

Implementasi Algoritma Genetika Pada Penempatan Tugas Asisten Laboratorium Berbasis Web (Studi Kasus: LAB ICT Universitas Multimedia Nusantara)

Adhi Kusnadi, David Setyadi Santoso
Program Studi Teknik Informatika, Universitas Multimedia Nusantara, Tangerang, Indonesia
Adhi.kusnadi@umn.ac.id

Diterima 4 Desember 2015
Disetujui 30 Desember 2015

Abstract— On education system in University, the given subject is not only theoretics subject, but practicum also. In teaching practical subjects, the lecturer often experiences a difficulty to answer and handle every question that the students have in laboratory. Therefore, the assistant laboratory is needed here. One of the division in Universitas Multimedia Nusantara, ICT LAB still arranges the assignment of the assistant laboratory manually. It is difficult and takes some times. This thing becomes the reason for the development of an application ICT LAB that can do the assignment process automatically. This research is using Genetic algorithm that can calculate fitness value which show the matching level of the assignment that has been created. Based on the testing, using one point crossover, mutation on bit level, and rank selection, the average fitness value is 68.571% from all of the laboratory assistant vacancy. Based on the data used, there are 22 assignment slots on certain subjects that don't have any applicant. So if that 22 slots are not used in the fitness calculation, there are only 118 assignment slot and the fitness value becomes 80,5%. This fitness value has problem with the quantity of student that didn't fit to all of the assignment slot.

Index Terms— Genetic algorithm, rank selection, one point crossover, assignment, laboratory assistant

I. Pendahuluan

Pada sistem pendidikan di universitas, materi perkuliahan yang disampaikan tidak hanya berupa materi teori saja, namun juga materi praktikum. Materi praktikum bertujuan agar mahasiswa dapat menerapkan secara langsung teori yang

telah diajarkan sebelumnya [1].

Dalam mengajarkan materi praktikum, sering kali dosen mengalami kesulitan untuk memenuhi setiap pertanyaan dan menangani masalah yang dihadapi oleh mahasiswa di laboratorium. Maka dari itu asisten laboratorium dibutuhkan untuk dapat membantu dosen sebagai perpanjangan tangan dalam menjawab setiap pertanyaan dari mahasiswa [2].

Bapak Yustinus Eko sebagai Kepala Laboratorium ICT UMN mengatakan bahwa penempatan tugas asisten laboratorium pada fakultas ICT UMN dilakukan dengan dua cara. Cara yang pertama adalah penempatan mahasiswa sebagai asisten laboratorium dilakukan setelah jadwal untuk setiap mata kuliah keluar. Hal ini mudah untuk dilakukan karena mahasiswa belum mengisi KRS sehingga pihak ICT LAB dapat menempatkan calon asisten laboratorium tanpa resiko bentrok dengan jadwal mata kuliah yang diambil oleh calon asisten laboratorium tersebut. ICT LAB dapat memberi informasi ke calon asisten laboratorium agar tidak mengambil mata kuliah yang bentrok dengan jadwal penempatan asisten laboratorium. Cara yang kedua adalah penempatan tugas asisten laboratorium dilakukan setelah calon asisten laboratorium telah melakukan pengisian KRS karena jadwal matakuliah terlambat muncul sehingga ICT LAB tidak sempat untuk melakukan penempatan asisten laboratorium. Hal ini menyebabkan ICT LAB harus menempatkan tugas asisten laboratorium tanpa bentrok dengan jadwal matakuliah yang diambil mahasiswa. Penempatan tugas ini masih dilakukan secara manual sehingga pihak ICT LAB mengalami kesulitan dan membutuhkan waktu yang cukup lama, karena banyaknya matakuliah praktikum. Seperti pada jurusan teknik informatika dengan 17 mata kuliah yang ada praktikumnya dikali jumlah kelas yang dibuka, belum lagi dengan jurusan lain seperti SI, SK dan jurusan pada

fakultas lain seperti Manajemen, akuntansi dan DKV yang mengambil mata kuliah yang berhubungan dengan komputer, yaitu mata kuliah Pengantar Teknologi Multimedia.

