Penerapan Algoritma *Artificial Bee Colony* dalam Aplikasi Penjadwalan Pelajaran untuk Sekolah Menengah Pertama

Rakhmad Fajar Nugroho[1], Mewati Ayub[2]
[1, 2] Jurusan S1 Teknik Informatika
Fakultas Teknologi Informasi, Universitas Kristen Maranatha.
Jl. Prof. Drg. Suria Sumantri No. 65, Bandung 40164
email: [1]oni.creed@gmail.com, [2]mewatia@yahoo.com

Abstract

Scheduling is a planning for work order with allocation of time and resources to perform assigned operations. The concept of scheduling is usually used for work planning, machine scheduling, and one case which serves as the main focus of this research, is school timetabling. In order for teaching and learning activity to be executed neatly and properly, a school timetable or lesson schedule that is accurate and possesses as minimal flaws as possible is highly necessary. Along with the growth of the school, the increase number of classes, teachers, and students need to be taken into consideration when making a proper school timetable. With those constraints, making a good school timetable will take more time due to its higher complexity. This problem is the idea behind the construction of Scheduling Application for a Junior High School. The application utilizes an optimization algorithm called the Artificial Bee Colony. Artificial Bee Colony algorithm is considered still brand-new to-date. In the implementation, the algorithm represents the clever behaviors of honey bees during their attempts to find favorable food sources, consisted by exploration, evaluation, selection, and exploitation, through communications and activities between each active agent.

Keywords: School timetable, Artificial Bee Colony algorithm.

1. PENDAHULUAN

Penjadwalan merupakan salah satu dari beberapa bagian penting yang tidak dapat dipisahkan dalam sebuah sekolah. Pembuatan sebuah jadwal yang efisien membutuhkan waktu yang lama dan keakuratan karena preferensi, jadwal kegiatan guru, dan periode waktu dalam mengajar sebuah mata pelajaran yang berbeda-beda. Oleh karena itu, sistem penjadwalan yang otomatis dan terkomputerisasi dibutuhkan guna meningkatkan efisiensi kerja dari sekolah yang bersangkutan. Perancangan aplikasi penjadwalannya membutuhkan sebuah algoritma optimasi untuk dijadikan basis, salah satu diantaranya adalah algoritma *Artificial Bee Colony (ABC)*.

Algoritma *Artificial Bee Colony* merupakan algoritma optimasi yang diusulkan oleh Dervis Karaboga pada tahun 2005, yang modelnya diambil dari tingkah laku cerdas lebah madu dalam mencari sumber makanannya. Dengan konsep sederhana yang memiliki ruang besar untuk pengembangan dan teorinya yang cukup mudah untuk dimengerti dan diimplementasikan, algoritma ABC semakin mendapat perhatian dari kalangan peneliti dan telah beberapa kali digunakan dalam beberapa kasus optimasi seperti *Job Shop Scheduling* dan *Travelling Salesman Problem* [3]. Namun seperti algoritma yang berbasiskan *local search* umumnya, penggunaan algoritma ABC erat hubungannya dengan konvergensi yang prematur, stagnasi, dan waktu eksekusi yang lama [6]. Selain itu, terdapat juga kemungkinan bahwa solusi terbaik yang didapat tidak ideal atau tidak sesuai dengan harapan.

Tujuan dari penelitian yang dilakukan dengan menerapkan algoritma ABC pada proses penjadwalan adalah membuat sebuah aplikasi yang dapat menghasilkan jadwal pelajaran dengan kasus bentrok seminimal mungkin. Sumber data dan contoh kasus diperoleh dari sebuah SMP yang bertempat di kota Bandung, Jawa Barat.

2. ALGORITMA *ARTIFICIAL BEE COLONY* (ABC)

2.1 Pengertian Algoritma ABC

Algoritma *Artificial Bee Colony* merupakan satu dari sekian banyak algoritma optimasi yang diadaptasikan dari konsep *Swarm Intelligence* (SI) yang mendapat perhatian dari banyak kalangan peneliti. Algoritma ini pertama diusulkan oleh Dervis Karaboga pada tahun 2005. Seperti diimplikasikan oleh namanya, algoritma ABC merupakan sebuah algoritma yang memodelkan kecerdasan kolektif lebah madu dalam mencari sumber makanan, yang terdiri dari 3 komponen esensial [8], yakni:

1. *Food Sources* (sumber makanan): nilai atau kualitas dari suatu sumber makanan ditentukan oleh jaraknya dengan sarang lebah, banyaknya jumlah makanan, dan kemudahan dalam mengambil makanan tersebut.
2. *Employed Foragers*: merupakan lebah-lebah yang bertugas menyimpan informasi dari sumber makanan yang ditemukan.
3. *Unemployed Foragers,* merupakan lebah-lebah yang bertugas mencari sumber-sumber makanan yang dapat dieksploitasi. Terdapat 2 jenis dari *Unemployed Foragers* :
   a. *Scouts*
   b. *Onlookers*

2.2 Operator Algoritma ABC

Algoritma ABC dimulai dengan sebuah populasi yang terdiri dari kumpulan agen lebah. Berdasarkan tugasnya, agen lebah terbagi menjadi tiga jenis, yaitu:

2.2.1 *Scout Bees*

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.*

2.2.2 *Employed Bees*

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-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 disebarluaskan. 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.

