PENGEMBANGAN APLIKASI PENYUSUNAN JADWAL PERKULIAHAN MENGGUNAKAN ALGORITMA GENETIKA

DEVELOPMENT OF A COURSE SCHEDULE PREPARATION APPLICATION USING GENETIC ALGORITHM

Isnaini Muhandhis[1*], Alven Safik Ritonga[2], Muhammad Shubhan[3]
E-mail: [1]imuhandhis@gmail.com, [2]alvensafik@uwp.ac.id, [3]20053028@student.uwp.ac.id

[1,2,3] Program Studi Teknik Informatika, Fakultas Teknik, Universitas Wijaya Putra

Abstrak

Mempersiapkan jadwal perkuliahan merupakan proses penting dalam administrasi kampus. Penyusunan jadwal harus memperhatikan keterbatasan yang ada seperti keterbatasan kelas, jam mengajar dan ketersediaan dosen. Algoritma genetik merupakan metode optimasi yang dapat menyelesaikan permasalahan penjadwalan. Algoritma genetika cukup baik dalam mengatur jadwal perkuliahan karena mampu menyelesaikan permasalahan dengan beberapa kriteria dan beberapa tujuan yang dimodelkan dalam proses evolusi. Penelitian ini bertujuan untuk membangun sebuah aplikasi pembangkitan jadwal perkuliahan secara otomatis dengan algoritma genetika. Aplikasi ini diharapkan dapat membantu proses administrasi penyusunan jadwal menjadi lebih cepat dan efisien. Hasil penelitian menunjukkan bahwa aplikasi berjalan dengan baik dan algoritma genetika mampu menyelesaikan permasalahan penjadwalan. Nilai parameter terbaik pada kasus ini adalah ukuran populasi 30, metode seleksi roda roulette, probabilitas mutasi 20% dan probabilitas kawin silang 20% dengan penemuan solusi tercepat pada generasi ke-37 dalam waktu 118 detik.

Kata kunci: roda roulete, kawin silang, mutasi, seleksi rangking, optimasi

Abstract

Preparing course schedules is an important process in campus administration. Schedule preparation must take into account existing limitations such as class limitations, teaching hours and lecturer availability. Genetic algorithm is an optimization method that can solve scheduling problems. Genetic algorithms are quite good at managing lecture schedules because they are able to solve problems with several criteria and several objectives that are modeled in the evolutionary process. This research aims to build an application to generate lecture schedules automatically with a genetic algorithm. This application is expected to help the administrative process of preparing schedules to be faster and more efficient. The research results show that the application runs well and the genetic algorithm is able to solve scheduling problems. The best genetic algorithm parameter values are population size 30, using roulette wheel selection method, mutation probability 20% and crossover probability 20% with the result of finding a solution in the 37th generation within 118 seconds.

Keywords: *roulete wheel, crossover, mutation, rank selection, optimization*

1. PENDAHULUAN

Penyusunan jadwal perkuliahan merupakan proses penting dalam administrasi kampus. Penyusunan jadwal membutuhkan waktu yang lama jika dilakukan secara manual. Hal ini karena ada batasan yang harus diperhatikan seperti jumlah ruang kelas, jam mengajar, dan ketersediaan dosen. Penyusunan jadwal merupakan suatu proses yang akan terus terjadi

dalam kegiatan kampus. Oleh karena itu, penting untuk menciptakan suatu sistem yang dapat membantu mempercepat dan mempermudah proses ini.
Pembuatan jadwal perkuliahan di Universitas 'ABC' disusun secara konvensional. Setiap program studi menata dan menyiapkan jadwalnya sendiri. Selain itu, ada mata kuliah umum di tingkat Universitas yang wajib diambil oleh semua mahasiswa di semua program studi. Hal ini membuat perlu waktu yang lama dalam menyusun dan mengkoordinasikan jadwal perkuliahan. Berdasarkan permasalahan tersebut, dibutuhkan sebuah program aplikasi yang bisa membantu otomatisasi penyusunan jadwal perkuliatan. Algoritma genetik merupakan salah satu metode optimasi yang dapat menyelesaikan permasalahan penjadwalan [1].

Metode algoritma genetika sering digunakan dalam berbagai bidang, termasuk bisnis dan teknik. Algoritma ini dimulai dengan kumpulan solusi yang disebut populasi. Konsep utama algoritma genetika adalah menemukan individu yang lebih baik selama proses evolusi. Proses evolusi meliputi persilangan dan mutasi kromosom. Proses c*rossover* atau kawin silang bertujuan untuk meningkatkan peluang munculnya individu yang lebih baik pada generasi berikutnya. Mutasi adalah proses perubahan struktur gen pada kromosom. Tujuan dari mutasi adalah untuk mencegah solusi menjadi optimum lokal [2].