Hal itu menjadi alasan pembuatan aplikasi yang dapat membantu ICT LAB dalam hal penempatan tugas asisten laboratorium secara otomatis. Sebelumnya sudah ada penelitian lain yang menerapkan algoritma genetika pada masalah penjadwalan mata kuliah pada jurusan Sistem Informasi di ITS. Pada penelitian tersebut penjadwalan dilakukan bergantung pada ketersediaan waktu dosen untuk menentukan jadwal mata kuliah yang berubah-ubah dengan fleksibel [3]. Terdapat pula sebuah penelitian yang dilakukan oleh Mahmudy [4] yang membahas tentang optimasi model penugasan dengan menerapkan algoritma genetika dengan persilangan satu titik yang dilakukan secara *random*. Pemilihan algoritma genetika, karena pada penelitian sebelumnya yang disebutkan diatas, algoritma genetika berhasil digunakan dalam menyusun jadwal dan disebutkan juga bahwa algoritma genetika cocok untuk digunakan pada masalah optimasi.

II. Landasan Teori

A. Algoritma Genetika

Algoritma genetika adalah suatu algoritma pencarian yang meniru mekanisme dari genetika alam. Algoritma genetika dikenal sejak 1975 dan sudah banyak dipakai pada aplikasi bisnis, teknik maupun pada bidang keilmuan. Algoritma ini ditemukan di Universitas Michigan, Amerika Serikat oleh John Holland dan dipopulerkan oleh salah satu muridnya, David Goldberg. Goldberg mendefinisikan algoritma genetika ini sebagai metode algoritma pencarian berdasarkan pada mekanisme seleksi dan genetika alam [5].

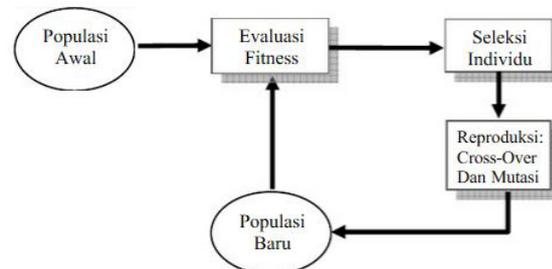
Sebelum algoritma dijalankan, masalah yang akan dioptimalkan itu harus dinyatakan dalam sebuah fungsi *fitness*. Pendekatan yang diambil oleh algoritma ini adalah menggabungkan secara acak berbagai pilihan solusi terbaik di dalam suatu kumpulan untuk mendapatkan generasi solusi terbaik berikutnya yaitu pada suatu kondisi yang memaksimalkan kecocokannya atau lazim disebut *fitness*. Jika nilai *fitness* semakin besar, maka sistem yang dihasilkan semakin baik. Pada awalnya semua nilai *fitness* kemungkinan sangat kecil karena algoritma ini menghasilkannya secara *random*, namun sebagian akan lebih tinggi dari yang lain. Kromosom dengan nilai *fitness* yang tinggi akan memberikan probabilitas yang tinggi untuk bereproduksi pada generasi selanjutnya. Untuk setiap generasi pada proses evolusi, fungsi *fitness* yang mensimulasikan seleksi alam, akan menekan populasi ke arah *fitness* yang meningkat [6].

Beberapa definisi penting yang perlu diperhatikan untuk membangun penyelesaian permasalahan dengan algoritma genetika adalah sebagai berikut [7].

- Genotype* (Gen), sebuah nilai yang menyatakan satuan dasar yang membentuk suatu arti tertentu dalam satu kesatuan gen yang dinamakan kromosom. Dalam algoritma genetika, gen ini bisa berupa nilai biner, *float*, *integer* maupun karakter, atau kombinatorial.
- Allele*, nilai dari gen.
- Kromosom, gabungan gen-gen yang membentuk nilai tertentu.
- Individu, menyatakan satu nilai atau keadaan yang menyatakan salah satu solusi yang mungkin dari permasalahan yang diangkat.
- Populasi, merupakan sekumpulan individu yang akan diproses bersama dalam satu siklus proses evolusi.
- Generasi, menyatakan satu siklus proses evolusi atau satu iterasi di dalam algoritma genetika.

Untuk menggunakan algoritma genetika, solusi permasalahan direpresentasikan sebagai kromosom. Tiga aspek yang penting untuk penggunaan algoritma genetika [8].

- Definisi fungsi *fitness*.
- Definisi dan implementasi representasi genetika.
- Definisi dan implementasi operasi genetika.