2.2.3 *Onlooker Bees*

Onlooker Bees merupakan agen lebah yang bertugas untuk memilih dan mengeksploitasi sumber makanan yang informasinya disimpan oleh *Employed Bees*.

Secara umumnya, struktur dari algoritma ABC seperti yang dijelaskan sebelumnya, dapat digambarkan seperti pada gambar 1.

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 1 Struktur Umum dari Algoritma ABC

3. DESAIN PERANGKAT LUNAK

3.1 Pemodelan Solusi

Diasumsikan bahwa satu slot periode pengajaran sama dengan 40 menit. Setiap kelas memiliki 7 slot untuk diisikan pelajaran tiap harinya. Namun khusus untuk hari Jumat, jumlah slot yang disediakan hanya 6. Terdapat pula periode istirahat selama 20 menit untuk setiap harinya.

Pemodelan solusi dibuat sedemikian rupa untuk merepresentasikan beberapa hal, yaitu slot periode pengajaran dalam satu hari, mata pelajaran apa yang akan diajarkan pada periode tersebut, siapa guru yang bertugas mengajar, dan kelas apa yang diajar. Pemodelannya adalah sebagai berikut:

a) 1 digit id hari (1 – 6)
b) 1 digit slot periode pengajaran (1 – 7)
c) 2 digit id kelas (1A – 3G)
d) 4 digit id mata pelajaran (MP01 – MP99)
e) 3 digit id guru (G01 – G99)

Hari	Slot	Kelas	Mata Pelajaran	Guru
1	1	1A	MP3	G20

Hari	Slot	Kelas	Mata Pelajaran	Guru
1	2	1A	MP3	G20

Hari	Slot	Kelas	Mata Pelajaran	Guru
1	3	1A	MP5	G17

Hari	Slot	Kelas	Mata Pelajaran	Guru
1	4	1A	MP5	G17

Hari	Slot	Kelas	Mata Pelajaran	Guru
1	5	1A	MP5	G17

Hari	Slot	Kelas	Mata Pelajaran	Guru
1	6	1A	MP13	G11

Hari	Slot	Kelas	Mata Pelajaran	Guru
1	7	1A	MP13	G11

Gambar 2 Contoh Pemodelan Solusi

Gambar 2 merupakan contoh model solusi yang merepresentasikan jadwal pelajaran yang diajar untuk satu kelas dalam satu hari. Pada model ini, kelas 1A memiliki 7 slot, 3 mata pelajaran (MP3, MP5, dan MP13), dan 3 guru berbeda (G20, G17, dan G11) yang bertugas, pada hari ke-1.

3.2 *Constraints*

Dalam melakukan penjadwalan, terdapat beberapa batasan yang tidak boleh dilanggar yang disebut *Hard Constraint*, dan batasan yang alangkah baiknya jika dapat dipenuhi yang disebut *Soft Constraint*.

Hard Constraints

Berikut ini adalah batasan-batasan yang mutlak tidak boleh dilanggar dalam penjadwalan:

a. Masing-masing mata pelajaran tidak boleh di-*assign* ke jadwal melebihi dari kuota slot per-minggunya untuk satu kelas.
b. Mata pelajaran yang diujiankan pada UAN, yakni matematika (6 slot), bahasa indonesia (4 slot), bahasa inggris (4 slot), dan IPA (masing-masing memiliki 3 slot), memiliki jatah slot yang lebih tinggi dibandingkan dengan pelajaran lain yang tidak di-UANkan.
c. Tidak boleh terdapat jadwal di mana satu guru bertugas di dua atau lebih kelas yang berbeda pada slot yang sama dalam satu hari.

Soft Constraints

Selain batasan yang wajib dipenuhi atau *hard constraints,* terdapat juga beberapa batasan yang alangkah baiknya bila dapat dipenuhi. Salah satu diantaranya adalah beberapa mata pelajaran yang hanya memiliki 2 slot tidak boleh dibagi antar slotnya dalam dua hari yang berbeda atau 2 slot yang tidak berurutan.

3.3 Inisialisasi Populasi (Eksplorasi)

Pada teorinya, inisialisasi populasi yang dilakukan dalam algoritma ABC dibentuk secara *random*, akibatnya populasi tersebut memiliki nilai *fitness* yang rendah atau kurang baik. Maka untuk dapat mempercepat proses inisialisasi populasi yang dilakukan pada aplikasi ini tidaklah dilakukan secara *random*, melainkan dilakukan dengan sebuah aturan penjadwalan sederhana yang dapat membantu menciptakan populasi awal yang lebih baik.

