24 timeslot 15 ...
      Lecturer 24 not satisfy constraints...
CBR >> Case failed...
CBR >> Timeslot 15 violate constraints
```

**Rajah 5** Penilaian Kes

Sebagai contoh, Rajah 5 memaparkan penyelesaian yang tidak menepati kekangan slot masa 15 dan bilik 8. Daripada penilaian yang dibuat, sistem mendapati penyelesaian tersebut gagal kerana pensyarah 24 didapati perlu menghadiri kelas lain pada slot masa 15. Oleh yang demikian, kes tersebut dibaiki oleh pembaiki kes dengan mencari slot masa lain yang memenuhi kekangan bilik, slot masa kumpulan subjek dan pensyarah sebagaimana tertera pada Rajah 7. Sebaliknya jika penilaian tersebut berjaya seperti pada Rajah 6, bilik 7 akan digunakan untuk menggantikan bilik 8 dan seterusnya sistem akan mengemaskini rekod di pangkalan data.

```

CBR >> Evaluating case...
        Check, either room 7 free or not at timeslot 32...
        Room 7 free at timeslot 32
        Check room constraint, room 7 no of student 100...
        Room 7 satisfy the constraints
CBR >> Case successful...
        Update database for lesson 9021...
        Data for lesson 9021 successfully inserted...
CBR >> Case successfully, reused...

```

**Rajah 6** Penilaian Kes

#### 5.4 Pembaikan Kes

Kes yang gagal akan dibaiki dan disesuaikan dengan situasi semasa. Di sini, terdapat dua pendekatan yang digunakan untuk membaiki penyelesaian kes iaitu sama ada dengan cara:

- Mendapatkan maklum balas daripada pengguna atau
- Sistem memilih penyelesaian yang terbaik berdasarkan kepada kecenderungan penyelesaian

Seterusnya kes yang telah dibaiki akan disimpan semula ke dalam pangkalan kes.

```

CBR >> Repairing case...
        Search other t/slot which satisfy room, t/slot and lecturer constraint
        Free timeslot order by priority (9 32 36)
        Please choose timeslot for lesson 9011... 9
        Update database for lesson 9011 ...
        Data for lesson 9011 successfully inserted...
CBR >> Case for lesson 9011 successfully repaired...
CBR >> New solution ((9011 9 8) (9021 15 7) (9031 2 7))

```

**Rajah 7** Pembaikan Kes

Berdasarkan Rajah 7, sistem mendapati hanya tiga slot masa yang memenuhi kekangan iaitu (9 32 36) yang telah disusun mengikut keutamaan masa. Keutamaan

masa dinilai berdasarkan kecenderungan pelajar dan pensyarah terhadap slot masa tersebut. Sistem akan memilih slot masa yang paling tinggi keutamaannya sebagai penyelesaian. Sebaliknya pengguna juga boleh bertindak untuk memilih sendiri slot masa yang dikehendakinya. Dari slot masa yang baru diperolehi itu, sistem akan memperbaiki kes iaitu dengan membuat penggantian-semula, iaitu menggantikan penyelesaian lama dengan penyelesaian baru. Seterusnya kes yang telah diadaptasi itu akan diguna-semula untuk menyelesaikan masalah semasa serta mengemaskini pangkalan data dan pangkalan kes.

### 5.5 Adaptasi Kes

Di dalam kajian ini, adaptasi kes berlaku selari dengan proses pembaikan kes, di mana sekiranya kes gagal, maka sistem akan membaiki penyelesaian tersebut dan diadaptasi dengan menukar-ganti penyelesaian baru dengan penyelesaian lama. Kaedah ini dikenali sebagai proses penggantian semula. Kes yang telah diadaptasi akan disimpan semula ke dalam pangkalan kes untuk membolehkan ia diguna-semula pada masa akan datang.

Semasa proses adaptasi, maklumat semasa di dalam pangkalan kes dan pangkalan data akan digunakan. Penggunaan maklumat ini menggambarkan tindakbalas di antara penggunaan pengalaman (dalam pangkalan kes) dan pengetahuan umum (dalam pangkalan data) di dalam satu sistem CBR.