Gambar 1 Alur Kerja Algoritma Genetika

Pada dasarnya proses algoritma genetika terdiri dari tahapan-tahapan yang akan dijelaskan sebagai berikut.

a. Representasi Gen

Tahap awal dalam proses algoritma genetika adalah membuat representasi dari tiap-tiap individu yang akan ikut dalam siklus algoritma genetika.

b. Pembangkitan Populasi Awal

Langkah berikutnya adalah membentuk sebuah populasi untuk sejumlah gen. Populasi adalah sekumpulan individu yang akan digunakan dalam setiap proses regenerasi dimana masing-masing individu terdiri dari beberapa gen. Untuk itu diperlukan suatu populasi awal yang digunakan untuk proses penentuan individu terbaik. Solusi atau individu terbaik dari populasi awal akan dipertahankan sedangkan individu-individu yang lain akan diubah menjadi variasi lainnya untuk memperoleh kemungkinan solusi yang lebih baik dari pada solusi sebelumnya.

c. Evaluasi Nilai *Fitness*

Pada setiap populasi baru yang terbentuk baik dari populasi awal maupun dari proses regenerasi akan dihitung nilai *fitness cost* dari setiap individu dalam populasi. *Fitness cost* merupakan nilai kualitas dari suatu individu.

d. Seleksi

Seleksi dilakukan dalam rangka untuk mendapatkan induk yang baik. Seleksi bertujuan untuk memberikan kesempatan reproduksi yang lebih besar bagi anggota populasi yang paling baik. Karena induk yang baik akan menghasilkan keturunan yang baik. Semakin tinggi nilai *fitness* suatu individu maka akan semakin besar pula kemungkinannya untuk terpilih. Ada beberapa metode yang dapat digunakan untuk memilih kromosom antara lain seleksi roda *roulette* (*Roulette Wheel Selection*), seleksi *rank* (*Rank Selection*), dan seleksi turnamen (*Tournament Selection*) [9].

e. Persilangan (*Crossover*)

Crossover adalah operator dari algoritma genetika yang melibatkan dua induk yang

membentuk individu baru. Operator *crossover* ini bergantung pada representasi kromosom yang dilakukan. Berbagai model *crossover* sesuai dengan representasi adalah *crossover* satu titik, dua titik, dan *order crossover*.

f. Mutasi

Mutasi merupakan proses mengubah nilai dari satu atau beberapa gen dalam suatu kromosom. Mutasi ini berperan untuk menggantikan gen yang hilang dari populasi akibat proses seleksi yang memungkinkan munculnya kembali gen yang tidak muncul pada inisialisasi populasi. Mutasi ditetapkan dengan probabilitas sangat kecil. Jika mutasi dilakukan terlalu sering, maka akan menghasilkan individu yang lemah karena konfigurasi bit pada kromosom yang unggul akan dirusak. Mutasi ini bukanlah operator yang utama, yang dilakukan secara acak pada gen dengan kemungkinan yang kecil itu. Berdasarkan bagian yang termutasi, mutasi dapat dibedakan atas tiga bagian, yaitu tingkat kromosom, gen dan bit [10].

g. Pengulangan

Setelah proses regenerasi selesai, maka dilakukan pengulangan sampai sejumlah generasi yang dikehendaki. Gen dari generasi sebelumnya digantikan posisinya dengan generasi yang baru. Individu yang diperoleh dari proses mutasi dan *crossover* dianggap sebagai populasi awal lagi.

B. Penugasan

Masalah penugasan merupakan masalah penjadwalan sumberdaya dan aktifitas berdasarkan penugasan satu-ke-satu, sedemikian sehingga dapat menghemat total biaya yang dibutuhkan. Masalah ini merupakan masalah permutasi suatu himpunan obyek dan termasuk *NP hard Problem*. Dalam pemecahannya, banyak sumberdaya dianggap sama dengan banyak aktifitas. Jika salah satu mempunyai jumlah yang berlebih, maka yang lain harus ditambah agar banyaknya sama. Penjadwalan setiap sumberdaya pada suatu aktifitas harus dilakukan untuk mendapatkan total biaya minimum sedemikian sehingga semua aktifitas dapat diselesaikan [11].

III. Metodologi dan Perancangan Sistem

Metodologi dan perancangan sistem dalam penelitian ini adalah sebagai berikut.

A. Metodologi

Metode yang digunakan dalam penelitian ini adalah sebagai berikut.

1. Wawancara

Melakukan wawancara untuk mengetahui permasalahan secara lebih mendalam dan mendapatkan informasi yang berkaitan dengan penugasan asisten laboratorium.