Proses dari penjadwalan sederhananya adalah sebagai berikut:

1. Aplikasi ini memiliki daftar mata pelajaran beserta kuota slotnya masing-masing, dan data guru sebagai bagian utama dari proses pembuatan jadwal pelajaran.
2. Aplikasi akan membuat jadwal pelajaran dengan memperhitungkan semua *hard constraints* kecuali kasus bentrokan, ditambah dengan *soft constraint* yang menjabarkan bahwa ada beberapa mata pelajaran yang lebih baik bila antar slotnya di-*assign* berurutan.
3. Setelah seluruh mata pelajaran dijadwalkan, maka jadwal pelajaran tersebut akan disimpan dan dihitung nilai *fitnessnya*.

3.4 Fungsi *Fitness*

Semakin kecil nilai *fitness* yang dimiliki oleh suatu solusi, semakin tinggi pula kualitasnya. Nilai *fitness* yang digunakan sebagai penentu kualitas dari satu solusi ditentukan dari banyaknya jumlah slot yang bentrok. Rumus umumnya dijabarkan sebagai berikut:

$$f(x) = \sum B$$

f = *Fitness*
x = Indeks Jadwal Pelajaran
B = Slot yang bentrok

Contoh penentuan nilai *fitness* sebagai berikut :

	Slot	Senin	Selasa	Rabu	Kamis	Jumat	Sabtu
1A	0	MP1	MP3	MP6	MP7	MP9	MP4
	1	MP1	MP3	MP6	MP7	MP8	MP4
	2	MP3	MP2	MP6	MP9	MP8	MP4
	3	MP3	MP2	MP6	MP9	MP12	MP13
	4	MP3	MP4	MP6	MP6	MP12	MP13
	5	MP5	MP10	MP2	MP11	MP12	MP14
	6	MP5	MP10	MP2	MP11		MP14

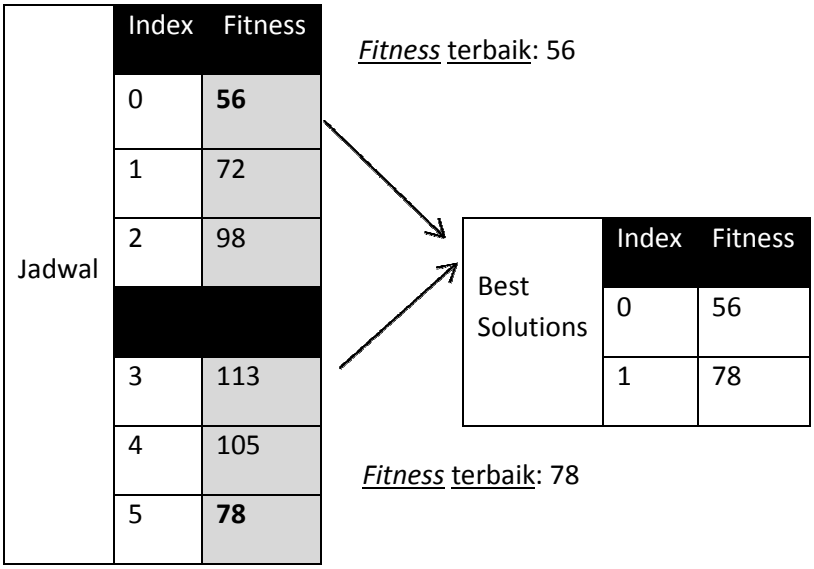
	Slot	Senin	Selasa	Rabu	Kamis	Jumat	Sabtu
1B	0	MP3	MP14	MP6	MP7	MP4	MP8
	1	MP3	MP14	MP6	MP7	MP4	MP11
	2	MP1	MP13	MP6	MP3	MP4	MP3
	3	MP1	MP12	MP2	MP9	MP10	MP3
	4	MP6	MP12	MP2	MP9	MP10	MP4
	5	MP6	MP12	MP2	MP5	MP8	MP13
	6	MP6	MP2	MP11	MP5		MP9

Gambar 3 Contoh Solusi untuk Penentuan *Fitness*

Gambar 3 merupakan contoh model solusi jadwal pelajaran dua kelas yang berbeda dalam satu minggu. Slot pelajaran yang bentrok ditandai dengan warna merah, seperti MP6 pada hari Rabu, slot ke-0, ke-1, dan ke-2 yang bentrok dengan slot dan hari yang sama pada kelas 1B. Dengan mengunakan rumus perhitungan *fitness*, dapat dihitung nilai *fitness* untuk contoh jadwal pada Gambar 3 adalah 7 (tujuh), ekuivalen dengan jumlah slot yang bentrok pada jadwal tersebut.

