



### **Hak cipta dan penggunaan kembali:**

Lisensi ini mengizinkan setiap orang untuk mengubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

### **Copyright and reuse:**

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

## BAB II

### LANDASAN TEORI

#### 2.1 *Curriculum Based Course Timetabling (CB-CTT)*

CB-CTT berkaitan dengan penugasan satu set kelas perkuliahan ke satu set ruangan dan jangka waktu setiap minggu, sesuai dengan *constraint* yang diberikan (Bolaji, 2011). Jadwal dianggap layak setelah semua kelas perkuliahan telah ditugaskan dalam kurun waktu dan ruang sehubungan dengan *hard constraints* (H1 - H4). Setelah memenuhi empat *hard constraints*, akan diberikan biaya penalti untuk pelanggaran empat *soft constraints* (S1 - S4), tujuan utama dari masalah CB-CTT adalah untuk meminimalkan jumlah pelanggaran *soft constraints* dalam penentuan solusi, dengan syarat seluruh *hard constraint* tidak dilanggar. *Soft constraints* dan *hard constraints* didefinisikan sebagai berikut:

##### *Hard Constraint*

- H1 Kelas perkuliahan (*lectures*) : semua kelas perkuliahan harus ditugaskan dalam periode waktu yang berbeda dan sebuah ruangan
- H2 Ruang perkuliahan (*room occupancy*): dua kelas perkuliahan tidak boleh dijadwalkan dalam ruangan yang sama dan periode waktu yang sama
- H3 *Conflicts* : Kelas perkuliahan dalam satu kurikulum atau yang diajarkan oleh pengajar yang sama, harus dijadwalkan dalam waktu yang berbeda.
- H4 Ketersediaan (*Availiability*) : Jika pengajar dari suatu mata kuliah tidak tersedia dalam periode waktu yang diberikan, maka tidak ada alokasi kelas perkuliahan untuk periode waktu tersebut.

### *Soft constraint*

- S1 Kapasitas ruangan (*room capacity*) : Jumlah murid yang ada dalam satu kelas perkuliahan harus lebih kecil atau sama dengan kapasitas ruangan yang dipakai
- S2 Stabilitas ruangan (*Room stability*) : Setiap kelas dari mata kuliah tertentu harus dijadwalkan ke dalam ruangan yang sama.
- S3 Kepadatan Kurikulum (*Curriculum Compactness*) : Kelas mata kuliah dalam kurikulum yang sama dijadwalkan dalam periode waktu yang berdekatan.
- S4 *Minimum Working Days* : Kelas dari tiap mata kuliah harus tersebar dalam beberapa hari tertentu.

Deskripsi masalah CB-CTT adalah melakukan penugasan set mata kuliah  $C = c_1, c_2, \dots, c_{tn}$  sebanyak  $tn$  terhadap set kelas  $R = r_1, r_2, \dots, r_{tt}$  sebanyak  $tt$  dan dalam set periode  $P = p_1, p_2, \dots, p_{tp}$ , dengan periode adalah komposisi minimum working days  $td$  dan time slots per working days  $tt$ ,  $tp = td \times tt$ . Tiap mata kuliah  $c_i$  berisi pertemuan  $l_i$ , yang harus ditugaskan dengan periode waktu yang berbeda. Permasalahan berisi sebuah set kurikulum  $CU = Cu_1, Cu_2, \dots, Cu_{tc}$  tiap kurikulum  $Cu_i$  adalah sebuah grup yang berisi kumpulan mata kuliah yang berisi kumpulan set mahasiswa  $std_i$ . Tiap mahasiswa akan ditugaskan ke dalam sebuah set kumpulan mata kuliah.