Terdapat beberapa metode yang bisa digunakan untuk optimasi jadwal, misalnya simulasi anil dan pendakian bukit. Metode pendakian bukit menghindari traversal dan memilih beberapa node yang dipersenjatai dengan informasi untuk mencapai efisiensi. Namun, ketika data puncak datar dicari, arah pencarian yang optimal tidak dapat ditentukan, sehingga terjadi *random walk*, yang membuat efisiensi pencarian turun [3]. Metode simulasi-anil sangat cepat dalam mencari solusi optimum lokal, namun terdapat fenomena probabilitas tertentu gagal menemukan solusi optimal [3].

Adapaun algoritma genetik memiliki kelebihan yaitu hanya melakukan sedikit perhitungan matematis untuk menyelesaikan masalah [4]. Dengan proses crossover dan mutasi, algoritma genetik memiliki kemampuan yang lebih baik untuk keluar dari jebakan local optimum dibandingkan metode optimasi konvensional. Sedangkan kelemahannya adalah kinerja algoritma geentik sangat bergantung pada pemilihan parameter seperti ukuran populasi, tingkat crossover, tingkat mutasi, dan jumlah generasi. Pemilihan parameter yang salah dapat menyebabkan konvergensi lambat atau solusi yang kurang optimal [5].

Algoritma genetika mempunyai kinerja yang baik dalam penjadwalan karena mampu mengatasi permasalahan yang memiliki beberapa kualifikasi dan beberapa tujuan yang dimodelkan dalam proses evolusi [6]. Sistem penjadwalan dapat mendukung proses administrasi secara tepat sesuai dengan batasan yang diterapkan [7][8]. Berdasarkan penelitian tersebut, algoritma genetika mampu menyelesaikan masalah optimasi jadwal dengan baik. Oleh karena itu, penelitian ini berfokus pada penggunaan algoritma genetika dalam optimasi penjadwalan.

Penelitian ini bertujuan untuk mengembangkan aplikasi penjadwalan otomatis yang dapat memenuhi semua keterbatasan tersebut melalui proses optimasi. Optimasi sendiri merupakan suatu metode atau sistem yang dapat menyajikan hasil sesuai dengan batasan yang telah ditentukan dengan cepat dan mudah [9]

2. METODOLOGI

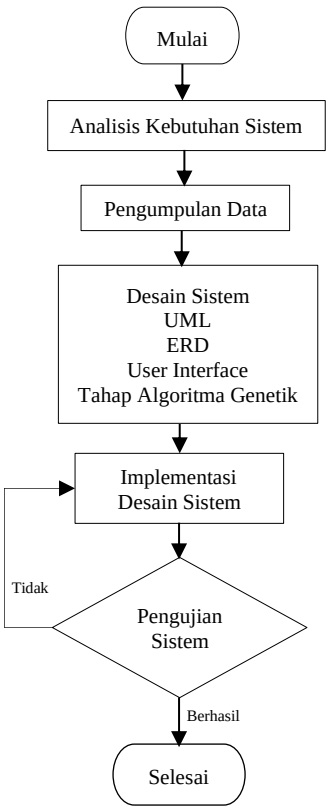
2.1 Pengembangan Aplikasi

Pengembangan aplikasi menggunkan metode SDLC Waterfall. Metode ini mengharuskan proses pengerjaan dilakukan secara berurutan atau bertahap. Keunggulan model Waterfall terletak pada kesederhanaannya dan mudah dipahami, karena setiap tahap telah didefinisikan dengan jelas, tersusun secara sistematis, serta didukung oleh dokumentasi

yang lengkap, sehingga mempermudah proses pemeliharaan [10]. Model ini juga sangat cocok untuk proyek berskala kecil. Terdapat metode SDLC lain yaitu Agile. Metode Agile dikembangkan untuk menangani permintaan perubahan yang cepat pada sistem oleh pengguna, mampu menangani biaya pengembangan yang besar, dan kurangnya efisiensi waktu ketika mengembangkan sebuah proyek. Metode ini dirancang untuk menangani proyek pengembangan sistem dalam waktu yang cepat [10].

Penelitian ini menggunakan metode Waterfall karena merupakan proyek berskala kecil. Setiap tahapan pengembangan sistem dalam penelitian ini dilakukan secara bertahap. Selama proses awal belum benar-benar valid, maka tidak akan melaju ke tahap berikutnya. Dengan demikian, pengerjaan sistem dapat didokumentasi secara lengkap, sehingga memudahkan pengelolaan kedepannya.

Tahapan metode waterfall diawali dengan analisa analisa kebutuhan pengguna. Hal ini berfungsi untuk menggali informasi tentang kebutuhan system secara spesifik. Tahap selanjutnya adalah perancangan desain program aplikasi berdasarkan kebutuhan pengguna. Selanjutnya, tahap implementasi yang merujuk pada pengkodean program menggunakan alat dan bahasa pemrograman yang sesuai. Tahap keempat adalah pengujian sistem untuk mengetahui apakah program telah sesuai dengan rancangan dan kebutuhan pengguna dan juga menelusuri *bug*. Tahap terakhir adalah pengoperasian program dan pemeliharaan. Alur pengembangan perangkat lunak dengan metode waterfall diatur secara runut dan sistematis. Metode penelitian dapat dilihat pada Gambar 1.