3.5 Tahap *Employed Bees* (Evaluasi dan Seleksi)

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 digambarkan pada gambar 4:



Gambar 4 Gambaran tahap *Employed Bees*

Gambar 4 merupakan gambaran dari proses seleksi dengan contoh enam jadwal pelajaran yang memiliki nilai *fitness* masing-masing 56, 72, 98, 113, 105, dan 78. Dalam penyeleksiannya, jadwal yang didapat diambil masing-masing tiga indeks sehingga didapatkan dua grup, 56, 72, 88 dan 113, 105, 78. Dari masing-masing grup tersebut, dipilih jadwal yang memiliki *fitness* terbaik, yakni 56 untuk grup satu, dan 78 untuk grup dua. Jadwal dengan *fitness* 56 dan 78 yang telah lulus seleksi akan disimpan di memori. Proses evaluasi dan seleksi ini

berguna untuk memudahkan *Onlooker Bees* dalam menentukan dan memilih solusi mana saja yang memiliki kualitas terbaik dan pantas untuk dieksploitasi.

3.6 Tahap *Onlooker Bees* (Eksploitasi)

Tahap ini dapat dibedakan menjadi dua macam , yaitu eksploitasi secara *random* atau secara *sequential.*

Eksploitasi dengan Metode *Random*

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.

Eksploitasi dengan Metode *Sequential*

Eksploitasi dengan metode *sequential* merupakan metode yang mengupayakan peningkatan kualitas atau perbaikan nilai fitness dari jadwal pelajaran dengan cara menukar slot yang didapati bentrok dengan semua slot pelajaran yang terdapat di kelas tersebut secara berurut dan bergiliran. Usaha peningkatan kualitas jadwal akan terus dilakukan sampai didapatkan jadwal pelajaran dengan fitness lebih baik atau semua slot pada jadwal tersebut sudah dicoba ditukar dengan slot yang bentrok, namun tidak menghasilkan jadwal yang lebih baik.

3.7 Perbaikan Jadwal

Bilamana masih terdapat bentrokan pada jadwal terbaik yang didapat pada akhir iterasi, aplikasi akan melakukan proses perbaikan jadwal dengan cara menggeser pelajaran yang bentrok ke semua slot pelajaran yang terdapat di kelas tersebut secara berurutan seperti eksploitasi dengan metode *sequential*. Dengan demikian, jumlah bentrokan dapat dikurangi bilamana memungkinkan. Namun jika ternyata solusi yang lebih baik masih tidak ditemukan, maka pelajaran tersebut akan dibiarkan bentrok.

3.8 Data Input

Tabel input yang diberikan untuk aplikasi berupa 3 buah *file* excel, yaitu:

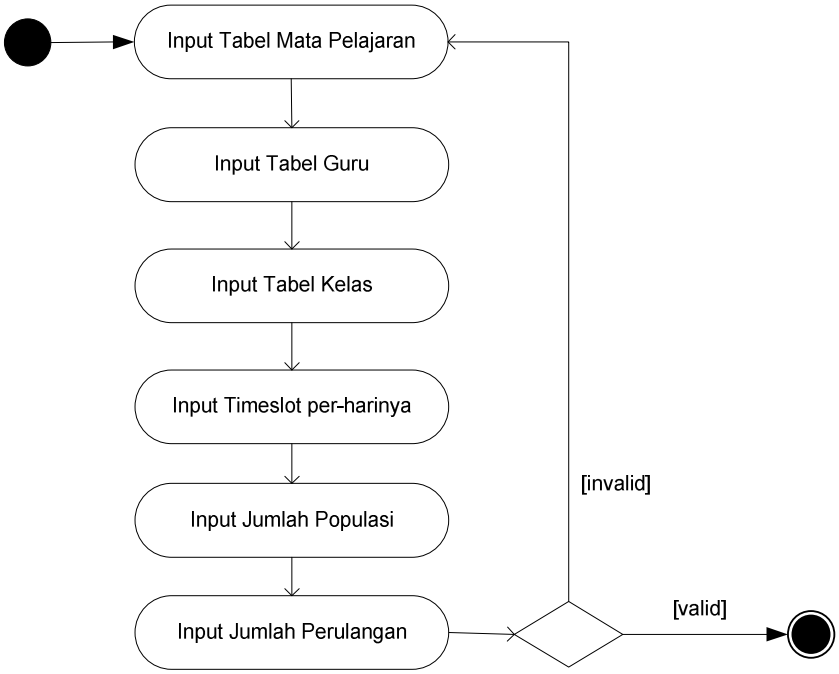
a. Tabel mata pelajaran dan masing masing jumlah slotnya.
b. Tabel guru beserta mata pelajaran yang diajar.
c. Tabel kelas.

Sedangkan input yang dijadikan konfigurasi pada aplikasinya adalah :

a. Jumlah *timeslot* per hari: banyaknya *timeslot* yang disediakan dalam satu hari.
b. Jumlah populasi: jumlah *employed bees* dan sumber makanan dalam satu populasi.
c. Jumlah perulangan: banyaknya perulangan fungsi ABC akan dilakukan.

