Implementasi Sistem Penjadwalan Akademik Fakultas Teknik Universitas Tanjungpura Menggunakan Metode Algoritma Genetika

Andreas Christian[#1], Herry Sujaini[#2], Arif Bijaksana Putra Negara[#3]
[#]*Program Studi Teknik Informatika Universitas Tanjungpura*
Jl. Prof. Dr. H. Hadari Nawawi, Pontianak, Kalimantan Barat 78115
[1]andreaschristian40@gmail.com1, [2]hs@untan.ac.id, [3]arifbpn@gmail.com

***Abstrak*— Penjadwalan akademik merupakan masalah mengalokasikan waktu, mata kuliah, dosen, ruangan kelas, dengan memperhatikan sejumlah aturan yang berhubungan dengan kapasitas dan lokasi ruangan yang tersedia, waktu bebas yang diperlukan dan sejumlah aturan lain yang berkaitan. Berdasarkan karakteristik permasalahan yang terjadi, algoritma genetika cocok untuk menyelesaikannya dengan merepresentasikan komponen-komponen masalah ke dalam kromosom dan dievaluasi berdasarkan constraint penjadwalan yang telah dibuat sehingga menghasilkan jadwal kuliah yang optimal. Aplikasi dibangun berbasis web dengan menggunakan bahasa pemrograman PHP dan database MySQL. Proses optimasi penjadwalan dengan menggunakan algoritma genetika akan melalui proses representasi kromosom, inisialisasi populasi, menghitung fitness, seleksi, crossover dan mutasi. Pengujian dilakukan dengan metode black box untuk menguji komponen-komponen yang ada didalam sistem penjadwalan dan pengujian performansi untuk mengukur parameter-parameter yang optimal untuk proses penjadwalan. Berdasarkan hasil pengujian, karakteristik bentrok yang sering terjadi dalam proses algoritma genetika adalah karakteristik bentrok persimpangan waktu, banyak generasi yang optimal yaitu 200 sampai dengan 250, ukuran populasi yang optimal 100 dan nilai probabilitas crossover rate 0.7 dan mutation rate 0.3, serta dengan diimplementasikannya soft constraint penggunaan kapasitas ruangan mempengaruhi tingkat keberhasilan Hard Constraint. Dengan diimplementasikannya soft constraint kapasitas ruangan maka tingkat keberhasilan soft constraint-nya meningkat mencapai 93% dari 75%, Tetapi dengan diimplementasikannya soft constraint tersebut maka waktu yang dibutuhkan untuk komputasi meningkat dari 2,8 menit menjadi 38 menit. Hal ini terjadi karena pada proses komputasi harus melakukan pengecekan terhadap jumlah peserta dan kapasitas ruangan pada setiap kromosom.**

***Kata Kunci*—Penjadwalan Akademik, Mata Kuliah, Algoritma Genetika, Optimasi**

I. PENDAHULUAN

Penjadwalan merupkan cara untuk mengalokasikan waktu terhadap suatu kegiatan. Tujuan dari penjadwalan agar kegiatan tersebut dapat berlangsung secara lancar. Penyusunan jadwal kegiatan berkaitan dengan berbagai batasan yang harus dipenuhi sehingga memerlukan banyak pertimbangan dalam pembuatan jadwal tersebut. Ada berbagai masalah penjadwalan yang sering dihadapi seperti alokasi kegiatan, orang, kejadian dan kendaraan. Sebagian besar masalah pada penjadwalan sangat rumit dilakukan karena keterbatasan sumber daya (waktu, tempat dan orang). Jadi penentuan jadwal yang baik menjadi masalah penting dalam suatu kegiatan.

Penjadwalan akademik merupakan penjadwalan perkuliahan. Berdasarkan hasil wawancara masalah yang terjadi adalah masalah mengalokasikan waktu, mata kuliah, dosen, ruangan kelas, dengan memperhatikan sejumlah aturan yang berhubungan dengan kapasitas dan lokasi ruangan yang tersedia, waktu bebas yang diperlukan dan sejumlah aturan lain yang berkaitan.

Komponen-komponen utama dari penjadwalan perkuliahan adalah dosen, mahasiswa, mata kuliah, waktu dan ruangan kelas. Hasil dari proses penjadwalan mata kuliah merupakan pengelompokan komponen-komponen utama secara bersama-sama dengan memperhatikan aturan-aturan dalam penyusunan jadwal yang bersifat harus dipenuhi (*hard constraint*) dan tidak harus terpenuhi (*soft constraint*), meskipun *soft constraint* tidak harus terpenuhi tetapi tetap menjadi acuan dalam proses pembuatan jadwal.

Masalah pada penjadwalan pada dasarnya merupakan kombinasi yang cukup rumit antara dosen, mata kuliah, ruang, waktu dan kelas. Masalah tersebut dapat diselesaikan dengan mengkombinasikan secara manual untuk mendapatkan jadwal yang baik. Namun cara tersebut membutuhkan waktu yang relatif lama dan harus dikerjakan secara cermat serta teliti agar tidak terjadi konflik pada pembuatan jadwal. Metode optimasi dibutuhkan untuk mengerjakan masalah penjadwalan ini.