2. Studi Pustaka

Melakukan studi terhadap sumber-sumber seperti jurnal, laporan, dan artikel yang telah dipublikasikan oleh peneliti lain yang berkaitan dengan *scheduling*, *penugasan*, algoritma genetika, dan lain-lain yang berguna untuk membangun aplikasi ini.

3. Perancangan

Melakukan perancangan aplikasi penugasan asisten laboratorium dengan mengimplementasikan algoritma genetika ke dalam kode program berbasis *web* disesuaikan dengan tujuan dan batasan yang sudah ditentukan sebelumnya.

4. Pembangunan Aplikasi

Membangun aplikasi LAB ICT berbasis *web* dengan mengimplentasikan algoritma genetika sesuai dengan rancangan yang telah dibuat.

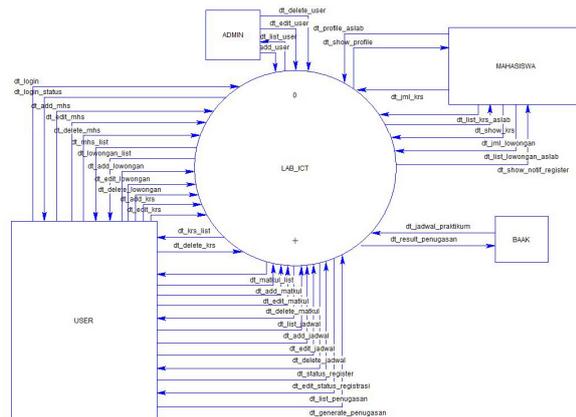
5. Penulisan Laporan

Menulis laporan penelitian mulai dari proses pembuatan aplikasi sampai hasil dan kesimpulan dari penelitian yang dilakukan.

B. Perancangan Sistem

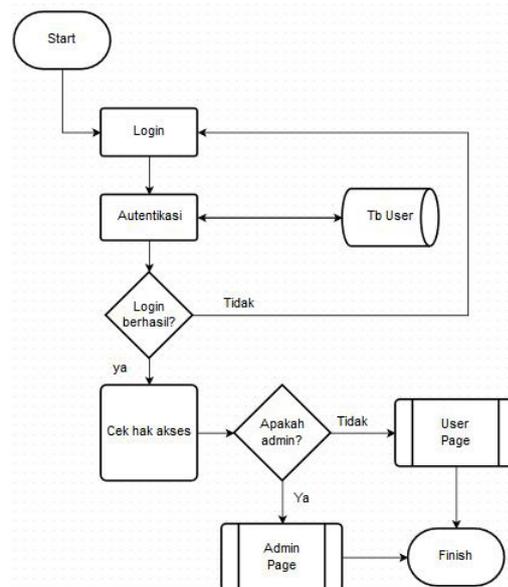
Gambar 2 merupakan DFD *Context Diagram* dengan empat entitas, yaitu *admin*, *user*, mahasiswa, dan BAAK. *Admin* merupakan

kepala atau ketua dari divisi yang mengatur asisten laboratorium di Universitas Multimedia Nusantara. *User* merupakan staff dari Universitas Multimedia Nusantara yang bekerja pada divisi yang sama dengan *admin*. *Admin* memiliki kedudukan yang lebih tinggi dari *user*. Aliran data yang dimiliki oleh *user* juga dimiliki oleh *admin*. *Admin* dapat melakukan apa yang dilakukan oleh *user*.



Gambar 2 DFD *Context Diagram* LAB ICT

Gambar 3 menjabarkan tentang urutan proses dari keseluruhan aplikasi LAB ICT yang dibangun. Pada awalnya *user* atau *admin* akan berada pada halaman *login*. Pada halaman ini, mereka dapat melakukan *login* dengan memasukkan data yang diperlukan yakni *email* dan *password*.

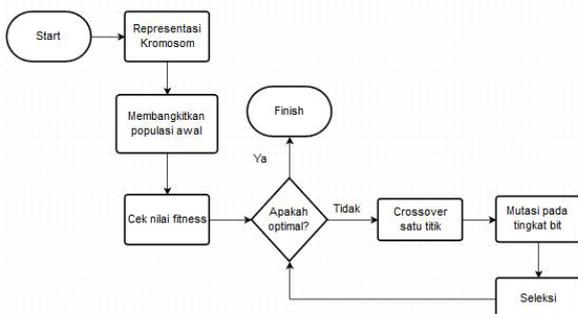


Gambar 3 *Flowchart* Aplikasi

Setelah mengisi data tersebut, data itu akan dicek valid atau tidaknya terhadap tabel user. *Login* berhasil jika pada tabel user terdapat *email* dan *password* yang sesuai dengan data *email* dan *password* yang dimasukkan.