4. IMPLEMENTASI

4.1 Aktivitas Input Parameter



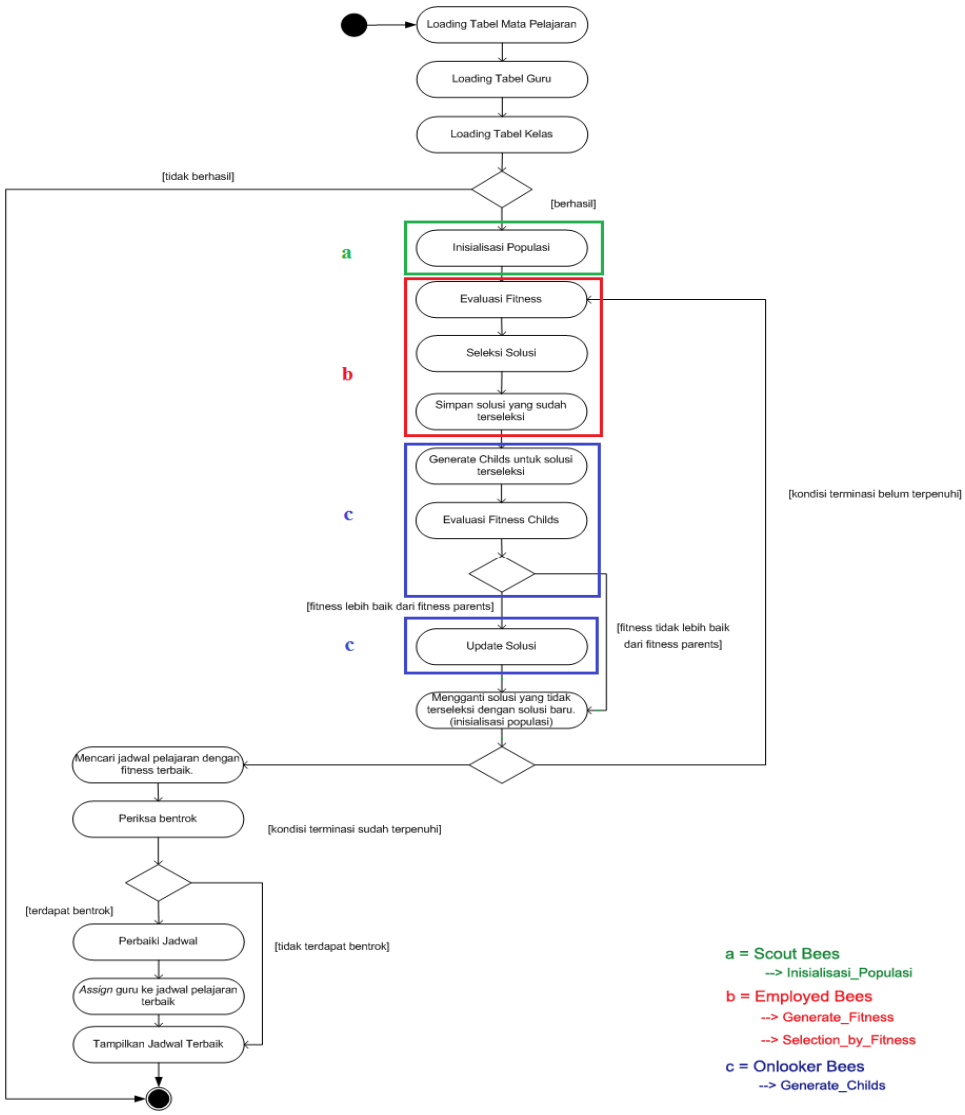
Gambar 5 *Activity Diagram* Input Parameter

Pada aktivitas ini dilakukan input beberapa parameter dan alamat *file* excel yang akan di-*export*. Agar aplikasi dapat membuat jadwal pelajaran yang *valid*, inputan seperti tabel guru, tabel kelas, dan tabel mata pelajaran tidak boleh kosong atau *null,* dan jumlah *timeslot* untuk satu minggu juga tidak boleh melebihi jumlah slot mata pelajaran pada tabel mata pelajaran yang diinput. Parameter-parameter dan alamat-alamat *file* excel ini seterusnya akan digunakan sebagai acuan dalam proses penyusunan penjadwalan.