Persoalan optimasi (*optimization problem)* adalah persoalan yang menuntut pencarian solusi optimum [1]. Penelitian-penelitian terbaru menyarankan bahwa algoritma genetika merupakan metode yang layak dan efektif dalam mengatasi masalah penjadwalan [2]. Algortima genetika merupakan pendekatan komputasional untuk menyelesaikan masalah yang dimodelkan dengan proses biologi dari evolusi. Diharapkan dengan digunakannya algoritma genetika akan diperoleh optimasi penjadwalan terbaik dari kombinasi antara dosen, mata kuliah, ruang, waktu dan kelas.

Berdasarkan pada latar belakang tersebut, maka dilakukan penelitian mengenai masalah penjadwalan dengan

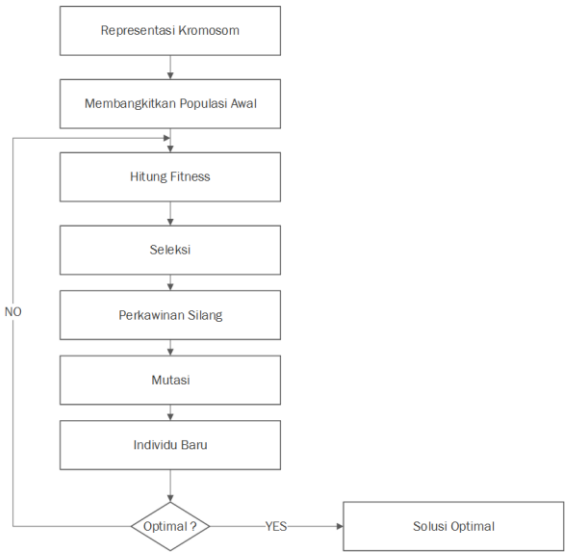
Submitted 20-02-2017; Revised 20-02-2017; Accepted 20-02-2017

menggunakan algoritma genetika untuk mengoptimasi penyusunan jadwal kuliah. Adapun judul yang ingin diteliti adalah Implementasi Sistem Penjadawlan Akademik di Fakultas Teknik Universitas Tanjungpura dengan Algoritma Genetika.

II. URAIAN PENELITIAN

A. *Algoritma Genetika*

Algoritma genetika adalah algoritma pencarian/heuristic yang didasarkan atas mekanisme seleksi alam dan evolusi biologis. Algoritma genetika pertama kali diperkenalkan oleh John Holland dalam bukunya yang berjudul "*Adaption in natural and artificial system"*, dan oleh De Jong dalam bukunya "*Adaption of the behaviour of a class of genetic adaptive systems"*, yang keduanya diterbitkan pada tahun 1975, yang merupakan dasar dari algoritma genetika [3].



Gambar. 1. *Flowchart* Algoritma Genetika.

Teknik pencarian yang dilakukan oleh algoritma genetika bersamaan dengan solusi yang mungkin yang dikenal dengan istilah populasi. Individu yang terdapat dalam satu populasi disebut dengan isitilah kromosom. Kromosom ini merupakan suatu solusi yang masih berbentuk simbol. Populasi awal dibangun secara acak, sedangkan populasi berikutnya merupakan hasil evolusi kromosom-kromosom melalui proses iterasi yang disebut dengan istilah generasi. Pada setiap generasi, kromosom akan melalui proses evaluasi dengan menggunakan alat ukur yang disebut dengan fungsi *fitness.* Nilai *fitness* dari suatu kromosom akan menunjukkan kualitas kromosom dalam populasi tersebut. Generasi berikutnya dikenal dengan istilah anak (*offspring*) yang terbentuk dari gabungan dua kromosom generasi sekarang yang bertindak sebagai induk (*parent)* dengan menggunakan operator penyilangan, suatu kromosom dapat juga dimodifikasi dengan menggunakan operator mutasi. Populasi generasi yang baru dibentuk dengan cara menyeleksi nilai *fitness* dari kromosom induk (*parent)* dan nilai *fitness* dari kromosom anak (*offspring*), serta menolak kromosom-kromosom yang lainnya sehingga ukuran populasi (jumlah kromosom dalam suatu populasi) konstan. Setelah melalui beberapa generasi, maka algoritma ini akan konvergen ke kromosom terbaik.

B. *Struktur Umum Algoritma Genetika*

Algoritma genetika memberikan suatu pilihan bagi penentuan nilai parameter dengan meniru cara reproduksi genetik, pembentukan kromosom baru serta seleksi alami seperti yang terjadi pada mahluk hidup.

Inisialisasi populasi awal dilakukan untuk menghasilkan solusi awal dari suatu permasalahan algoritma genetika. Inisialisasi ini dilakukan secara acak sebanyak jumlah kromosom yang diinginkan. Selanjutnya dihitung nilai *fitness* dan seterusnya dilakukan seleksi dengan menggunakan metode roda roullete, tournament atau ranking. Kemudian dilakukan perkawinan silang (*crossover)* dan mutasi. Setelah melalui beberapa generasi maka algoritma ini akan berhenti sebanyak generasi yang diinginkan. Sebagaimana halnya proses evolusi di alam, suatu algoritma genetika yang sederhana umumnya terdiri dari tiga operator yaitu: operator reproduksi, operator *crossover(*persilangan) dan operator mutasi. [4].

Ada dua hal penting yang harus dilakukan pada awal proses algoritma genetika. pertama, pendefinisian atau pengkodean kromosom yang merupakan solusi yang masih berbentuk simbol. Kedua, penentuan fungsi *fitness* atau fungsi obyektif. Dua hal ini berperan penting dalam algoritma genetika untuk menyelesaikan suatu masalah [4].