Gambar 2.1 adalah solusi penjadwalan dari model CB-CTT direpresentasikan dalam matrix  $X_{tp,tr}$  yang memiliki elemen  $x_{i,j}$  dengan  $i$  sebagai timeslot atau periode dan  $j$  sebagai representasi nomor ruangan atau nama kelas.

$$\begin{pmatrix} x_{0,0} & x_{0,1} & \dots & x_{0,tr} \\ x_{1,0} & x_{1,1} & \dots & x_{1,tr} \\ x_{2,0} & x_{2,1} & \dots & x_{2,tr} \\ \dots & \dots & \dots & \dots \\ x_{tp,0} & x_{tp,1} & \dots & x_{tp,tr} \end{pmatrix}$$

Gambar 2.1 Matrix Solusi dari CB-CTT (Bolaji, 2011)

## 2.2 Algoritma *Artificial Bee Colony*

Algoritma *artificial bee colony* terinspirasi dari perilaku lebah madu (Karaboga dkk, 2012). Lebah madu dalam algoritma ini dibagi menjadi 3 kelompok yaitu: *employed bees*, *onlooker bees*, dan *scout bees*. Setiap kelompok lebah mempunyai tugas masing – masing. *Scout bee* bertugas untuk menemukan makanan di sekitar sarang lebah. *Employed bee* bertugas untuk mengambil informasi dari *scout bees* untuk menyimpan informasi dan membagikannya kepada *onlooker bees*. Tugas *onlooker bees* untuk melakukan eksploitasi terhadap sumber makanan. Gambar 2.2 adalah pseudocode algoritma *artificial bee colony*.

```

Initialization Phase
REPEAT
Employed Bees Phase
Onlooker Bees Phase
Scout Bees Phase
Memorize the best solution achieved so far
UNTIL (Cycle = Maximum Cycle Number or a Maximum CPU time)

```

Gambar 2.2 Struktur Umum Algoritma *Artificial Bee Colony*

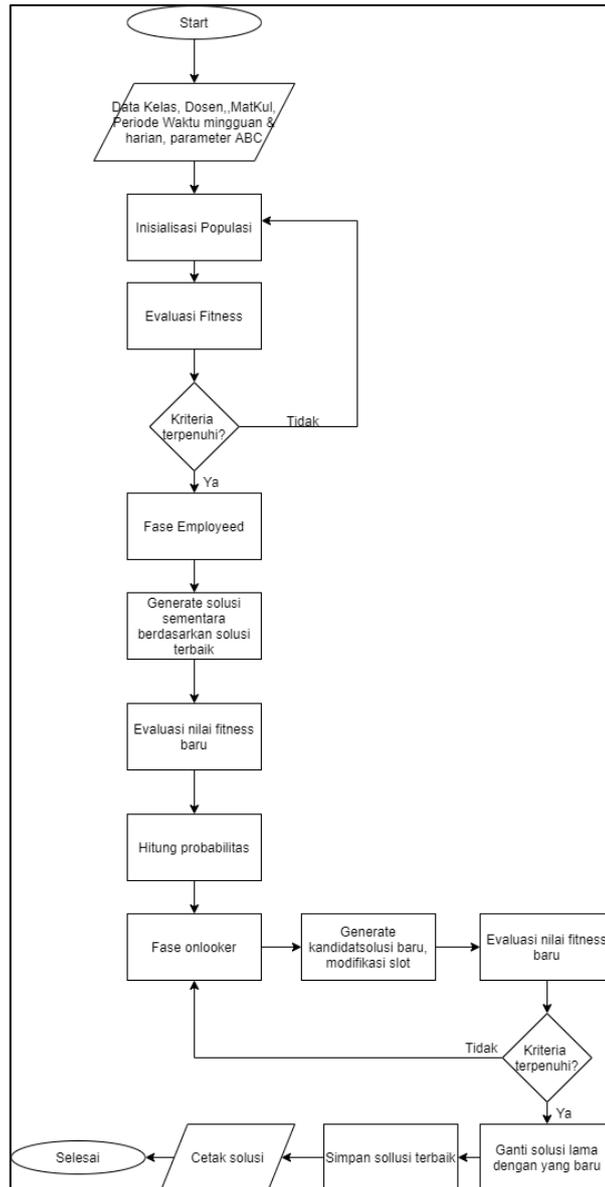
- a. *Scout Bees* merupakan agen lebah yang bertugas mencari posisi sumber makanan di lingkungan sekitar sarang secara acak. Informasi posisi sumber

makanan yang telah ditemukan akan diteruskan ke agen lebah berikutnya, *Employed Bees*.