4.2 Aktivitas Menyusun Jadwal

Berikut ini akan dijelaskan mengenai aktivitas penyusunan jadwal yang terdapat pada Gambar 6:

1. Proses penjadwalan dimulai dengan memasukkan data-data yang didapat dari *file* excel yang diimport ke dalam struktur data masing-masing.
2. Setelah seluruh proses *load* berhasil, maka akan didapatkan sejumlah *list* yang mewakili masing-masing data-data tersebut untuk kemudian dijadikan sebuah jadwal pelajaran.
3. Langkah ke-2 akan terus berulang sampai jumlah jadwal pelajaran yang diinisialisasi sudah ekuivalen dngan jumlah populasi yang diinput.
4. Nilai *fitness* masing-masing jadwal yang ada akan diperiksa dan dievaluasi untuk kemudian diseleksi. Jadwal pelajaran yang memiliki nilai *fitness* lebih akan dibawa ke proses berikutnya, sedangkan jadwal pelajaran yang tidak lulus seleksi akan diabaikan.
5. Pada bagian *Generate Childs* untuk solusi terseleksi, atau dapat juga disebut sebagai proses eksploitasi dalam pembuatan jadwal ini; masing-masing solusi tersebut akan di-tes kualitasnya dengan diadakannya perbandingan antara jadwal tersebut dengan jadwal yang sama, namun dengan beberapa slot yang sudah dimodifikasi (*child*). Bilamana *fitness* dari *child* terhitung lebih baik dibandingkan dengan *fitness* sebelumnya (*parent*), maka aplikasi akan menyimpan jadwal pelajaran child beserta *fitness*nya untuk menggantikan jadwal pelajaran *parent*. Namun bilamana *fitness child* tidak lebih baik dari *fitness parent,* aplikasi tidak akan melakukan *update*.
6. Aplikasi akan menginisialisasi jadwal-jadwal pelajaran baru untuk menggantikan jadwal pelajaran yang sebelumnya diabaikan atau tidak lulus seleksi.
7. Langkah-langkah sebelumnya akan terus diulang sampai dengan terpenuhinya kondisi terminasi, seperti kondisi di mana jumlah bentrokan pada jadwal terbaik adalah 0 (nol) atau jumlah perulangan yang telalu dilalui aplikasi ekuivalen dengan jumlah perulangan yang diinput.
8. Bilamana *fitness* dari jadwal pada akhir iterasi sudah memenuhi kondisi optimal, maka jadwal tersebut akan ditampilkan sebagai jadwal yang terpilih. Namun jika *fitness*nya atau jumlah bentrokan pada jadwal pelajaran tersebut tidak merupakan 0 (nol), maka aplikasi akan melakukan perbaikan jadwal untuk mengoptimalkan jadwal tersebut. Bilamana kondisi optimal tetap tidak terpenuhi, maka jadwal tersebut akan dibiarkan bentrok.



Gambar 6 *Activity Diagram* Menyusun Jadwal

5. PENGUJIAN

5.1 Pengujian Besar Populasi dan Jumlah Generasi

Pada tabel 1 ditunjukkan hasil pengujian yang bertujuan mendapatkan nilai *fitness* inisial untuk besar populasi yang berbeda-beda.

Tabel 1 Pengujian Besar Populasi dan Jumlah Generasi

No	Jumlah Kelas	Jumlah Mata Pelajaran	Jumlah Slot/Minggu	Populasi	Jumlah Generasi	Waktu (menit)	Fitness
1	21	15	41	30	-	1	378
2	21	15	41	99	-	1	355
3	21	15	41	300	-	1	338
4	21	15	41	600	-	1	323
5	21	15	41	999	-	1	315

5.2 Pengujian Perkembangan Generasi dengan Fungsi ABC

Pada pengujian berikut akan dilakukan dua percobaan untuk masing-masing metode eksploitasi dengan input 6 dan 15 kelas, besar populasi dengan nilai 99, dan 1000 kali perulangan. Dikarenakan seleksi antar solusi dilakukan dengan membandingkan masing-masing tiga solusi yang terdapat pada populasi, besar populasi yang diinput harus merupakan kelipatan tiga.

5.2.1 Pengujian Eksploitasi dengan Metode *Random*

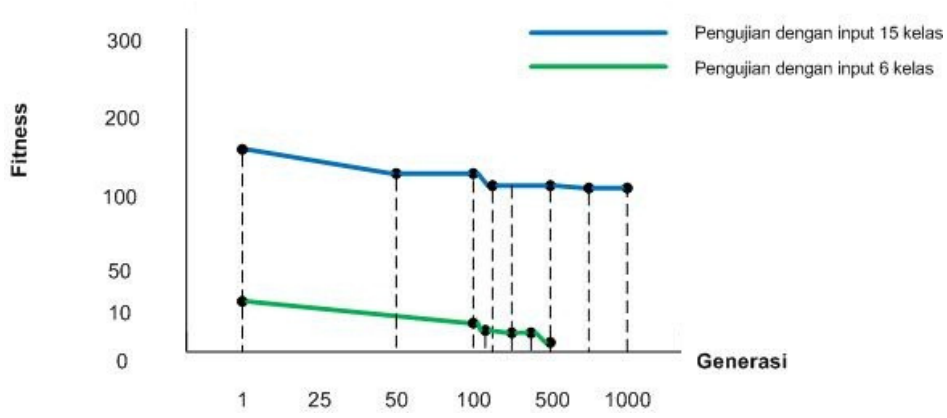
Tabel 2 Pengujian dengan Fungsi ABC Eksploitasi *Random* – 6 Kelas