Gambar 1. Tahapan penelitian

2.2 Algoritma Genetik

Algoritma genetik merupakan salah satu algoritma pencarian acak dan dapat digunakan untuk proses optimasi. Algoritma genetik memeiliki beberapa tahapan antara lain

inisialisasi populasi awal, perhitungan fitness, proses seleksi, mutasi, kawin silang dan elitisme. Algoritma genetik mampu melakukan pencarian solusi dengan memaksimalkan atau meminimalkan fungsi yang diberikan. Pada awalnya, algoritma genetik mempunyai prosedur heuristic dan bersifat tidak menjamin menemukan solusi yang optimal. Akan tetapi, banyak penelitian menyimpulkan bahwa kinerja algoritma genetik sangat baik dalam menemukan solusi untuk berbagai masalah [11].

Tahapan Langkah algoritma genetik dalam penyelesaikan kasus ini dijelaskan sebagai berikut:

2.2.1 Representasi Kromosom

Jadwal mata kuliah direpresentasikan dengan deretan bilangan bulat dari 1 sampai jumlah mata kuliah di semester tersebut. Jam kosong direpresentasikan dengan bilangan 0. Setiap semester terdapat sekitar 25 - 30 mata kuliah. Terdapat empat ruang kelas untuk program studi Teknik Informatika, dengan tiga plot jam mengajar. Satu kromosom memiliki panjang N gen. Jumlah N diperoleh dari total jam mengajar selama seminggu, dimana terdapat lima hari kerja, tiga ruang kelas dan tiga plot jam pelajaran. Posisi setiap gen menentukan ruang kelas dan jam mengajar mata kuliah tersebut. Representasi kromosom jadwal pada penelitian dapat dilihat pada Gambar 2.

HARI	SENIN												SELASA								
KROMOSOM	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
RUANG	RUANG A			RUANG B			RUANG C			RUANG D			RUANG A			RUANG B			RUANG C		
JAM KE-	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
KODE MK	0	24	0	0	0	0	45	46	23	0	0	35	34	4	49	0	0	36	0	5	0

Gambar 2. Representasi kromosom jadwal kuliah

2.2.2 Pembangkitan populasi awal

Pembangkitan populasi awal dilakukan dengan cara memasukkan setiap kode mata kuliah ke dalam tipe data array dan sejumlah jam kosong sepanjang kromosom. Setelah itu, posisi setiap gen diacak menggunakan fungsi permutasi. Jumlah populasi awal yang dibangkitkan bervariasi mulai dari 30 sampai 1000. Setiap jumlah populasi awal akan dianalisa untuk mengetahui pengaruh jumlah populasi awal terhadap kecepatan penemuan solusi.

2.2.3 Perhitungan fitness

Setelah pembentukan kromoson, perhitungan nilai fitness dilakukan dengan cara mengecek susunan jadwal perkuliahan. Nilai fitness digunakan untuk mengukur seberapa bagus kromosom tersebut. Dalam penelitian ini, setiap pelanggaran *hard constraint* akan menambah nilai fitness sebanyak 1. Semakin banyak pelanggaran, maka semakin besar nilai fitnessnya. Adapun *hard constraint* yang diterapkan untuk menyusun jadwal program studi adalah sebagai berikut:

a) tiap ruang kelas hanya dapat digunakan oleh satu mata kuliah pada hari dan jam yang sama.
b) tiap dosen pengajar hanya dapat mengajar satu mata kuliah pada hari dan jam yang sama.
c) mata kuliah pada semester yang sama tidak dapat berada pada hari dan jam yang sama.
d) dosen W hanya dapat mengajar pada jam pertama.
e) dosen X hanya dapat mengajar pada jam kedua keatas.
f) dosen Y hanya dapat mengajar pada hari Senin, Selasa dan Rabu.
g) dosen Z tidak dapat mengajar pada hari Jumat.
h) mata kuliah MKDU di-plot pada tingkat Universitas yang jadwalnya telah ditetapkan sebelumnya.
i) Mata kuliah non-MKDU per prodi disusun agar tidak kres dengan jadwal MKDU

Dengan mekanisme tersebut, solusi terbaik yang dicari dalam penelitian ini adalah mencari gen yang memiliki nilai kebugaran (fitness) terkecil yaitu 0.