Jika *login* berhasil akan dilakukan pengecekan terhadap hak akses yang dimiliki, apabila hak akses dari data yang dimasukkan merupakan *admin* maka akan menuju ke proses *admin page*, apabila hak akses yang dimiliki oleh data yang dimasukkan merupakan *user* maka akan menuju ke proses *user page*.

Gambar 4 merupakan merupakan *flowchart* dari algoritma yang digunakan. *Generate* penugasan awalnya dimulai dengan representasi kromosom lalu membangkitkan populasi awal, kemudian mengecek nilai *fitness* apakah sudah optimal atau belum. Jika nilai *fitness* belum optimal maka akan dilakukan *crossover* dan mutasi pada tingkat bit. Setelah itu akan dilakukan seleksi dan cek kembali apakah sudah optimal atau belum.



Gambar 4 *Generate* Penugasan dengan Genetika

IV. Implementasi dan Pengujian

Hasil implementasi dan pengujian dari perancangan adalah sebagai berikut.

A. Implementasi

Gambar 5 merupakan tampilan halaman *login* dari aplikasi LAB ICT.



Gambar 5 Tampilan Halaman *Login* Aplikasi

Terdapat *form* untuk melakukan *login* yang memerlukan masukan berupa *email* dan *password* dari *user*. Notifikasi akan muncul apabila terdapat kesalahan data yang dimasukkan. Apabila data yang dimasukkan benar, maka akan diarahkan ke halaman *home*.

Tampilan halaman *Home* saat proses *login* sebagai *admin* berhasil dilakukan ditunjukkan pada Gambar 6. Menu User Management ditampilkan apabila berhasil melakukan *login* sebagai *admin*, sedangkan apabila *login* sebagai *user*, maka menu User Management tidak akan ditampilkan.

Selain menu User Management, terdapat beberapa menu lain, yaitu Mahasiswa Management, Mata Kuliah Management, Pendaftaran Page Management, About Apps, dan Log Out. Mahasiswa Management merupakan sebuah *dropdown* menu yang berisikan menu Mahasiswa Management, KRS Management, dan Lowongan Management. Menu Mata Kuliah Management juga merupakan sebuah *dropdown* menu yang berisikan Mata Kuliah Management dan Jadwal Management. *Banner* pada bagian atas terdapat pada setiap halaman aplikasi. *Banner* ini juga dapat dipilih untuk mengarahkan kembali ke halaman *home*.



Gambar 6 Tampilan Halaman *Home*

Gambar 7 tampilan awal setelah menu KRS Management dipilih. Terdapat tabel yang berisi baris-baris data mahasiswa yang setiap barisnya terdapat tombol view untuk melihat KRS yang diambil oleh mahasiswa tersebut.

Setelah tombol view ditekan, maka daftar KRS yang diambil oleh mahasiswa yang terpilih akan ditampilkan. Daftar KRS yang tampil tersebut dapat ditambah, diubah, dan dihilangkan. Data KRS tersebut meliputi nama mata kuliah, jam mulai, dan jam akhir dari perkuliahan.

ID MHS	NIM	NAMA	PRODI	
30	10001	mhs2011_1	Sistem Informasi	
47	10002	mhs2011_2	Teknik Informatika	
48	10003	mhs2011_3	Teknik Informatika	
49	10004	mhs2011_4	Teknik Informatika	
50	10005	mhs2011_5	Teknik Informatika	
51	10006	mhs2011_6	Sistem Informasi	

Gambar 7 Tampilan Awal KRS Management

Tampilan untuk menu-menu lain yang terdapat dalam aplikasi hampir sama dengan tampilan pada menu pada KRS Management, hanya saja data yang ditampilkan pada setiap menu berbeda. Data yang ditampilkan sesuai dengan nama menu yang dipilih sebelumnya.

Gambar 8 merupakan tampilan hasil dari penugasan aslab yang dilakukan dengan algoritma genetika. Penjadwalan yang terbentuk dimasukkan ke dalam sebuah tabel sehingga mempermudah untuk melihat jadwal yang terbentuk.