C. *Istilah Umum Algoritma Genetika*

Dalam mempelajari algoritma genetika ada beberapa istilah yang harus dipahami, yaitu [5]:

- Genotype (gen): sebuah nilai yang menyatakan satuan dasar yang membentuk suatu arti tertentu dalam satu kesatuan yang dinamakan kromosom.
- Allele : nilai dari gen.
- Kromosom: gabungan gen-gen yang membentuk nilai terentu.
- Individu: merupakan suatu nilai atau keadaan yang menyatakan salah satu solusi yang mungkin dari permasalahan yang diangkat.
- Populasi: merupakan sekumpulan individu yang akan diproses bersama dalam siklus evolusi.
- Nilai fitness: menyatakan seberapa baik nilai dari suatu individu atau solusi yang didapatkan.
- Generasi, merupakan satu siklus proses evolusi atau satu iterasi di dalam algoritma genetika.

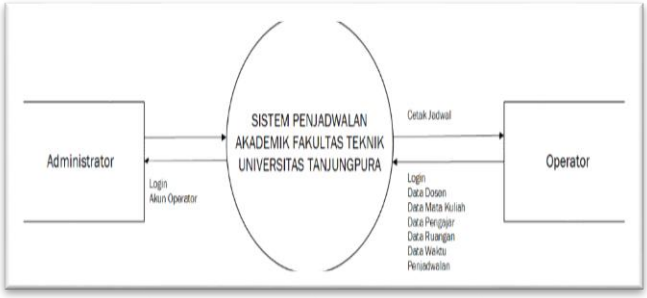
Dalam algoritma genetika nilai fitness dijadikan acuan dalam mencapai nilai optimal, dimana untuk masalah pencarian nilai maksimum maka yang dicari adalah nilai fitness tertinggi.

Korespondensi: 20-02-2017; Revisi 20-02-2017; Submitted 20-02-2017; Revised 20-02-2017; Accepted 20-02-2017

III. PERANCANGAN DAN HASIL

A. *Diagram Konteks Sistem*

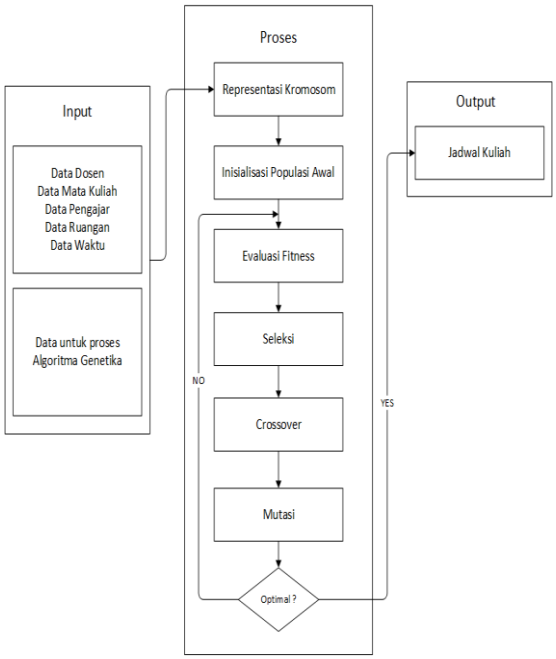
Diagram konsteks pada sistem penjadwalan akademik dapat dilihat pada Gambar 2.



Gambar. 2. Diagram Konteks Sistem

B. *Gambaran Umum Sistem*

.Gambaran umum sistem yang dirancang dengan menggunakan algoritma genetika dapat dilihat pada Gambar 3.



Gambar. 3. *Gambaran Umum Sistem*

Gambaran umum sistem merupakan gambaran langkah-langkah penjadwalan dengan menggunakan algoritma genetika. Sistem penjadwalan yang dibangun memiliki beberapa proses berupa data masukkan, proses utama dan keluaran.

Data masukkan membutuhkan data dosen, data mata kuliah, data pengajar, data ruangan, data waktu serta parameter-parameter yang diperlukan untuk proses algoritma genetika seperti parameter populasi, probabilitas *crossover*, probabilitas mutasi dan jumlah generasi.

Proses utama, menyatakan proses-proses utama yang terdapat pada perancangan sistem penjadwalan akademik menggunakan algoritma genetika, yaitu dimulai dari representasi kromosom dan inisialisasi populasi awal, evaluasi *fitness* dan seleksi ranking (*rank-based selection*), pindah silang (*crossover*) dilakukan dengan penyilangan dua titik (*two-point crossover*) berdasarkan probabilitas *crossover* yang telah ditentukan dan mutase berdasarkan probabilitas mutase yang telah ditentukan. Setelah melalui beberapa generasi, maka algoritma ini akan menghasilkan kromosom terbaik yang merupkan solusi optimal.

Keluaran merupakan hasil dari proses-proses utama yang terjadi pada sistem. Keluaran dari sistem berupa jadwal perkuliahan.