2.2.4 Proses seleksi

Proses seleksi adalah proses memilih dua buah kromosom sebagai calon gen orang tua untuk membentuk kromosom baru. Pada penelitian ini, analisis kinerja dilakukan terhadap metode seleksi rangking dan seleksi roda rolet. Seleksi ranking merupakan proses memilih gen dengan nilai terbaik dengan cara mengurutkan populasi berdasarkan nilai fitnessnya. Pada kasus ini, kegagalan dalam memenuhi limitasi jadwal yang telah ditetapkan akan menambah nilai fitness sebanyak satu. Gen atau individu terbaik adalah gen yang memiliki nilai fitness lebih kecil atau nol. Gen orang tua dipilih berdasarkan nilai fitness tertinggi. Adapun seleksi roda rolet adalah proses pemilihan gen orang tua diumpamakan seperti memutar roda rolet. Gen dengan nilai fitness lebih baik akan mendapat proporsi lebih besar terpilih [12].

2.2.5 Proses kawin silang

Proses kawin silang adalah proses tukar menukar gen antar dua *gen parent*. Hasil pertukaran gen ini akan membentuk komposisi kromosom baru. Kromosom baru tersebut diharapkan mempunyai nilai fitness yang lebih baik dari generasi sebelumnya. Pada penelitian ini, metode kawing silang yang digunakan adalah *Order Crossover*. Pada kasus *Travelling Salesman Problem* (TSP), setiap gen memiliki nilai unik yang dan tidak boleh ada duplikasi. Gen mata kuliah pada penelitian ini juga memiliki nilai kromosom yang unik, tidak ada mata kuliah yang sama sehingga metode kawin silang yang dapat digunakan salah satunya adalah *Order Crossover* [13]. Probabilitas kawing silang akan menentukan seberapa banyak susunan gen yang diubah. Penelitian ini akan menganalisis pengaruh probabilitas kawin silang terhadap penemuan solusi.

2.2.6 Proses mutasi

Konsep mutasi adalah proses menukar posisi antar gen di dalam sebuah kromosom. Penelitian ini menggunakan mutasi *Reciprocal Exchange Mutation*. *Reciprocal Exchange Mutation* akan memilih secara acak beberapa gen dalam satu kromosom, selanjutnya menukar letak dari tiap gen dalam kromosom. Jumlah gen dalam kromosom yang akan ditukar posisinya ditentukan dari probabilitas mutasi. Penelitian ini akan menganalisis pengaruh probabilitas mutasi terhadap kecepatan penemuan solusi.

2.2.7 Proses pembentukan populasi baru dan elitisme

Pembentukan populasi baru dalam proses evolusi dilakukan dengan cara menghapus sebagian kromosom awal, kromosom baru hasil evolusi berdasarkan nilai fitnessnya. Proses elitisme adalah sebuah proses yang mempertahankan gen-gen dengan nilai terbaik untuk tetap bertahan, tidak mengalami perubahan selama iterasi berlangsung [14]. Pada tahap ini, kromosom dengan nilai terbaik tetap dipertahankan dalam populasi sedangkan kromosom dengan nilai fitness rendah akan dihapus, sehingga terbentuklah susunan populasi baru. Proses elitisme memastikan bahwa kromosom terbaik tetap berada dalam populasi selama proses evolusi dilakukan berulang kali.

3. HASIL DAN PEMBAHASAN

3.1 Perancangan Sistem

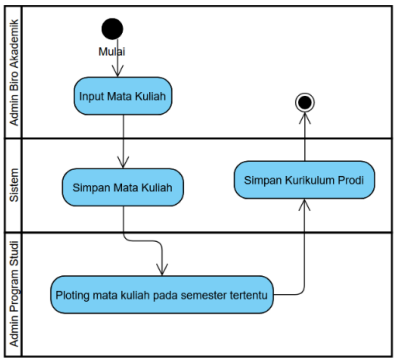
Rancangan *use case diagram* dapat dilihat pada Gambar 3. Sistem dirancang dengan tiga peran utama yaitu administrator, admin biro akademik dan admin program studi.

Administrator system mengelola proses pendaftaran akun, memasukkan data-data utama seperti nama program studi, ruang kelas, dan dosen. Admin biro akademik bertugas memasukan jadwal mata kuliah tidap program studi, data mata kuliah dasar umum (MKDU) di tingkat universitas dan menyusun jadwal MKDU dengan mempertimbangkan kurikulum tiap program studi dan jumlah mahasiswa yang akan mengambil mata kuliah tersebut. Setelah jadwal MKDU berhasil disusun, admin program studi bertugas menyusun data mata kuliah program studi dan ploting dosen pengampu di tiap semester dan mengenerate jadwal perkuliahan secara otomatis.