## 6.0 PERLAKSANAAN DAN HASIL KAJIAN

Kajian ini menggunakan maklumat perancangan penjadualan kursus Fakulti Sains Komputer dan Sistem Maklumat (FSKSM), Universiti Teknologi Malaysia (UTM) sebagai kajian kes. FSKSM yang mula ditubuhkan pada 1982 ini, kini mempunyai lebih kurang 3557 orang pelajar. Setiap tahun pengambilan pelajarannya kian meningkat dan kini mencecah 600 orang setiap tahun. FSKSM menawarkan pengajian peringkat diploma, ijazah, sarjana dan doktor falsafah di dalam bidang yang berkaitan dengan Sains Komputer. Pada setiap semester, setiap kursus akan menawarkan beberapa bilangan subjek yang berbeza. Bagi setiap subjek itu pula mempunyai 1 hingga 3 kali pertemuan seminggu, bergantung kepada jam kredit subjek tersebut. Setiap subjek juga mungkin mempunyai beberapa seksyen yang mungkin dikendalikan oleh tenaga pengajar yang berbeza. Jumlah tenaga pengajarnya adalah lebih kurang 150 orang, dengan sokongan lebih kurang 50 orang pensyarah daripada fakulti lain.

Langkah pertama di dalam proses penjanaan jadual ini adalah memodelkan masalah penjadualan sebagai satu masalah pemuasan kekangan (CSP). Permodelan dibuat berdasarkan kepada maklumat di dalam Jadual 2.

Seterusnya, jadual awal dijana dengan menggunakan algoritma hibrid [9, 10] dan menghasilkan penyelesaian hampir-optimum. Kemudian hasil penjadualan tersebut

**Jadual 2** Perincian Nilai Parameter dan Hasil Larian Penjanaan Semula Jadual

<b>Perkara</b>	<b>Kuantiti</b>
Jumlah Subjek	74
Jumlah Pertemuan	456
Jumlah Pensyarah	150
Jumlah Slot masa yang digunakan	36
Jumlah hari per minggu	5
Jumlah slot masa per hari	9
Jumlah Pelajar	<> 3557
Jumlah Bilik	22
Jumlah Bilik dengan kapasiti <60 orang	12
Jumlah Bilik dengan kapasiti <120 orang	8
Jumlah Bilik dengan kapasiti <250 orang	2

dipaparkan oleh antaramuka penjadualan interaktif bagi memudahkan pengguna menilai dan menganalisa keputusan untuk memastikan semua penyelesaian tersebut konsisten dan mencapai penyelesaian hampir-optimum.

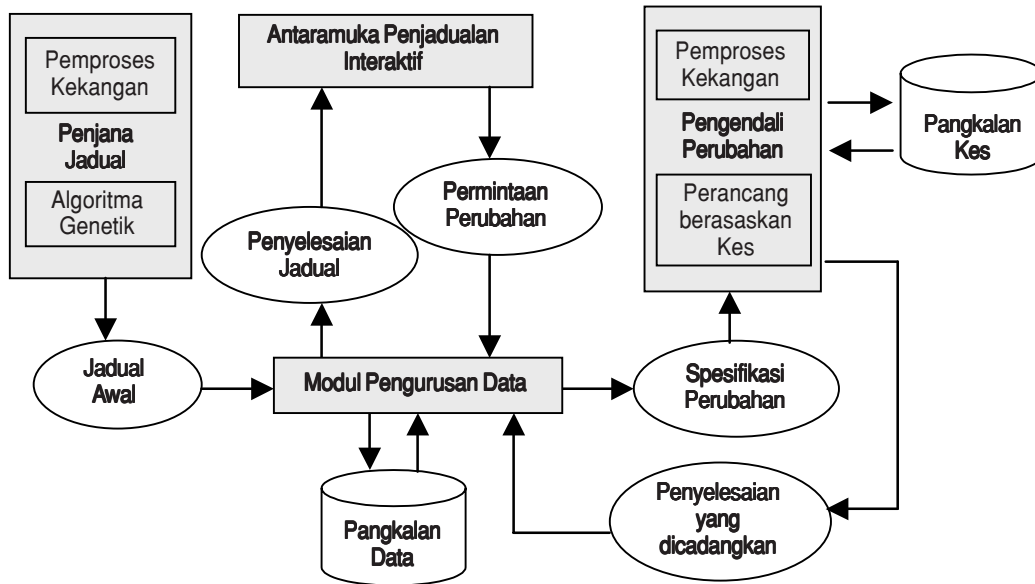
Setelah jadual awal siap dijana, pendekatan CBR akan digunakan untuk mengubah dan mengemaskini jadual apabila sebarang perubahan diperlukan.

