

Penerapan Algoritma Genetika Dalam Penjadwalan Mata Kuliah Program Studi Teknik Informatika Pada STMIK Widya Cipta Dharma

Andreas Johan ¹⁾, Hanifah Ekawati ²⁾, dan Wahyuni ³⁾

^{1,2,3}Teknik Informatika, STMIK Widya Cipta Dharma

^{1,2,3}Samarinda, 75123

E-mail: email1@domain.ekstensi¹⁾, email2@domain.ekstensi²⁾, email3@domain.ekstensi³⁾

ABSTRAK

Pelaksanaan perkuliahan program studi teknik informatika setiap awal semester selalu didahului dengan melakukan pemetaan mata kuliah, dosen, mahasiswa, waktu dan ruangan. Permasalahan klasik selalu muncul dalam proses pemetaannya, seperti sulitnya optimasi ruangan dan waktu karena banyak memiliki variabel kendala yang harus terpenuhi. Penelitian yang bertujuan untuk mengembangkan sistem informasi penjadwalan mata kuliah berbasis algoritma genetika di Program Studi Teknik Informatika ini berhasil menghasilkan sistem yang mampu menghasilkan jadwal perkuliahan secara otomatis dan lebih cepat dibandingkan cara manual. Hasil penelitian ini ditemukannya kombinasi parameter algoritma genetika terbaik untuk kasus ini adalah: jumlah populasi 30, jumlah maksimal generasi 5.000, probabilitas *crossover* 70-80%, dan probabilitas mutasi 15%. Kombinasi ini mampu menghasilkan jadwal yang memenuhi semua constraint dengan waktu estimasi 37 detik. Terbuktinya sistem informasi ini mampu mempermudah dan mempercepat proses penjadwalan kuliah dibandingkan dengan cara manual, yaitu hanya memakan waktu kurang dari 3 menit untuk menghasilkan jadwal optimal.

Kata kunci: Algoritma Genetika, *Waterfal*, STMIK Widya Cipta Dharma

Application of Genetic Algorithm in Scheduling Informatics Engineering Study Program Courses at STMIK Widya Cipta Dharma

ABSTRACT

The implementation of lectures in the Informatics Engineering Study Program at the beginning of each semester is always preceded by mapping courses, lecturers, students, time and rooms. Classic problems always arise in the mapping process, such as the difficulty of optimizing the room and time because it has many constraint variables that must be met. The research aimed at developing a genetic algorithm-based course scheduling information system in the Informatics Engineering Study Program succeeded in producing a system that was able to produce lecture schedules automatically and faster than the manual method. The results of this study found the best combination of genetic algorithm parameters for this case are: population size of 30, maximum number of generations of 5,000, crossover probability of 70-80%, and mutation probability of 15%. This combination is able to produce a schedule that meets all constraints with an estimated time of 37 seconds. The proof of this information system is able to simplify and speed up the lecture scheduling process compared to the manual method, which only takes less than 3 minutes to produce the optimal schedule.

Keywords: *Genetic Algorithm, Waterfal, STMIK Widya Cipta Dharma*

1. PENDAHULUAN

Pelaksanaan perkuliahan STMIK Widya Cipta Dharma setiap awal semester selalu didahului dengan melakukan pemetaan mata kuliah, dosen, mahasiswa, waktu dan ruangan. Permasalahan klasik selalu muncul dalam proses pemetaannya, seperti sulitnya optimasi ruangan dan waktu karena banyak memiliki variabel kendala yang harus terpenuhi. Disamping itu, beberapa kali terdapat mata kuliah wajib atau bahkan mata kuliah pilihan pada semester yang sama mengalami benturan penggunaan ruangan. Alokasi penggunaan ruangan perkuliahan dilakukan pada saat proses penjadwalan yang

masih belum optimal, menyebabkan beberapa mata kuliah mengalami benturan pelaksanaan.

Penjadwalan mata kuliah merupakan kegiatan yang sangat penting untuk dapat terlaksananya sebuah proses belajar mengajar yang baik bagi Program Studi Teknik Informatika pada STMIK Widya Cipta Dharma. Penjadwalan perkuliahan yang baik dapat membuat mahasiswa lebih rileks dan nyaman mendengarkan perkuliahan. Permasalahan utama dalam proses menyusun jadwal perkuliahan adalah bagaimana cara mengkombinasikan antara dosen, mata kuliah, ruang, waktu, dan kelas dengan baik tanpa terjadi konflik.

Walaupun sebetulnya masalah tersebut dapat diatasi dengan cara konvensional yaitu secara manual tetapi cara tersebut memerlukan waktu yang panjang serta harus dilakukan dengan teliti dan fokus agar hasil penyusunan jadwalnya tidak terjadi konflik.

Pada penelitian ini yang menjadi objek penelitian adalah Program Studi Teknik Informatika STMIK Widya Cipta Dharma. Tentunya merancang jadwal perkuliahan merupakan suatu pekerjaan yang memerlukan ketelitian dan keterampilan khusus, oleh karena itu diperlukan sistem penjadwalan mata kuliah yang bisa dilakukan secara otomatis dengan bantuan komputer yang dapat membuat jadwal dengan cepat, akurat dan efisien dan bisa diakses dengan lebih mudah melalui web.