No	Jumlah Kelas	Jumlah Mata Pelajaran	Jumlah Slot/Minggu	Besar Populasi	Generasi	Waktu (menit)	Fitness
1	6	15	41	99	1	1	10
2	6	15	41	99	100	1	8
3	6	15	41	99	200	1	4
4	6	15	41	99	300	1	4
5	6	15	41	99	400	1	4
6	6	15	41	99	500	1	2

Tabel 3 Pengujian dengan Fungsi ABC Eksploitasi *Random* – 15 Kelas

No	Jumlah Kelas	Jumlah Mata Pelajaran	Jumlah Slot/Minggu	Besar Populasi	Generasi	Waktu (menit)	Fitness
1	15	15	41	99	1	1	166
2	15	15	41	99	50	5	136
3	15	15	41	99	100	9	136
4	15	15	41	99	250	23	127
5	15	15	41	99	500	45	127
6	15	15	41	99	750	70	123
7	15	15	41	99	1000	93	123

Dengan menggunakan hasil pengujian eksploitasi dengan metode *Random* yang ditunjukkan pada tabel 2 dan tabel 3, dapat dibuat sebuah grafik untuk membandingkan perubahan *fitness* yang terjadi seiring bertambahnya jumlah generasi pada pengujian yang telah dilakukan. Grafiknya digambarkan sebagai berikut:



Gambar 7 Grafik Perkembangan Jadwal per-Generasi Metode *Random*

5.2.2 Pengujian Eksploitasi dengan Metode *Sequential*

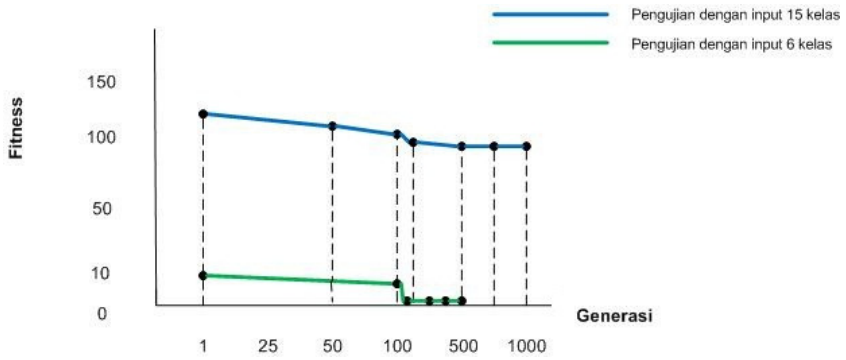
Tabel 4 Pengujian dengan Fungsi ABC Eksploitasi *Sequential* – 6 Kelas

No	Jumlah Kelas	Jumlah Mata Pelajaran	Jumlah Slot/Minggu	Besar Populasi	Generasi	Waktu (menit)	Fitness
1	6	15	41	99	1	1	9
2	6	15	41	99	100	1	6
3	6	15	41	99	200	1	1
4	6	15	41	99	300	1	1
5	6	15	41	99	400	2	1
6	6	15	41	99	500	2	1

Tabel 5 Pengujian dengan Fungsi ABC Eksploitasi *Sequential* – 15 Kelas

No	Jumlah Kelas	Jumlah Mata Pelajaran	Jumlah Slot/Minggu	Besar Populasi	Generasi	Waktu (menit)	Fitness
1	15	15	41	99	1	1	152
2	15	15	41	99	50	15	132
3	15	15	41	99	100	31	127
4	15	15	41	99	250	65	124
5	15	15	41	99	500	140	123
6	15	15	41	99	750	210	123
7	15	15	41	99	1000	280	123

Dengan menggunakan hasil pengujian eksploitasi dengan metode *Sequential* yang ditunjukkan pada tabel 4 dan tabel 5, dapat dibuat sebuah grafik untuk membandingkan perubahan *fitness* yang terjadi seiring bertambahnya jumlah generasi pada pengujian yang telah dilakukan. Grafiknya digambarkan sebagai berikut:



Gambar 8 Grafik Perkembangan Jadwal per-Generasi Metode *Sequential*

6. SIMPULAN DAN SARAN

6.1 Simpulan

Berdasarkan hasil analisis yang dilakukan dalam pembuatan aplikasi penjadwalan dengan menggunakan algoritma ABC untuk kasus jadwal pelajaran SMP, dapat diambil kesimpulan sebagai berikut:

1. Telah berhasil dibuat aplikasi yang mampu menghasilkan jadwal pelajaran dengan menggunakan algoritma *Artificial Bee Colony* (ABC) untuk mengurangi jumlah bentrokan.
2. Perbandingan hasil pengujian eksploitasi dengan metode *random* dan hasil pengujian eksploitasi dengan metode *sequential* (contoh perbandingan antara Tabel 2 dan Tabel 4), menunjukkan bahwa walaupun eksploitasi *sequential* menghabiskan waktu yang lebih lama dibandingkan dengan eksploitasi *random*, eksploitasi *sequential* mampu mendapatkan jadwal pelajaran dengan *fitness* setara dengan jadwal pelajaran terbaik dari pengujian eksploitasi *random* dengan jumlah generasi atau perulangan yang lebih sedikit.
3. Eksploitasi dengan metode *sequential* menghabiskan waktu eksekusi yang lebih lama dibandingkan eksploitasi dengan metode *random* dikarenakan kompleksitas algoritma metode *sequential* yang lebih tinggi.
4. Pada saat menggunakan aplikasi, perubahan parameter-parameter seperti besar populasi dan jumlah generasi akan memberikan pengaruh yang signifikan terhadap performa dari aplikasi. Dari uji coba yang dilakukan, diperoleh hasil bahwa lebih baik memperbesar jumlah generasi dibandingkan dengan memperbanyak jumlah populasi untuk mendapat solusi paling optimal.
5. Algoritma ABC sangat dipengaruhi oleh fungsi *random* yang digunakan, terutama untuk penerapan eksploitasi dengan metode *random*, sehingga hasil yang didapatkan dari beberapa pengujian tidak akan sama persis walaupun parameter yang diinput tidak berbeda.

6.2 Saran

Beberapa saran untuk penyempurnaan aplikasi penjadwalan pelajaran untuk SMP dengan menggunakan algoritma ABC adalah sebagai berikut :

1. Perlu dilakukan optimasi terhadap fungsi ABC, terutama pada operator seleksi dan eksploitasi secara menyeluruh sehingga sistem dapat menghasilkan penjadwalan yang paling optimal dalam waktu eksekusi yang lebih cepat.
2. Fungsi *fitness* perlu dapat mempertimbangkan tidak hanya *hard constraints*, namun juga *soft constraints* sebagai berikut:

a. Penyusunan jadwal pelajaran memperhatikan seberapa lama seorang guru mengajar pada satu harinya agar tidak terlalu banyak.
b. Kecuali untuk hari Jumat dan Sabtu; dalam satu hari, minimal terdapat 3 mata pelajaran yang berbeda pada satu kelas.
c. Satu mata pelajaran tidak boleh menghabiskan lebih dari jatah 3 slot untuk satu kelas dalam satu hari.

DAFTAR PUSTAKA

[1] E, Bonabeu, Dorigo M, Theraulaz G. 1999. *Swarm Intelligence: From Natural to Artificial Systems*. New York, NY: Oxford University Press Inc.

[2] J. Li, Q. Pan, S. Xie, S. Wang. 2011. *A Hybrid Artificial Bee Colony Algorithm for Flexible Job Shop Scheduling Problems*. China: School of Computer, Liaocheng University.

[3] Karaboga, Dervis, Beyza Gorkemli, Celal Ozturk, Nurhan Karaboga. 2012. *A Comprehensive Survey: Artificial Bee Colony (ABC) Algorithm and Applications*. Turkey: Intelligent Systems Research Group, Engineering Faculty, and Erciyes University.

[4] Knuth, Donald E. 1998. The Art of Computer Programming, Volume 2: Seminumerical Algorithms. 3rd Edition. Reading: Addison-Wesley.

[5] Pilon, Dan, Neil Pitman. 2005. *UML 2.0 in a Nutshell*. Gravenstein Highway North: O'Reilly Media.

[6] Prakash, Divya. 2012. *Bespoke Artificial Bee Colony Algorithm to Determine the Earthquake locations*. India: Siet Gangoh.

[7] Rui, Zhang, Cheng Wu. 2011. *An Artificial Bee Colony Algorithm for the Job Shop Scheduling Problem with Random Processing Times*. China: School of Economics and Management, Nanchang University.

[8] Seeley TD. 1995. *The Wisdom of the Hive. Harvard University Press*. Cambridge: MA.

[9] V. Tereshko, A. Loengarov. 2005. Collective Decision-Making in Honey Bee Foraging Dynamics. Scotland: University of Paisley.

[10] Widjaja, Andi Irvan, Mewati Ayub, Tjatur Kandaga. 2010. Penjadwalan Sidang Otomatis dengan Menggunakan Algoritma Genetik. Bandung: Fakultas Teknologi Informasi Universitas Kristen Maranatha.

[11] Yan, Gaowei, Li Chuangqin. 2011. *An Effective Refinement Artificial Bee Colony Optimization Algorithm Based On Chaotic Search and Application for PID Control Tuning.* China: Taiyuan University of Technology.