## 6.1 Senibina dan Spesifikasi Sistem

Secara umumnya senibina sistem yang dibangunkan adalah seperti di dalam Rajah 8.

Rajah 8 menggambarkan bahawa terdapat 4 komponen utama di dalam sistem ini, iaitu:

- (i) Antaramuka penjadualan interaktif (API) adalah komponen yang membenarkan pengguna berinteraksi dengan sistem dan berperanan sebagai pengantara untuk menerima arahan, menyimpan serta membekal maklumat kepada sistem dan memaparkan keputusan yang diperolehi daripada pangkalan data.
- (ii) Penjana jadual menggunakan algoritma hibrid yang menggabungkan GA dan CoBR untuk menjana jadual awal dan menyimpan maklumat penyelesaian ke dalam pangkalan data;
- (iii) Pengendali perubahan menggunakan CBR dengan sokongan CoBR untuk mengendalikan perubahan ke atas jadual semasa dan mengemaskini pangkalan data dan pangkalan kes.
- (iv) Modul pengurusan data yang mengurus dan memproses maklumat yang diperlukan oleh sistem penjadualan dengan memanipulasi maklumat di dalam pangkalan data



Rajah 8 Senibina Sistem

Algoritma hibrid (AH) telah dibangunkan dengan menggunakan perisian Borland C++ 5.02. Manakala CBR pula dibangunkan dengan perisian Allegro CL 5.0.1. Seterusnya antaramuka pengguna dibangunkan dengan menggunakan perisian JBuilder 3 dan pangkalan datanya adalah Oracle8i. Penerangan yang lebih terperinci boleh dilihat di dalam Jadual 3.

Jadual 3 Persekitaran Pembangunan

Perkara	Pelayan	Pelanggan
Perisian Antaramuka	-	JBuilder 3
CBR	-	Allegro CL 5.0.1
Algoritma Hibrid	-	Borland C++ 5.02
Pangkalan data	Oracle8i Server	Oracle8i Client
Sistem Pengoperasian	Windows NT 4.0	Windows 98
Pemproses	Pentium III 600	Pentium III 450

## 6.2 Hasil Pelaksanaan

Daripada pembangunan yang telah dilakukan, sistem ini telah dapat melaksanakan perubahan-perubahan berikut:

- Tukar jadual bagi satu subjek tertentu
- Tukar slot masa bagi satu pertemuan tertentu

- Tukar slot masa bagi satu subjek tertentu
- Tukar bilik bagi satu pertemuan tertentu
- Tukar bilik bagi satu subjek tertentu
- Tukar pensyarah bagi satu subjek tertentu
- Tambah satu subjek atau seksyen baru
- Hapus satu subjek atau seksyen

Perubahan ini boleh dilakukan dengan bantuan pengguna atau dengan dikendalikan sepenuhnya oleh sistem.

Satu pengujian telah dibuat untuk membandingkan masa yang diambil untuk membuat perubahan dengan menggunakan pendekatan penjadualan semula algoritma hibrid dan juga pendekatan CBR. Jadual 4 menunjukkan masa yang diambil untuk menjana semula jadual adalah 0:18:49:7 iaitu lebih daripada 18 minit dan penyelesaian yang diperolehi dipercayai telah mengubah hampir keseluruhan jadual asal.

**Jadual 4** Perincian Hasil Larian Penjanaan Semula Jadual dengan Algoritma Hibrid

Perkara	Nilai
Jumlah populasi	100
Jumlah generasi	100
Saiz genome	456
Saiz allele	36
Fitness function	Kecenderungan pensyarah dan pelajar
Masa mula	10:25:3:62
Masa tamat	10:43:52:55
Masa yang diambil	0:18:49:7

Manakala Jadual 5 pula menunjukkan masa yang diambil untuk melakukan perubahan dengan menggunakan pendekatan CBR dengan bilangan kes sebanyak 500 kes. Perubahan yang dilakukan hanya melibatkan subjek, seksyen dan pensyarah yang terlibat sahaja, manakala jadual asal yang lainnya tidak berubah.