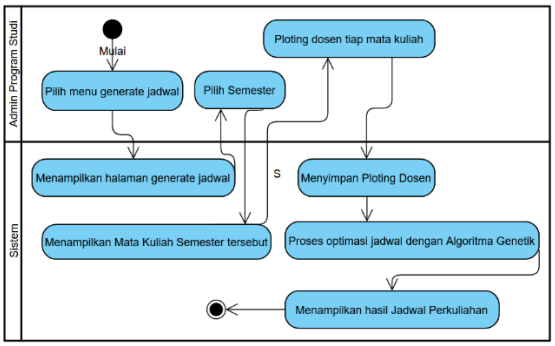
Alur penyusunan kurikulum program studi dapat dilihat pada Gambar 4. Admin biro akademik bertugas memasukkan data mata kuliah dari semua program studi, kemudian admin program studi dapat Menyusun susunan mata kuliah tiap semester dan disimpan dalam data kurikulum.

Alur untuk menyusun jadwal kuliah secara otomatis dapat dilihat pada Gambar 5. Admin program studi dapat melakukan ploting mata kuliah dan dosen pengampunya. Setelah itu, admin prodi dapat melakukan proses generate data jadwal berdasarkan ploting pada semester tersebut. Sistem akan mengolah data jadwal dengan algoritma genetik untuk mencari Solusi terbaik. Terakhir, hasil penyusunan jadwal dengan algoritma genetik ditampilkan di system.

Gambar 3. Use case diagram

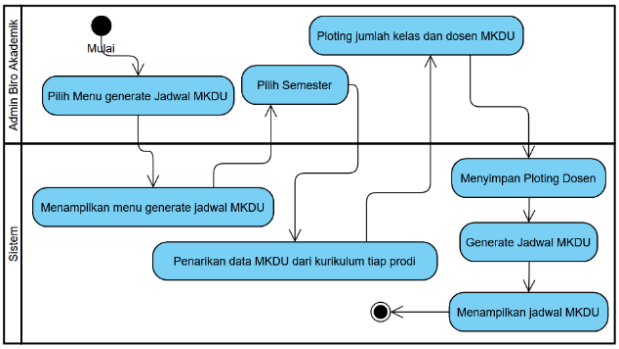


Gambar 4. Activity diagram penyusunan kurikulum program studi



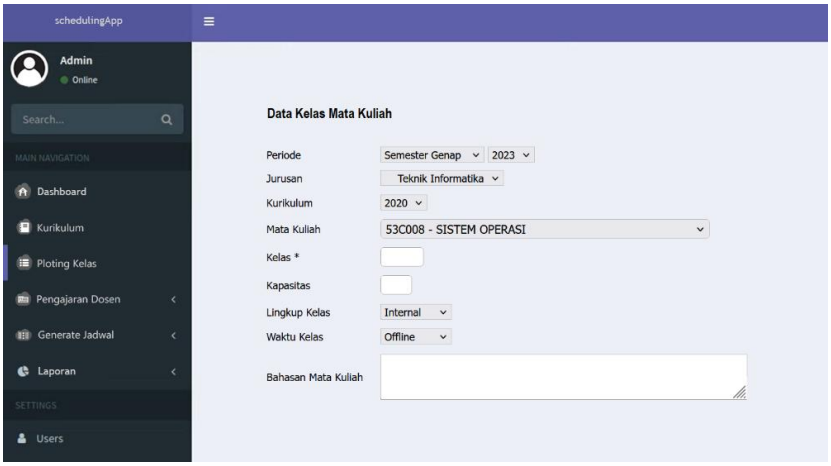
Gambar 5. Activity diagram generate jadwal program studi

Alur generate jadwal MKDU dapat dilihat pada Gambar 6. Admin biro akademik akan melakukan penarikan data kurikulum setiap program studi untuk melihat mata kuliah dasar umum (MKDU) yang ada di semester tersebut. Admin akan memploting jumlah kelas MKDU sesuai kebutuhan. Setelah selesai ploting mata kuliah dan dosen pengampu, maka data disimpan kemudian dilakukan generate jadwal secara otomatis. Hasilnya adalah jadwal MKDU yang sesuai dengan kebutuhan program studi.



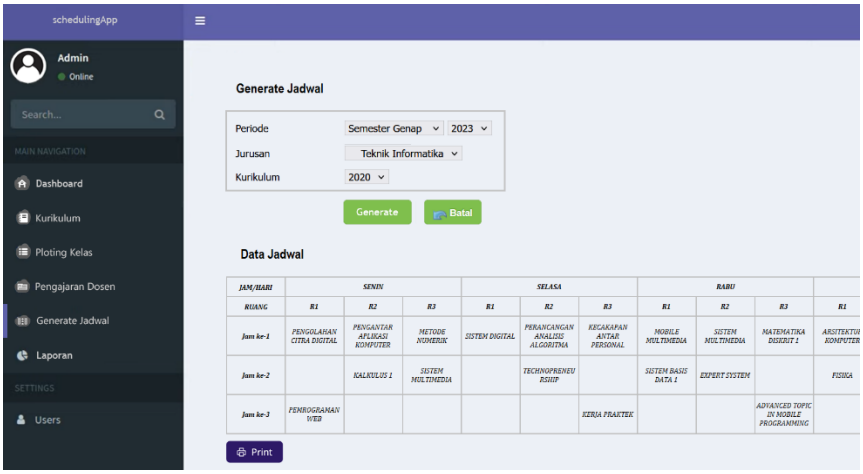
Gambar 6. Activity diagram generate jadwal MKDU di tingkat universitas