- b. *Employed Bees* merupakan agen lebah yang berhubungan langsung dengan sumber makanan yang sebelumnya ditemukan oleh *Scout Bees*. Tugas dari *Employed Bees* adalah menyimpan informasi yang berhubungan dengan tiap sumber makanan, baik berupa informasi tentang jarak dan arah dari sarang, informasi tingkat profitabilitas atau kekayaan dari sebuah sumber makanan, maupun nilai kepantasan informasi sumber makanan tersebut untuk disebar luaskan. Oleh karena itu, jumlah dari *Employed Bees* harus ekuivalen dengan jumlah sumber makanan yang ditemukan. Dalam membagikan informasi sumber makanannya, *Employed Bees* melakukan tarian yang bernama *Waggle Dance* di *Dancing Room* yang bertempat pada pusat dari sarang lebah dengan *Onlooker Bees* sebagai yang menonton dan memilih sumber makanan yang akan dieksploitasi.
- c. *Onlooker Bees* merupakan agen lebah yang bertugas untuk memilih dan mengeksploitasi sumber makanan yang informasinya disimpan oleh *Employed Bees*.

### **2.3 Pemodelan Solusi**

Untuk mempermudah perancangan sistem dilakukan pemodelan solusi dalam flowchart pada Gambar 2.3 sistem penjadwalan menggunakan algoritma *artificial bee colony* (Nugroho dan Ayub, 2013).



Gambar 2.3 Flowchart Penjadwalan ABC

### 2.3.1 Tahapan Inisialisasi Populasi

Inisialisasi populasi yang dilakukan dalam algoritma ABC dibentuk secara random. Maka untuk dapat mempercepat proses inisialisasi populasi yang dilakukan dengan sebuah aturan penjadwalan sederhana yang dapat membantu menciptakan populasi awal yang lebih baik. Proses dari penjadwalan sederhananya adalah sebagai berikut:

1. Sistem ini memiliki daftar mata kuliah beserta kuota pembagian waktu mata kuliah masing-masing, dan data dosen sebagai bagian utama dari proses pembuatan jadwal pelajaran.
2. Sistem akan membuat jadwal pelajaran dengan memperhitungkan semua aturan penjadwalan oleh BIA kecuali kasus bentrokan.
3. Setelah seluruh mata kuliah dijadwalkan, maka jadwal pelajaran tersebut akan disimpan dan dihitung nilai *fitnessnya*.

Proses inisialisasi populasi dimulai dengan memasukkan data ada pada buku panduan akademik, seperti jumlah kelas, dosen pengajar, mata kuliah, periode waktu/ timeslot mengajar per minggu, periode waktu/ timeslot perhari dan disimpan ke dalam *database*. Pada tahap ini dihasilkan kandidat jadwal mata kuliah, setiap kandidat akan direpresentasikan dengan nama ruangan, kode jurusan, kode mata kuliah, kode dosen, hari dan slot waktu.

### 2.3.2 Fungsi *Fitness*

Nilai *fitness* didapat jika ada pelanggaran *constraints* seperti yang sudah ditentukan. Semakin kecil nilai *fitness* yang dimiliki oleh suatu solusi, semakin tinggi pula kualitasnya (Nugroho dan Ayub, 2013). Nilai *fitness* yang digunakan sebagai penentu kualitas dari satu solusi ditentukan dari banyaknya jumlah slot yang bentrok. Rumus umumnya dijabarkan sebagai berikut:

$$F(x) = \sum B \quad \dots(2.1)$$

$F = fitness$

$x = index$

$B = slot bentrok$

Contoh penentuan nilai *fitness* sebagai berikut :