C. *Pengujian Aplikasi*

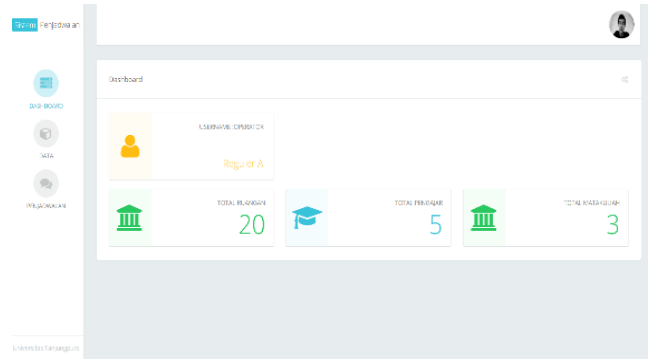
Pengujian aplikasi dilakukan dengan menggunakan pengujian *Black Box* untuk memeriksa kinerja antar komponen yang telah dibangun dan pengujian performansi untuk mengetahui parameter-parameter algoritma genetika yang optmal agar menghasilkan penjadwalan optimal dan mengetahui karakteristik bentrok dalam penggunaan metode algoritma genetika.

D. *Hasil Aplikasi*

Sistem penjadwalan yang dibangun berbasis *web*. Aplikasi ini dibangun untuk membuat jadwal kuliah secara otomatis dengan waktu yang relatif cepat dibandingkan cara konvensional. Berikut beberapa tampilan hasil perancangan aplikasi, yang diperlihat pada Gambar 4 sampai dengan Gambar 9.

Gambar. 4. Tampilan Halaman Login.

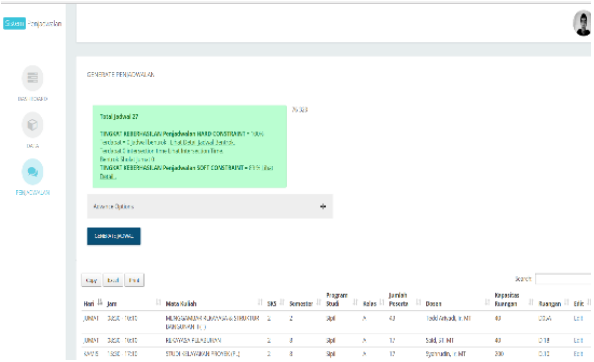
Gambar 4 merupakan tampilan halaman login dari sistem penjadwalan akademik.



Gambar. 5. Tampilan Halaman *Dashboard* Operator.

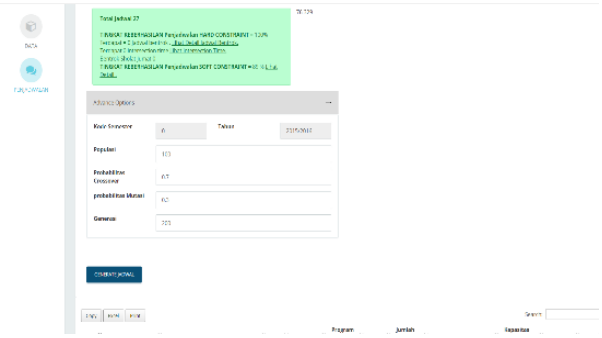
Gambar 5 merupakan tampilan halaman *dashboard* operator dimana pada menu ini berisi informasi-informasi tentang komponen-komponen penjadwalan.

Korespondensi: 20-02-2017; Revisi: 20-02-2017; Accepted 20-02-2017
Submitted 20-02-2017; Revised 20-02-2017; Accepted 20-02-2017



Gambar. 6. Tampilan Halaman *Generate* Penjadwalan.

Gambar 6 merupakan tampilan halaman untuk *generate* jadwal kuliah.

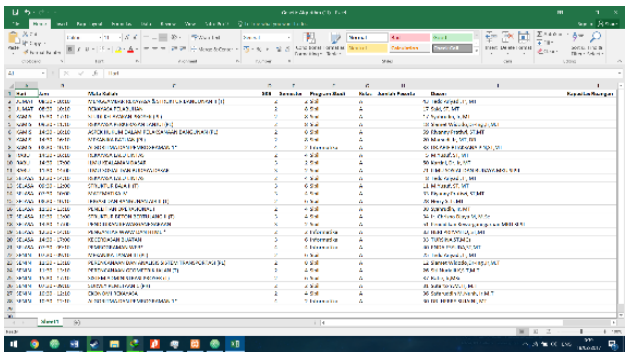


Gambar. 7. Tampilan *Input* parameter algoritma genetika

Gambar 7 merupakan tampilan *input* parameter algoritma genetika, dimana *input* dari parameter ini akan digunakan untuk proses algoritma genetika.

Gambar. 8. Tampilan *Output* Jadwal Kuliah

Gambar 8 merupakan tampilan *output* dari proses penjadwalan dengan menggunakan algoritma genetika.



Gambar. 9. Tampilan *Export Dalam Excel*

Gambar 9 merupakan tampilan *export* jadwal dalam bentuk *excel*.

E. Hasil Pengujian

Pengujian dilakukan pada sistem menggunakan metode *black box* yang akan memeriksa kinerja antar komponen yang telah dibangun dan juga pengujian performansi untuk menguji parameter-parameter algoritma genetika sehingga menghasilkan penjadwalan yang optimal.

E.1 Hasil Pengujian Black Box

Pengujian dengan *Black Box* dilakukan untuk menguji kesesuaian antara masukkan dengan hasil yang ditampilkan pada aplikasi. Hasil pengujian *Black Box* dapat dilihat pada Tabel 1.