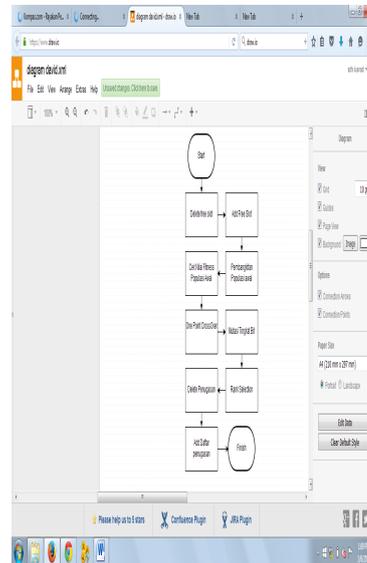
GENERATE PENUGASAN ASLAB						
<div style="text-align: center;"> <input type="button" value="Clear Free slot Mhs"/> <input type="button" value="Generate Assistant Laboratory Assignment Bars"/> </div>						
NAMA MATKUL	HARI	WAKTU AWAL	WAKTU AKHIR	KELAS	NAMA DOSEN	NAMA ASLAB
Algoritma & Pemrograman	Senin	8	9	B305	Rasi R	test
Algoritma & Pemrograman	Senin	8	9	B305	Rasi R	tidak ada penugasan
Algoritma & Pemrograman	Senin	8	9	asdas	test	test
Pengantar Teknologi Multimedia	Sabtu	15	17	TI-A	test3	David Setyadi Santoso
Pengantar Teknologi Multimedia	Sabtu	15	17	TI-A	test3	tidak ada penugasan
Sistem Basis Data	Senin	8	9	qwq	qwqw	test
Sistem Basis Data	Senin	8	9	qwq	qwqw	tidak ada penugasan

Gambar 8 Tampilan Hasil Rancangan Layout

B. Pengujian

Setelah aplikasi LAB ICT ini selesai dibuat, maka langkah selanjutnya adalah melakukan pengujian. *Function* yang dijalankan untuk

menghasilkan penugasan dengan algoritma genetika adalah *function* generate_jadwal_aslab(). Terdapat sembilan tahapan pada *function* ini termasuk tahapan algoritma genetika yaitu sebagai berikut :



Gambar 9. Flowchart Tahapan Pengujian

Berikut ini penjelasan gambar 9 tahapan pengujian aplikasi :

1. Delete Free Slot

Pada tahapan ini akan dilakukan penghapusan data pada tabel `free_slot_mhs` sehingga data free slot yang lama tidak akan diproses.

2. Add Free Slot

Pada tahapan ini akan dilakukan penambahan data pada tabel `free_slot_mhs` berdasarkan tabel `krs_mhs` dan `time_slot`. Data `jam_awal` dan `jam_akhir` pada `krs_mhs` akan diubah menjadi bentuk `time slot` berdasarkan tabel `time_slot` kemudian ditampung ke dalam `array`. Dari `array` tersebut terlihat `time slot` dimana mahasiswa tersebut tidak dapat menerima penugasan sehingga `free time slot` setiap mahasiswa didapatkan dan dimasukkan ke dalam `database`.

3. Pembangkitan Populasi Awal

Pembangkitan populasi awal dilakukan

dengan inisialisasi *array \$populasi_penugasan* yang akan menampung beberapa *array \$set_penugasan1*. *Array \$set_penugasan1* berisi penugasan asisten laboratorium untuk setiap jadwal yang ada. Variabel *\$jml_kromosom_uji* menyatakan jumlah kromosom yang digunakan dalam satu populasi yakni jumlah *array \$set_penugasan1* yang diizinkan dalam *array \$populasi_penugasan*. Variabel *\$max_generation* menyatakan jumlah generasi maksimal yang diizinkan dalam sekali proses. Pembangkitan populasi awal ini dilakukan secara *random* dalam hal penugasan awal mahasiswanya. Apabila seorang mahasiswa memiliki *free time slot* yang sama dengan *time slot* dari jadwal penugasan maka mahasiswa akan ditempatkan.

4. Cek Nilai *Fitness* Populasi Awal

Pengecekan nilai *fitness* untuk populasi awal dilakukan terhadap *array \$set_penugasan1* yang telah ditampung dalam *array \$populasi_penugasan*. Untuk setiap penugasan yang benar akan membuat nilai *fitness* bertambah. Nilai *fitness* dari setiap *\$set_penugasan1* tersebut akan ditampung ke dalam sebuah *array \$nilai_fitness*.