	Periode	Senin	Selasa	Rabu	Kamis	Jumat	Sabtu
B320	08.00	IF322	IF205	UM609	UM163	DV555	
	09.00	IF322	IF205	UM609	UM163	DV555	
	10.00	IF322	PR607	UM609	UM163	DV555	
	11.00	UM163	PR607	IS707	IF322	DT655	
	12.00	UM163	PR607	IS707	IF322	DT655	
	13.00	SC256	MC505	FK416	IF322	SI976	
	14.00	SC256	MC505	FK416	AC122	SI967	
	15.00	FK412	SC866	SK707	AC122	IF404	
	16.00	FK412	SC866	SK707	EM932	IF404	
	17.00	FK412	SC866	SK707	EM932	IF404	
18.00							
	Periode	Senin	Selasa	Rabu	Kamis	Jumat	Sabtu
B320	08.00	IF322	IF205	IF303	UM163	DV555	
	09.00	IF322	IF205	IF303	UM163	DV555	
	10.00	IF322	PR607	IF303	UM163	DV555	
	11.00	UM163	PR607	IS707	IF322	DT655	
	12.00	UM163	PR607	IS707	IF322	DT655	
	13.00	SC256	MC505	FK416	IF322	SI976	
	14.00	SC256	MC505	FK416	AC122	SI967	
	15.00	FK412	SC866	SK707	AC122	IF404	
	16.00	FK412	SC866	SK707	EM932	IF404	
	17.00	FK412	SC866	SK707	EM932	IF404	
18.00							

Gambar 2.4 Contoh Solusi Pemodelan *Fitness* Mata Kuliah Bentrok

Gambar 2.4 merupakan contoh model solusi mata kuliah berbeda yang bentrok dalam satu minggu. Kedua mata kuliah bentrok pada hari rabu periode jam 08.00 – 10.00. Dengan menggunakan rumus perhitungan *fitness*, dapat dihitung nilai *fitness* dari Gambar 2.4 adalah 3. Sistem akan melakukan penyimpanan populasi atau kandidat jadwal dengan nilai terkecil sebagai solusi terbaik sementara.

### 2.3.3 Tahap *Employed Bees*

Pada tahap *Employed Bees*, populasi telah terbentuk dan sejumlah sumber makanan/solusi dalam bentuk jadwal pelajaran didapatkan. Sesuai dengan tugas dari agennya, aplikasi akan menyeleksi jadwal – jadwal pelajaran yang memiliki nilai *fitness* lebih baik dibandingkan dengan jadwal lain yang dijadikan perbandingan. Prosesnya pada Gambar 2.5:

Tabel 2. 1 Contoh Tahapan Employeeed Bees

B320	Index	Fitness
	0	3
	1	6
	2	9
	3	10

Tabel 2.1 merupakan gambaran dari proses seleksi dengan contoh 4 kejadian mata kuliah yang bentrok pada satu ruangan dengan nilai fitness masing – masing 3, 6, 9, dan 10. Dari tiap nilai *fitness*, dipilih nilai *fitness* terbaik yakni 3. Kemudian nilai tersebut akan disimpan pada memori untuk dilakukan evaluasi dan seleksi yang berguna untuk memudahkan *Onlooker Bees* dalam menentukan dan memilih solusi mana saja yang memiliki kualitas terbaik dan pantas untuk dieksploitasi.

#### 2.3.4 Tahap *Onlooker Bees* (Eksploitasi)

Eksploitasi dengan metode *random* merupakan metode yang mengupayakan peningkatan kualitas atau perbaikan nilai *fitness* dari jadwal pelajaran dengan cara menukar slot yang didapati bentrok dengan slot lain yang didapatkan dari hasil *me-random* indeks hari dan indeks slot. Usaha peningkatan kualitas jadwal akan terus dilakukan sampai didapatkan jadwal pelajaran dengan *fitness* lebih baik.

#### 2.3.5 *Fase Scout Bee*

Populasi atau kandidat jadwal terbaik yang disimpan akan dilakukan pengecekan apakah telah memenuhi kriteria yang diinginkan. Kriteria terbaik adalah tidak ada jadwal bentrok. Apabila kriteria tidak terpenuhi maka proses akan diulang dimulai dari Fase *onlooker bee*. Apabila kriteria sampai batas akhir iterasi kriteria yang diinginkan tidak diperoleh maka populasi atau kandidat jadwal yang disimpan sebagai yang terbaik akan ditetapkan sebagai solusi terbaik kemudian disimpan.