Gambar 7. Tampilan login aplikasi penjadwalan perkuliahan



Gambar 8. Tampilan ploting kelas tiap progam studi

Tampilan halaman login aplikasi dapat dilihat pada Gambar 7. Setiap akun memiliki user, password dan wewenang masing-masing. Tampilan halaman ploting kelas dapat dilihat pada Gambar 8. Setiap admin program studi memploting mata kuliah dengan dosen pengampu masing-masing. Tampilan halaman *generate* jadwal dapat dilihat pada Gambar 9. Setelah memploting kelas, admin program studi dapat melakukan proses otomasi penyusunan jadwal dengan memasukkan, semester, tahun, nama program studi dan kurikulum yang dipakai.



Gambar 9. Tampilan menu generate jadwal otomatis

3.2 Pengujian Kinerja Algoritma Genetik

Spesifikasi computer yang digunakan untuk menjalankan algoritma genetik akan mempengaruhi waktu yang dibutuhkan karena algoritma genetik membutuhkan komputasi yang tinggi. Spesifikasi yang digunakan dalam penelitian ini adalah sebagai berikut:

- Procesor Intel Core i5
- RAM 8 Gb
- SSD 512 Gb

Pengujian algoritma genetik dilakukan selama beberapa kali dengan mengubah beberapa parameternya. Pada pengujian ini, jika solusi tidak ditemukan lebih dari satu jam, maka pengujian dihentikan dan dicatat hasilnya sebagai solusi tidak ditemukan atau solusi tidak efektif karena waktu komputasi yang lama. Ringkasan hasil pengujian menggunakan metode seleksi *Roulete Wheel Selection* dapat dilihat pada Tabel 1.

Table 1. Hasil pengujian dengan metode seleksi roda rolet

Jumlah Populasi	Probabilitas Mutasi	Probabilitas Crossover	Jumlah Generasi Mendapat Solusi	Waktu Penemuan Solusi (detik)
30	20%	20%	37	118
100	20%	20%	18	224
1000	20%	20%	19	1575
30	50%	20%	177	541
100	50%	20%	272	2735
1000	50%	20%	-	> 3600
30	20%	50%	61	125
100	20%	50%	71	726
1000	20%	50%	7	500
30	50%	50%	-	> 3600
100	50%	50%	-	> 3600
1000	50%	50%	-	> 3600

Dari pengujian yang telah dilakukan membuktikan bahwa nilai parameter yang diberikan berpengaruh terhadap kinerja algoritma genetik. Terdapat 3 macam jumlah populasi yang digunakan, jumlah populasi 30 adalah yang paling sedikit pada percobaan ini, memiliki kinerja yang paling baik. Artinya, untuk menemukan solusi kita dapat memulai dari jumlah populasi yang sedikit. Membangkitkan populasi yang lebih sedikit memerlukan proses komputasi yang lebih rendah, sehingga penemuan solusi menjadi lebih cepat. Seperti terlihat pada Tabel 1, populasi yang lebih sedikit menemukan solusi pada generasi ke-37 dengan waktu komputasi yang lebih cepat dibanding ukuran populasi 100 dan 1000. Dengan populasi kecil, eksplorasi ruang solusi masih cukup efektif, tetapi dengan kecepatan yang lebih tinggi karena lebih sedikit individu yang perlu diproses dalam setiap generasi.

Pada kasus ini, seleksi roda rolet lebih cepat mampu menemukan solusi dengan baik. Hal ini sejalan dengan beberapa penelitian sebelumnya [15] dan [16]. Seleksi roda rolet memilih individu berdasarkan proporsi fitness, sehingga memberikan peluang kepada individu dengan fitness tinggi tanpa sepenuhnya mengabaikan individu dengan fitness rendah. Adapun seleksi ranking membatasi pemilihan gen parent hanya pada individu dengan fitness tertinggi dapat menyebabkan hilangnya keberagaman genetik lebih cepat, yang mungkin menyebabkan solusi yang kurang optimal. Dalam penelitian ini, pemilihan gen *parent* secara acak berdasarkan proporsi fitnessnya lebih baik kinerjanya dibanding pemilihan gen *parent* yang dibatasi pada sejumlah gen teratas dengan fitness terbaik.