Tabel 1 Pengujian *Black Box*

No	Fungsi	Keterangan
1	Fungsi Login	Berhasil
2	Manajemen Akun Operator	Berhasil
3	Manajemen Data Dosen	Berhasil
4	Manajemen Data Jam	Berhasil
5	Manajemen Waktu Berhalangan	Berhasil
6	Manajemen Data Hari	Berhasil
7	Manajemen Data Pengajar	Berhasil
8	Manajemen Data Mata Kuliah	Berhasil
9	Manajemen Data Ruangan	Berhasil
10	Proses Penjadwalan	Berhasil

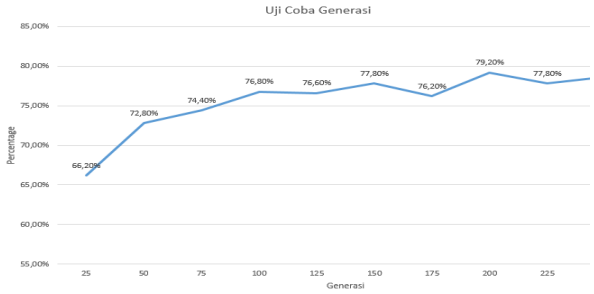
E.2 Hasil Pengujian Performansi

Pengujian performansi dilakukan untuk mengetahui parameter-parameter algoritma genetika yang optmal agar menghasilkan penjadwalan optimal.

Pada pengujian performansi ini terdapat 5 skenario. Skenario pertama akan menggunakan 40 mata kuliah, 40 jumlah kelas dan 2 ruangan. Skenario kedua akan menggunakan 40 mata kuliah, 40 jumlah kelas dan 3 ruangan. Skenario ketiga akan menggunakan 40 mata kuliah dan 60 jumlah kelas dan 3 ruangan. Skenario yang keempat akan menggunakan 40 mata kuliah, 60 jumlah kelas dan 4 ruangan dan Skenario Kelima akan menggunakan 60 jumlah kelas dan 3 ruangan serta mengimplementasikan kapasitas ruangan terhadap jumlah peserta. Pengujian dilakukan dengan menggunakan data jadwal teknik sipil fakultas teknik universitas tanjungpura pada semester genap tahun ajaran 2015/2016.

E.2.1 Skenario Pertama

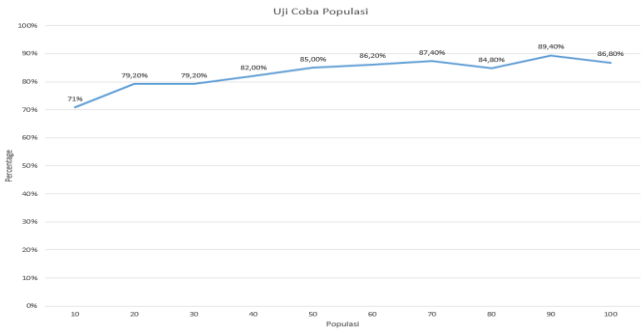
Pada skenario pertama akan menguji parameter berupa banyaknya generasi, populasi dan kombinasi *crossover* dan mutasi untuk mengetahui parameter yang optimal. Pengujian dilakukan sebanyak 5 kali percobaan dan di ambil nilai rata-ratanya.



Gambar. 10. Uji Coba Generasi Skenario Pertama

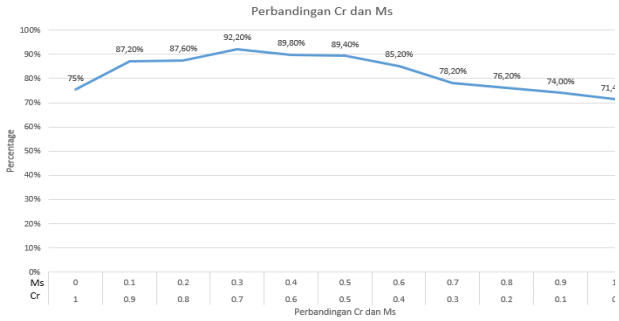
Korespondensi: 20-02-2017; Revisi 20-02-2017; Accepted 20-02-2017

Berdasarkan Gambar 10, secara umum semakin banyaknya generasi maka persentase keberhasilan semakin meningkat dan semakin banyaknya generasi waktu yang dibutuhkan semakin banyak.



Gambar. 11. Uji Coba Populasi Skenario Pertama

Berdasarkan Gambar 11, secara umum semakin besar ukuran populasi maka persentase keberhasilannya semakin meningkat tetapi mengalami grafik naik turun karena terjadi konvergensi dimana kromosom mempunyai nilai yang berdekatan.

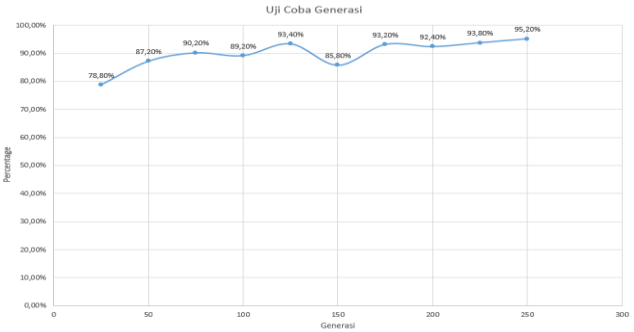


Gambar. 12. Uji Coba Perbandingan *Crossover* dan Mutasi Skenario Pertama

Berdasarkan Gambar 12, Persentase rata-rata terbaik adalah 92.20% pada kombinasi *crossover rate* 0.7 dan *Mutation Rate* 0.3.