Hasil pengujian menunjukkan bahawa perubahan yang dilakukan dengan bantuan pengguna, berserta kes adalah lebih cepat berbanding dengan perubahan tanpa kes. Walau bagaimanapun perbezaan masanya tidak begitu ketara iaitu 1 hingga 5 saat sahaja. Tetapi berlaku sebaliknya pula kepada perubahan yang dilakukan dengan bantuan sistem, di mana perubahan dengan kes lebih memakan masa berbanding dengan perubahan tanpa kes. Perbezaan masanya agak ketara, iaitu 1 hingga 11 saat.

Selain daripada itu, perubahan yang dilakukan dengan kes, berserta pembetulan pula lebih memakan masa daripada kes tanpa pembetulan. Perbezaan masa yang terlibat adalah 2 hingga 10 saat.

**Jadual 5** Masa yang Diambil untuk Melakukan Sebarang Perubahan dengan CBR

Spesifikasi Masalah	Pertemuan		Pengguna dengan kes		Pengguna tanpa kes	Sistem dengan kes		Sistem tanpa kes
	tanpa kes	dengan kes	tanpa pem-betulan	dengan pem-betulan		tanpa pem-betulan	dengan pem-betulan	
Tukar jadual	–	–	2 saat	12 saat	13 saat	14 saat	16 saat	5 saat
Tukar slot masa	17 saat	2 saat	5 saat	8 saat	9 saat	3 saat	8 saat	3 saat
Tukar bilik	2 saat	2 saat	5 saat	7 saat	10 saat	3 saat	9 saat	2 saat
Tukar pensyarah	–	–	2 saat	12 saat	3 saat	3 saat	12 saat	3 saat
Tambah subjek/ seksyen	–	–	5 saat	3 saat	6 saat	13 saat	14 saat	12 saat
Hapus subjek/ seksyen	–	–	–	–	10 saat	–	–	–

Seterusnya didapati perubahan yang dilakukan dengan bantuan pengguna adalah lebih cepat daripada perubahan dengan bantuan sistem. Ini adalah disebabkan penyelesaian yang diperolehi daripada sistem melibatkan pemeriksaan dan pemilihan daripada semua sumber yang terlibat.

Walaupun bagaimanapun, sebenarnya masa yang diambil untuk melakukan sebarang perubahan adalah tidak tetap kerana ia bergantung kepada beberapa faktor iaitu:

- jumlah sumber kosong yang ada
- kekompleksan pembetulan yang dilakukan
- jumlah kes yang terdapat di dalam pangkalan kes

Masa pelaksanaan bagi perubahan yang melibatkan pengguna pula adalah bergantung kepada masa yang diambil oleh pengguna untuk memberi tindak balas kepada sistem, sekiranya pengguna mengambil masa yang lama untuk memberikan input, maka masa larian akan menjadi lebih lama dan begitulah sebaliknya. Daripada jadual ini didapati masa yang paling lama untuk membuat perubahan adalah sebanyak 17 saat.

Daripada perbandingan yang telah dilakukan ini, jelas menunjukkan bahawa banyak masa yang telah dapat dijimatkan apabila melakukan perubahan jadual dengan menggunakan pendekatan CBR. Ini adalah kerana di dalam kaedah konvensional (penjana semula), sebarang perubahan adalah melibatkan pengumpulan semula sumber-sumber kepada semua pembolehubah (subjek-subjek) yang terdapat di dalam jadual. Oleh itu, ia meliputi semua proses penjana jadual daripada awal. Oleh sebab itu, proses ini memakan masa yang lama. Sedangkan dengan menggunakan pendekatan CBR, suatu penyelesaian itu dapat



dihasilkan dengan tepat dan pantas kerana ia hanya melibatkan proses pengumpulan beberapa pembolehubah (subjek) yang terlibat sahaja. Nilai bagi pembolehubah lain dikekalkan dan digunakan sebagai panduan (untuk memproses kekangan) untuk menentukan nilai bagi pembolehubah yang terlibat. Oleh itu, daripada perbandingan ini didapati pelaksanaan pendekatan CBR 63.5 kali lebih cepat atau 98.49 % lebih cepat berbanding penjadualan semula.