Probabilitas *crossover* lebih berpengaruh daripada probabilitas mutasi dalam kasus ini. Proses *crossover* atau kawin silang adalah proses menukarkan gen antar dua gen parent. Proses ini memungkinkan pembentukan populasi baru yang lebih beragam, dibanding proses mutasi. Namun, pada kasus pm 50% dan pc 50%, di beberapa percobaan solusi tidak ditemukan dalam 1 jam pengujian. Artinya, tingkat keragaman yang tinggi dari generasi sebelumnya tidak menjamin ditemukan solusi optimal. Jika terlalu banyak perubahan terjadi dari generasi ke generasi, maka individu yang memiliki solusi baik dalam satu generasi dapat dengan mudah hilang karena perubahan besar yang terjadi pada generasi berikutnya. Hal ini dapat menyebabkan algoritma gagal menemukan solusi dalam waktu yang diberikan (1 jam pengujian), karena tidak ada konvergensi ke solusi yang lebih baik secara bertahap.

Pada dasarnya, algoritma genetik merupakan *heuristic search* yang masih bisa terjebak pada *local optima* [17]. Pada kasus ini, beberapa percobaan yang tidak dapat menemukan solusi kurang dari 1 jam terjadi karena *stuck* pada *local optima*. Pada percobaan tersebut, nilai fitness berada di angka 1 setelah belasan generasi, namun tidak mampu mencapai angka 0 hingga ratusan generasi berikutnya.

Algoritma genetik bekerja dengan proses evolusi untuk menemukan solusi. Setiap parameter yang digunakan akan memiliki kinerja yang berbeda untuk setiap kasus. Oleh karena itu, penggunaan algoritma genetik dalam setiap kasus harus melakukan beberapa percobaan parameter yang berbeda, sampai menemukan parameter terbaik yang hasilnya dirasa optimal. Dalam penelitian ini, nilai parameter terbaik adalah jumlah populasi 30, metode seleksi roda rolet, probabilitas mutasi 20% dan probablitas crossover 20% dengan hasil penemuan solusi di generasi ke-37 dalam waktu 118 detik.

3.3 Blackbox Testing

Hasil pengujian *blackbox* berdasarkan keluaran yang diharapkan dan hasil pengamatan program aplikasi dapat dilihat pada Tabel 2 berikut. Secara keseluruhan, aplikasi yang dikembangkan dapat berjalan dengan baik dan dapat menjalankan fungsi utamanya untuk menyusun jadwal kuliah secara otomatis dengan akurat.

Table 2. Hasil pengujian *blackbox* sistem

Pengujian	Hasil yang diharapkan	Hasil Pengamatan	Kesimpulan
Validasi login	User dapat masuk ke sistem jika memasukkan username dan kata sandi yang valid, namun gagal masuk ke sistem dan muncul pesan gagal login jika memasukkan username dan kata sandi yang tidak valid.	Sesuai	Berhasil
Validasi logout	User dapat keluar dari sistem dan kembali ke halaman login.	Sesuai	Berhasil
Kelola data mata kuliah	User yang berwenang dapat memasukkan, mengubah, melihat dan menghapus data mata kuliah	Sesuai	Berhasil
Kelola data kurikulum	User yang berwenang dapat memasukkan, mengubah, melihat dan menghapus data kurikulum	Sesuai	Berhasil
Kelola ploting kelas	User yang berwenang dapat memasukkan, mengubah, melihat dan menghapus data ploting kelas	Sesuai	Berhasil
Generate jadwal otomatis	Sistem dapat menghasilkan jadwal kuliah secara otomatis dan akurat	Sesuai	Berhasil
Tampilkan data jadwal	Sistem dapat menampilkan data jadwal dalam halaman html	Sesuai	Berhasil

4. KESIMPULAN

Berdasarkan hasil analisa dan pembahasan yang telah dilakukan, dapat ditarik beberapa kesimpulan berikut:

1) Aplikasi penjadwalan yang dikembangkan dapat berjalan dengan baik menyusun jadwal secara otomatis dan akurat dalam waktu relatif singkat.
2) Nilai parameter berpengaruh terhadap kinerja algoritma genetik, untuk menemukan solusi pada algoritma genetik, dapat dimulai dari jumlah populasi yang sedikit dulu. Jika

populasi dirasa kurang memberikan hasil, baru ditambahkan jumlah populasi secara bertahap.

3) Probabilitas *crossover* lebih berpengaruh terhadap kecepatan penemuan solusi daripada probabilitas mutasi.
4) Dalam penelitian ini, nilai parameter terbaik adalah jumlah populasi 30, metode seleksi roda rolet, probabilitas mutasi 20% dan probablitas crossover 20 % dengan hasil penemuan solusi di generasi ke-37 dengan waktu 118 detik.

5. DAFTAR RUJUKAN

[1] D. Wahyuningsih and E. Helmud, "Penerapan Algoritma Genetika Untuk Optimasi Penjadwalan pada MTS Negeri 1 Pangkalpinang," *Jurnal Sisfokom (Sistem Informasi Dan Komputer)*, vol. 9, no. 3, pp. 435–441, 2020, doi: https://doi.org/10.32736/sisfokom.v9i3.994.