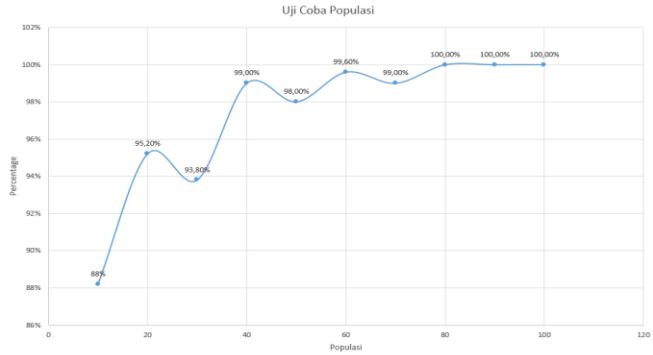
E.2.2 Skenario Kedua

Pada skenario kedua akan menguji parameter-parameter algoritma genetika dengan penambahan jumlah ruangan.



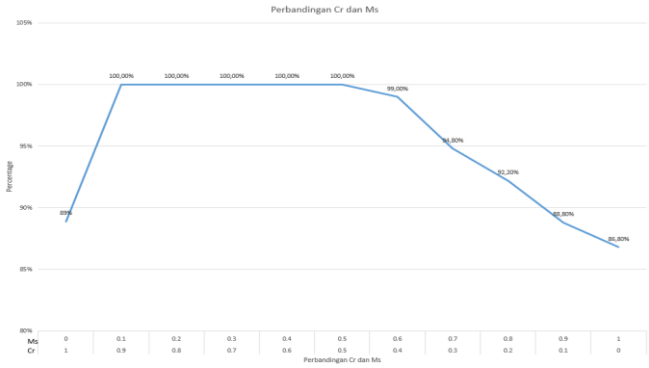
Gambar. 13. Uji Coba Generasi Skenario Kedua

Berdasarkan Gambar 13, jumlah generasi berpengaruh terhadap proses pada algoritma genetika yang terlihat pada persentasenya. Secara umum semakin banyak jumlah generasi maka persentase keberhasilannya semakin meningkat.



Gambar. 14. Uji Coba Populasi Skenario Kedua

Berdasarkan Gambar 14, secara umum semakin besar ukuran populasi maka persentase keberhasilannya semakin meningkat. Terlihat pada grafik dengan penambahan jumlah ruangan tingkat keberhasilannya mencapai 100%.

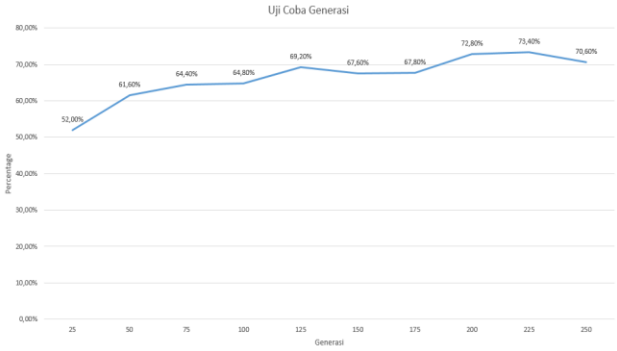


Gambar. 15. Uji Coba Perbandingan *Crossover* dan Mutasi Skenario Kedua

Berdasarkan Gambar 15, persentase terbaik terdapat pada *crossover rate* 0.9, 0.8, dan 0.7 dan *Mutation Rate* 0.1, 0.2 dan 0.3.

E.2.3 Skenario Ketiga

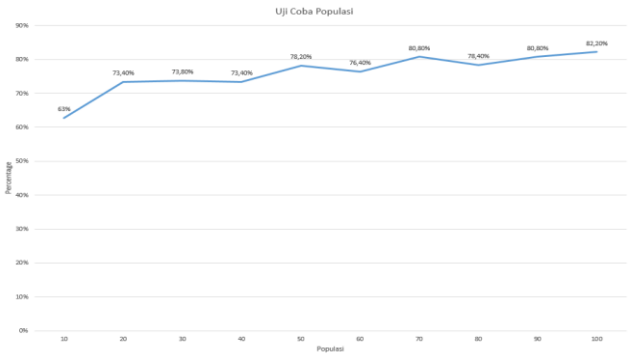
Pada skenario ketiga akan menguji parameter-parameter algoritma genetika dengan penambahan data perkuliahan.



Gambar. 16. Uji Coba Generasi Skenario Ketiga

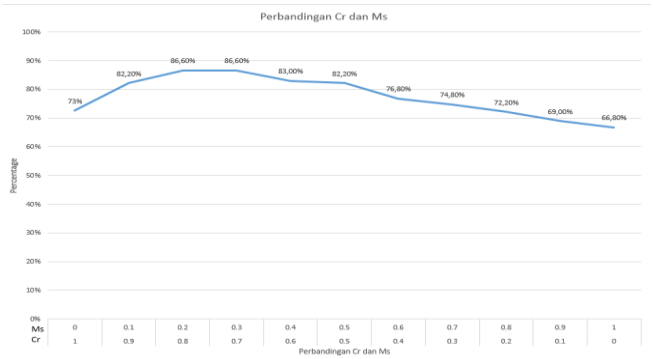
Berdasarkan Gambar 16, secara umum banyaknya jumlah generasi berpengaruh terhadap persentase keberhasilannya. Semakin banyak jumlah generasi maka persentase keberhasilannya semakin meningkat. Tetapi waktu yang diperlukan juga semakin lama.