5. *One Point Crossover*

Sebelum melakukan *crossover*, set penugasan yang ada akan diduplikasi sehingga jumlah kromosom dalam satu populasi ini menjadi dua kali lipat. *Crossover* akan dilakukan pada setiap kromosom pada satu populasi. *One point crossover* ini membutuhkan sebuah titik potong untuk melakukan *crossover*. Titik potong ini berada di tengah-tengah yang didapat dengan cara membagi dua jumlah penugasan yang terdapat dalam satu set penugasan. Setelah titik potong didapat, maka akan dilakukan *crossover* dengan batas titik potong itu sendiri.

6. Mutasi Tingkat Bit

Mutasi dilakukan pada setiap set penugasan pada setiap *array \$set_penugasan1*. Mutasi dilakukan dengan cara mengganti penugasan yang masih salah pada suatu

jadwal dengan menempatkan mahasiswa yang cocok untuk mata kuliah dan jadwal tersebut. Jika terdapat penugasan yang salah, akan diambil data mahasiswa yang sesuai dengan mata kuliah dimana penugasan itu salah. Setelah mendapatkan data mahasiswa, setiap mahasiswa tersebut akan melalui tiga pengecekan yaitu pengecekan apakah jadwal penugasan ini sesuai dengan *free slot* mahasiswa, pengecekan untuk jadwal yang membutuhkan dua asisten laboratorium apakah mahasiswa sudah ditugaskan pada jadwal yang sama, dan yang terakhir pengecekan apakah waktu penugasan yang baru ini bentrok dengan waktu penugasan yang telah didapat mahasiswa ini sebelumnya. Jika berhasil melewati semua pengecekan tanpa adanya bentrokan, maka mahasiswa akan mendapatkan penugasan yang baru.

7. *Rank Selection*

Pada tahapan ini akan dilakukan seleksi *rank*, yaitu pengurutan set penugasan yang terdapat pada *array \$populasi_penugasan* berdasarkan nilai *fitness*. Set penugasan akan diurutkan berdasarkan nilai *fitness* tertinggi. Jumlah kromosom atau set penugasan yang ada *array \$populasi_penugasan* tentu memiliki jumlah dua kali lipat dari jumlah kromosom awal. Setelah diurutkan, maka akan dipilih set penugasan sejumlah kromosom awal yaitu variabel *\$jml_kromosom_uji* dimana jumlah kromosom atau set penugasan yang ada *array \$populasi_penugasan* saat ini tentu memiliki jumlah dua kali lipat dari jumlah kromosom awal. Set-set penugasan yang terpilih akan digunakan sebagai induk untuk *crossover* dan mutasi pada perulangan selanjutnya.

8. *Delete* Penugasan

Penghapusan semua data pada tabel *daftar_penugasan* sehingga penugasan yang baru tidak berganda dengan penugasan yang lama.

9. *Add* Daftar Penugasan

Pada tahap ini telah ditemukan set penugasan dengan nilai *fitness* yang terbaik. Data

penugasan terbaik dimasukkan ke dalam tabel daftar_penugasan.

Hasil penugasan dari pengujian yang dilakukan diharapkan memenuhi ekspektasi hasil sebagai berikut.

1. Penugasan tidak bentrok dengan KRS mahasiswa.
2. Penugasan tidak bentrok dengan penugasan lain yang telah diberikan.
3. Penugasan tidak bentrok pada jadwal yang sama apabila jadwal tersebut membutuhkan dua asisten laboratorium.
4. Penugasan sebisa mungkin memenuhi semua jadwal yang ada.

Pengujian dilakukan dengan menggunakan data yang didapat dari LAB ICT. Berdasarkan pengujian terhadap penggunaan jumlah kromosom dan jumlah generasi, jumlah maksimal generasi yang digunakan adalah sebanyak 12 generasi. Jumlah kromosom dalam satu generasi yang digunakan adalah sebanyak 12 kromosom. Penggunaan jumlah generasi dan kromosom melebihi 12 tidak memberikan perbedaan yang signifikan. Dari hasil uji coba dapat dikatakan 12 merupakan jumlah kromosom dan generasi yang optimal. Berikut hasil pengujian yang didapatkan.