[2] A. Lambora, K. Gupta, and K. Chopra, "Genetic algorithm-A literature review," in *2019 international conference on machine learning, big data, cloud and parallel computing (COMITCon)*, 2019, pp. 380–384. doi: https://doi.org/10.1109/COMITCon.2019.8862255.

[3] X. Chen, X. G. Yue, R. Y. Man Li, A. Zhumadillayeva, and R. Liu, "Design and Application of an Improved Genetic Algorithm to a Class Scheduling System," *International Journal of Emerging Technologies in Learning*, vol. 16, no. 1, pp. 44–59, 2020, doi: 10.3991/IJET.V16I01.18225.

[4] L. P. S. Ardiyani, "Perbandingan Algoritma Genetika dengan Algoritma Steepest Ascent Hill Climbing untuk Optimasi Penjadwalan Kuliah," *Jurnal Nasional Pendidikan Teknik Informatika: JANAPATI*, vol. 11, no. 1, pp. 63–73, 2022.

[5] R. Erama and R. Wardoyo, "Modifikasi Algoritma Genetika untuk Penyelesaian Permasalahan Penjadwalan Pelajaran Sekolah," *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, vol. 8, no. 2, pp. 111–120, 2014.

[6] J. Liu, Y. Liu, Y. Shi, and J. Li, "Solving resource-constrained project scheduling problem via genetic algorithm," *Journal of Computing in Civil Engineering*, vol. 34, no. 2, p. 4019055, 2020, doi: https://doi.org/10.1061/(ASCE)CP.1943-5487.00008.

[7] S. Parera, H. T. Sukmana, and L. K. Wardhani, "Application of genetic algorithm for class scheduling (Case study: Faculty of science and technology UIN Jakarta)," in *2016 4th International Conference on Cyber and IT Service Management*, 2016, pp. 1–5. doi: 10.1109/CITSM.2016.7577525.

[8] A. N. Toscany and R. Roestam, "Pengembangan Sistem Penjadwalan Kuliah Menggunakan Algoritma Genetik (Studi Kasus: Pascasarjana Universitas Jambi)," *Jurnal Manajemen Sistem Informasi*, vol. 2, no. 2, pp. 379–393, 2017, doi: http://dx.doi.org/10.11591/jurnalmsi.v12i4.xxxx.

[9] G. Zhang, Y. Hu, J. Sun, and W. Zhang, "An improved genetic algorithm for the flexible job shop scheduling problem with multiple time constraints," *Swarm Evol Comput*, vol. 54, p. 100664, 2020, doi: https://doi.org/10.1016/j.swevo.2020.100664.

[10] D. T. Haniva, J. A. Ramadhan, and A. Suharso, "Systematic Literature Review Penggunaan Metodologi Pengembangan Sistem Informasi Waterfall, Agile, dan Hybrid," *JIEET (Journal of Information Engineering and Educational Technology)*, vol. 7, no. 1, pp. 36–42, 2023.

[11] S. Katoch, S. S. Chauhan, and V. Kumar, "A review on genetic algorithm: past, present, and future," *Multimed Tools Appl*, vol. 80, pp. 8091–8126, 2021, doi: https://doi.org/10.1007/s11042-020-10139-6.

[12] M. Aziz, "Pemodelan Algoritma Genetika Pada Sistem Penjadwalan Perkuliahan Prodi Ilmu Komputer Universitas Lambungmangkurat," *KLIK-KUMPULAN JURNAL ILMU KOMPUTER*, vol. 1, no. 1, pp. 67–79, 2017.
[13] S. Wulandari, H. Helmi, and Y. Yudhi, "Penyelesaian Multiple Travelling Salesman Problem (Multi-TSP) dengan Metode Order Crossover dalam Algoritma Genetika (Studi Kasus: Data Pelanggan Agen Surat Kabar di Kota Singkawang)," *Bimaster: Buletin Ilmiah Matematika, Statistika dan Terapannya*, vol. 8, no. 2, 2019.
[14] R. S. Agustina *et al.*, "Prototype Sistem Optimasi Biaya Produksi Jamur Tiram dalam Sekali Produksi dengan Metode Algoritma Genetika," *JURNAL ILMIAH FAKULTAS ILMU KOMPUTER*, vol. 8, no. 2, 2019.
[15] P. Sharma and A. Wadhwa, "Analysis of selection schemes for solving an optimization problem in genetic algorithm," *Int J Comput Appl*, vol. 93, no. 11, 2014.
[16] W. Chinnasri and N. Sureerattanan, "Comparison of performance between different selection strategies on genetic algorithm with course timetabling problem," in *2010 IEEE International Conference on Advanced Management Science (ICAMS 2010)*, 2010, pp. 105–108.
[17] F. Gerges, G. Zouein, and D. Azar, "Genetic algorithms with local optima handling to solve sudoku puzzles," in *Proceedings of the 2018 international conference on computing and artificial intelligence*, 2018, pp. 19–22.