Korespondensi: Submitted 20-02-2017; Revised 20-02-2017; Accepted 20-02-2017



Gambar. 17. Uji Coba Populasi Skenario Ketiga

Berdasarkan Gambar 17, secara umum semakin besar ukuran populasi maka persentase keberhasilannya semakin meningkat.

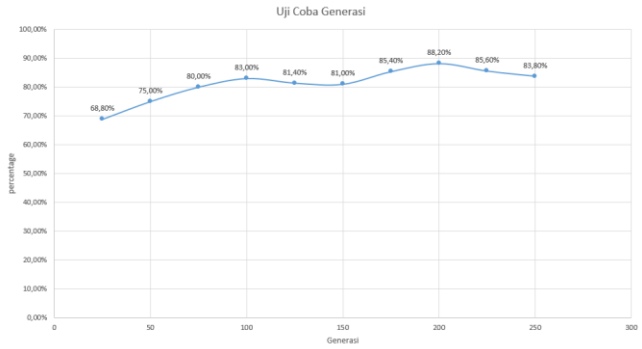


Gambar. 18. Uji Coba Perbandingan *Crossover* dan Mutasi Skenario Ketiga

Berdasarkan Gambar 18, persentase rata-rata terbaik pada kombinasi *crossover rate* 0.8 dan 0.7 dan *Mutation Rate* pada 0.2 dan 0.3.

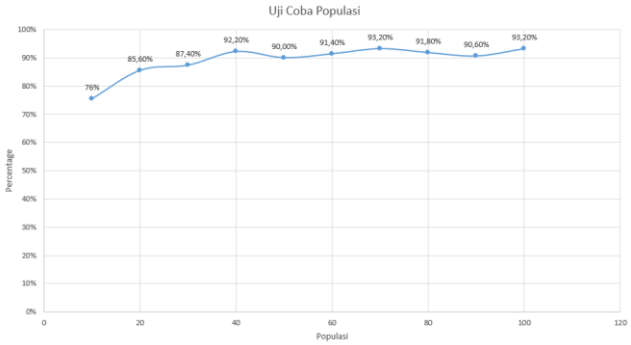
E.2.4 Skenario Keempat

Pada skenario keempat akan menguji parameter-parameter algoritma genetika dengan penambahan data perkuliahan dan data ruangan.



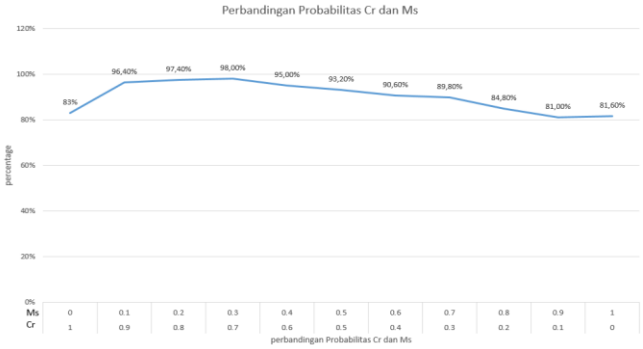
Gambar. 19. Uji Coba Generasi Skenario Keempat

Berdasarkan Gambar 19, jumlah generasi berpengaruh terhadap hasil persentase keberhasilan penjadwalan akademik. Secara umum semakin banyaknya generasi maka persentase keberhasilannya semakin meningkat. Tetapi terjadi grafik naik turun. Kondisi ini yang disebut konvergenasi, artinya hampir semua kromosom mempunyai nilai yang berdekatan.



Gambar. 20. Uji Coba Populasi Skenario Keempat

Berdasarkan Gambar 20, secara umum semakin besar ukuran populasi maka persentase keberhasilannya semakin meningkat. Tetapi ukuran populasi lainnya yang berdekatan mengalami naik turun, tetapi dengan selisih yang tidak signifikan. Kondisi ini disebut dengan konvergensi.



Gambar. 21. Uji Coba Perbandingan *Crossover* dan Mutasi Skenario Keempat

Berdasarkan Gambar 21, persentase rata-rata terbaik adalah 98.00% pada kombinasi *crossover rate* dan *mutation rate* 0.3. rata-rata persentase terendah adalah 81% pada kombinasi *crossover* 0.1 dan *mutation rate* 0.9.

E.2.5 Skenario Kelima

Skenario kelima menggunakan jumlah populasi 100 dengan banyak generasi 225 dan kombinasi *crossover rate* dan *mutation rate* adalah 0.7 dan 0.3. Parameter tersebut diambil karena pada skenario ketiga parameter tersebut dinilai menghasilkan solusi yang optimal. Dalam pengujian dilakukan sebanyak 5 kali dan dihitung rata-ratanya.

Tabel 2 Pengujian Skenario Kelima

Sebelum Implementasi *Soft Constraint* kapasitas kaungan			Sesudah Implementasi *Soft Constraint* kapasitas ruangan		
Hard Constraint	*Soft Constraint*	*Time*	*Hard Constraint*	*Soft Constraint*	*Time*
87%	75%	2.8 Menit	85.4%	93%	38.06 Menit

Berdasarkan Tabel 2, penggunaan *soft constraint* kapasitas ruangan pada komputasi algoritma genetika mempengaruhi tingkat keberhasilan *hard constraint*. Meskipun perbedaan persentase keberhasilannya tidak terlalu signifikan. Dengan diimplementasikannya *soft constraint* kapasitas ruangan maka tingkat keberhasilan *soft constraint*-nya meningkat mencapai 93% dari 75%. Tetapi dengan diimplementasikannya *soft*

Korespondensi: Andreas Chrisian
Submitted 20-02-2017; Revised 20-02-2017; Accepted 20-02-2017

constraint tersebut maka waktu yang dibutuhkan untuk komputasi meningkat dari 2.8 menit menjadi 38.06 menit. Hal ini terjadi karena proses komputasi harus melakukan pengecekan terhadap jumlah peserta dan kapasitas ruangan pada setiap kromosom.