## 7.0 KESIMPULAN

Walaupun masalah penjadualan reaktif memberi impak besar di dalam menentukan kecekapan satu sistem penjadualan, tetapi literatur yang membincangkan tentang penyelesaian masalah dinamik, masih lagi terhad dan percubaan ke arah menyelesaikannya masih lagi di tahap permulaan. Justeru itu, banyak lagi usaha-usaha yang perlu dilakukan untuk menyelesaikan masalah tersebut.

Kajian ini mendapati CBR adalah salah satu pendekatan yang sesuai digunakan untuk mengendalikan masalah dinamik memandangkan ia boleh menyimpan pengalaman yang lepas dan dapat meningkatkan kecekapan pelaksanaan penyelesaian masalah dengan kaedah guna-semula. CBR juga dapat menjimatkan masa dan tenaga yang diperlukan untuk membuat sebarang perubahan kerana dengannya sistem tidak perlu lagi menjana semula jadual asal. Selain daripada itu, CBR juga dapat mengurangkan kitaran perolehan pengetahuan, kerana kes-kes baru sentiasa ditambah ke dalam pangkalan kes.

## RUJUKAN

- [1] Baker, K. R. 1974. *Introduction to Sequencing and Scheduling*. New York: John Wiley.
- [2] Safaai Deris. 1997a. *Studies on Intelligent Optimization Techniques for Planning, Scheduling and Timetabling*. Osaka Prefecture University: Tesis Ph.D.
- [3] Burke, E.K., Jackson, K.S., Kingston J.H. and Weare, R.F. 1997. Automated Timetabling: The State of the Art. *The Computer Journal*. 40(9): 565–571.
- [4] Dorn, J. 1995. *Case-based reactive scheduling*. Roger Kerr and Elisabeth Szelke (eds) *Artificial Intelligence in Reactive Scheduling*. London: Chapman & Hall. 32–50.
- [5] Schaerf, A. 1995. *A Survey of Automatic Timetabling*. Technical Report, Centrum vpr Wiskunde en Informatica.
- [6] Safaai Deris, Omatu, S., Ohta, H. and Puteh Saad. 1999. Incorporating constraint propagation in genetic algorithm for university timetable planning. *Journal of Eng. Application of AI*.
- [7] Abdennadher, S. and Marte, M. 1998. *University Timetabling using Constraint Handling Rules*. Journées Francophones de Programmation en Logique et Programmation par Contraintes (JFPLC), Nantes.
- [8] Vassileva, J. 1995. *Reactive Instructional Planning To Support Interacting Teaching Strategies*. Proceedings of AI-Ed'95, 334-342. Charlottesville, VA: AACE.
- [9] Goltz, H.J. and Matzke, D. 1999. *University Timetabling using Constraint Logic Programming*. Practical Aspects of Declarative Languages, G.Gupta (ed.). Lecture Notes in Computer Science. 1551. Springer-Verlag. 320–334.
- [10] Koo, J., Jung, J.J. and Jo, G.S. 1995. *Reactive Scheduling in Meta-Programming: A Case Study of Course Scheduling*. Proc. of Pacific-Asian Conference on Expert Systems '95.
- [11] Riesbeck, C. K. and Schank, R. S. 1989. *Inside Case-Based Reasoning*. Northvale, NJ: Erlbaum.

- [12] Kolodner, J. 1993. *Case-Based Reasoning*. George Institute of Technology, Morgan Kaufman Publishers.
- [13] Aamodt, A. and Plaza, A. 1994. *Case-based Reasoning: Foundational Issues, Methodological Variants and System Approaches*. Artificial Intelligence Communications (AICom), IOS Press. 7(1): 39–59.
- [14] Zalmiyah Zakaria. 2001. *Penyelesaian Masalah Penjadualan Reaktif dengan menggunakan Taakulan Berasaskan Kes (CBR)*. Universiti Teknologi Malaysia: Tesis Sarjana.