Tabel 1 Hasil Pengujian dengan Data LAB ICT

No	Hasil Uji Coba	Nilai Fitness
1	94 penugasan	67.142%
2	96 penugasan	68.571%
3	95 penugasan	67.857%
4	94 penugasan	67.142%
5	94 penugasan	67.142%
6	94 penugasan	67.142%
7	95 penugasan	67.857%
8	96 penugasan	68.571%
9	94 penugasan	67.142%
10	94 penugasan	67.142%
	Rata-rata nilai fitness	67.571%

Berdasarkan hasil pengujian diketahui bahwa

nilai *fitness* tertinggi yang didapat adalah 68.571% dengan 96 penugasan. Dari sepuluh kali uji coba yang dilakukan, didapat rata-rata nilai *fitness* sebesar 67.571%. Nilai *fitness* yang didapat ini terkendala oleh jumlah mahasiswa yang tidak memenuhi jumlah *slot* penugasan. Berdasarkan data yang digunakan, terdapat 22 slot penugasan untuk mata kuliah tertentu yang tidak memiliki peminat. Apabila 22 slot tersebut tidak digunakan dalam perhitungan nilai *fitness*, hanya terdapat pada 118 slot penugasan saja dan nilai *fitness* yang dimiliki berubah menjadi 80,5%.

V. Simpulan dan Saran

Dapat diambil kesimpulan bahwa algoritma genetik dapat diimplementasikan pada aplikasi ICT LAB sebagai alternatif solusi untuk menyelesaikan masalah penugasan asisten laboratorium. Berdasarkan hasil uji coba, dengan menggunakan *one point crossover*, mutasi pada tingkat bit, dan *rank selection* didapatkan rata-rata nilai *fitness* sebesar 67.571% dari total lowongan asisten laboratorium yang tersedia. Nilai *fitness* yang didapat ini terkendala oleh jumlah mahasiswa yang tidak memenuhi jumlah *slot* penugasan dan terdapat slot penugasan untuk mata kuliah tertentu yang tidak memiliki peminat. Nilai *fitness* yang dimiliki berubah menjadi 80,5% saat slot-slot tersebut tidak dimasukkan dalam perhitungan nilai *fitness*.

Daftar Pustaka

- [1] Sucita, Tasma. 2006. "Pengembangan Model Pembelajaran Praktikum Berbasis Software Komputer". Bandung: Universitas Pendidikan Indonesia.
- [2] Aji, Satria Bayu. 2015. "Sistem Pendukung Keputusan Seleksi Asisten Laboratorium Dosen Elektro Menggunakan Metode Weighted Product Di Polines". Semarang: Universitas Dian Nuswantoro.
- [3] Puspaningrum, Wiga Ayu, dkk. 2013. "Penjadwalan Mata Kuliah Menggunakan Algoritma Genetika di Jurusan Sistem Informasi ITS". JURNAL TEKNIK POMITS Vol. 2, No. 1.
- [4] Mahmudy, WF 2006, "Penerapan Algoritma Genetika pada Optimasi Model Penugasan". Natural, vol.10, no. 3, pp. 197-207.

- [5] Goldberg, David Edward. 1989. "Genetic Algorithms in Search, Optimization, and Machine Learning". Michigan: Addison-Wesley Publishing Company.
- [6] Luger, George F. 2009. "Artificial Intelligence: Structures and Strategies for Complex Problem Solving Sixth Edition". Boston: Pearson Education, Inc.
- [7] Russell, Stuart dan Peter Norvig. 2010. "Artificial Intelligence: A Modern Approach Third Edition". New Jersey: Pearson Education, Inc.
- [8] Pardede, Abigael Angela, Shanny Avelina Halim & Denny Nugrahadi. 2006. "Penerapan Algoritma Genetika pada Permainan Rubik's Cube". Bandung: Departemen Teknik Informatika, Institut Teknologi Bandung.
- [9] Ulfa, Lina Maria. 2011. "Optimasi Penjadwalan Perkuliahan Menggunakan Algoritma Genetika": Skripsi pada Universitas Islam Negeri Maliki Malang.
- [10] Suyanto. 2007. "Artificial Intelligence: Searching, Reasoning, Planning and Learning". Bandung : Informatika.
- [11] Bronson, Richard. 1982 "Schaum's outline of theory and problems of operations research". Michigan: McGraw-Hill.