F. *Analisis Hasil Pengujian*

Rincian hasil analisis pengujian pada aplikasi sistem penjadwalan akademik dengan algoritma genetika adalah sebagai berikut :

1. Hasil pengujian *Black Box* menunjukkan bahwa komponen-komponen dari sistem telah berfungsi sesuai dengan yang diharapkan.
2. Dari hasil pengujian yang telah dilakukan, dapat disimpulkan bahwa nilai parameter algoritma genetika berpengaruh terhadap hasil optimasi yang dihasilkan. Ukuran parameter yang kecil menyebabkan area pencarian algoritma genetika semakin sempit. Tetapi, ukuran parameter yang terlalu besar menyebabkan waktu komputasi semakin lama.
3. Berdasarkan hasil pengujian dari keempat skenario algoritma genetika dapat disimpulkan bahwa rasio jumlah total sks terhadap rasio ruangan dan waktu sangat berpengaruh terhadap tingkat keberhasilan penjadwalan.
4. Dari keempat skenario tersebut dapat disimpulkan pelanggaran/bentrok yang sering terjadi pada komputasi algoritma ada pada bentrok penyimpangan waktu, dimana masih terjadi *overlapping* time pada hari dan ruangan yang sama.
5. Proses implementasi *Soft Constraint* kapasitas ruangan memakan waktu yang lama, karena pengecekan terhadap jumlah peserta dan kapasitas ruangan pada setiap kromosom.

IV. KESIMPULAN/RINGKASAN

Berdasarkan hasil dan analisis terhadap implementasi algoritma genetika pada sistem penjadwalan akademik di fakultas teknik universitas tanjungpura, dapat disimpulkan bahwa :

1. Analisis manual akan berbeda dengan hasil pengujian, karena proses komputasi algoritma genetika berdasarkan nilai acak yang dibangkitkan pada masing-masing kromosom.
2. Karakteristik bentrok yang sering terjadi di dalam komputasi algoritma genetika adalah karakteristik bentrok penyimpangan waktu.
3. Tingkat keberhasilan penjadwalan menggunakan algoritma genetika sangat bergantung pada rasio ruangan dan waktu terhadap rasio total jumlah sks.

DAFTAR PUSTAKA

[1] Marwana. 2012. "*Optimasi Penjadwalan Mata Kuliah Menggunakan Algoritma Genetika Berbasis Permintaan Mahasiwa*", Prosiding Konferensi Nasional Ilmu Komputer 2012.

[2] Teno Siswono dan Sarngadi Palgunadi, 2014. "*Analisa Kombinasi Algoritma Genetika dengan Algoritma Palgunadi untuk Penjadwalan Mata Kuliah di Universitas Sebelas Maret*", Prosiding SNST ke-5: Surakarta.

[3] Putra, Yendrika. 2009. "*Aplikasi Penjadwalan Perkuliahan Menggunakan Algoritma Genetika*". Riau.

[4] Maharani, Febria. 2013. "*Sistem Penjadwalan Proyek Menggunakan Algoritma Genetika*", Pekanbaru.

[5] Adriana Fanggidae dan Fadly Rano Lado. 2015. *Algoritma Genetika dan Penerapannya*. Yogyakarta: Teknosain.

[6] Abdul, Kadir. 2008. *Dasar Perancangan dan Implementasi Database Relasional.* Yogyakarta:Andi.

[7] Arief, M.Rudianto. 2011. *Pemrograman Web Dinamis Menggunakan PHP dan MySQL*. Yogyakarta: Andi.

[8] Larantika, Fitria. 2015. "*Sistem Penjadwalan Otomatis Menggunakan Algoritma Genetika di Fakultas Sains dan Teknologi"*, Yogyakarta.

[9] Hanita, Mery. 2011. "*Penerapan Algoritma Genetika pada Penjadwalan Mata Kuliah (Studi Kasus:Program Studi Matematika FMIPA Universitas Bengkulu*), Bengkulu.

[10] Kusumadewi, Sri. 2003. *Artificial Intelligence* (*Teknik dan Aplikasinya*). Graha Ilmu. Yogyakarta.

[11] Nidhra, Srinivas dan Dondeti, Jagruthi. 2012, Black Box And White Box Testing Techniques A Literature Review, International Journal Of Embedded Systems and Applications (IJESA) Vol.2, No.2,8-9.

[12] Suhartono, Entot. 2015. "*Optimasi Penjadwalan Mata Kuliah dengan Algoritma Genetika*". Vol 11, No 5:Semarang.

[13] Wiga Ayu P., Arif Junaidi, dan Retno Aulia V., 2013, "*Penjadwalan Mata Kuliah Menggunakan Algoritma Genetika di Jurusan Sistem Informasi ITS*", Vol 2. No.1:Surabaya.

Korespondensi: 20-02-2017; Revisi 20-02-2017; Submitted 20-02-2017; Revised 20-02-2017; Accepted 20-02-2017