Salah satu algoritma yang digunakan dalam menyelesaikan permasalahan penjadwalan adalah algoritma genetika, algoritma genetika telah terbukti sebagai metode optimisasi yang sangat efektif dalam menyelesaikan masalah-masalah yang kompleks dan sulit diselesaikan dengan metode konvensional (Wahyuningsih dan Helmud, 2020). Keunggulan algoritma genetika terletak pada kemampuannya untuk menjelajahi ruang pencarian yang luas, menemukan solusi optimal di tengah kompleksitas parameter-parameter yang terlibat, serta kemampuannya untuk menangani masalah yang melibatkan konstrain dan fungsi tujuan yang kompleks (Hidayat, dkk, 2019).

Berdasarkan permasalahan tersebut, dibutuhkan suatu sistem yang lebih efektif dan efisien dalam mengoptimalkan penjadwalan perkuliahan, maka penulis mencoba melakukan penelitian dengan judul “Penerapan Algoritma Genetika Dalam Penjadwalan Mata Kuliah Program Studi Teknik Informatika Pada STMIK Widya Cipta Dharma ” sebagai tujuan utama untuk dikembangkan lebih lanjut ke dalam suatu sistem terkomputerisasi yang dapat menggantikan cara manual penjadwalan perkuliahan sehingga penjadwalan lebih optimal.

2. RUANG LINGKUP

2.1 Rumusan Masalah

Berdasarkan latar belakang diatas, maka yang menjadi rumusan masalah dalam penelitian ini adalah sebagai berikut: “Bagaimana menerapkan Algoritma Genetika dalam penjadwalan mata kuliah Program Studi Teknik Informatika pada STMIK Widya Cipta Dharma”.

2.2 Batasan Masalah

1. Pembuatan aplikasi penjadwalan berbasis Web menggunakan bahasa pemrograman PHP dan database MySQL.
2. Sistem penjadwalan ini dibuatasi hanya untuk Kaprodi Teknik Informatika saja.
3. Sistem penjadwalan kuliah ini dibatasi untuk mata kuliah yang bersifat teori saja, tanpa jadwal praktikum.
4. Sistem penjadwalan kuliah ini dibatasi untuk mata kuliah kelas pagi dari jam 08:00 sampai 14:40 tanpa mata kuliah kelas malam.

5. Penelitian ini memakai jadwal kuliah Program Studi Teknik Informatika STMIK Widya Cipta Dharma semester ganjil tahun akademik 2023/2024.
6. Desain sistem yang digunakan dalam perancangan adalah Flowchart dan UML (Unified Modelling Language), yaitu Use Case Diagram, Activity Diagram.
7. Metode pengembangan aplikasi menggunakan metode Waterfall.
8. Pengumpulan data yang digunakan dalam penelitian adalah wawancara dan observasi.
9. Parameter-parameter dalam penjadwalan yang digunakan adalah mata kuliah yang diajarkan, sks, jumlah mahasiswa, dosen pengampu, kapasitas ruang kelas, jam, dan hari.

2.3 Tujuan Penelitian

Penelitian ini dilakukan dengan memiliki beberapa tujuan, yaitu:

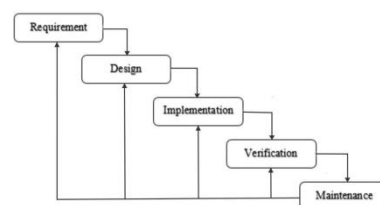
1. Menghasilkan sistem penjadwalan kuliah yang optimal.
2. Agar hasil penelitian ini dapat digunakan oleh Prodi Teknik Informatika untuk mempermudah proses penjadwalan dan mencegah konflik antar jadwal

2.4 Manfaat Penelitian

Adapun manfaat dari penelitian ini, terbangunnya sistem yang dapat membantu proses penjadwalan mata kuliah yang dilakukan di Program Studi Teknik Informatika untuk meningkatkan efektifitas dan efisiensi dalam pembuatan jadwal mata kuliah, sehingga dapat memberikan kemudahan dalam mengelola jadwal perkuliahan..

3. METODE PENGEMBANGAN *Waterfal*

Metode *waterfall* merupakan model pengembangan perangkat lunak yang sering digunakan. Model pengembangan ini melakukan pendekatan secara sistematis dan berurutan. Disebut *waterfall* karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan. Model pengembangan ini bersifat linear dari tahap awal pengembangan sistem yaitu tahap perencanaan sampai tahap akhir pengembangan sistem yaitu tahap pemeliharaan.



Gambar 1 Tahapan *Multimedea Development Life Cycle* (MDLC)

3.1 Analisis Kebutuhan (*Analysis Requirement*)

Analisis Kebutuhan (*Analysis Requirement*), tahap ini pengembang sistem diperlukan komunikasi yang bertujuan untuk memahami perangkat lunak yang diharapkan oleh pengguna dan batasan perangkat lunak

tersebut. Informasi dapat diperoleh melalui wawancara, diskusi atau survei langsung. Informasi dianalisis untuk mendapatkan data yang dibutuhkan oleh pengguna. Desain (*Design*)

3.2 Perancangan (*Design*)

Pada tahap ini, pengembang membuat desain sistem yang dapat membantu menentukan perangkat keras (*hardware*) dan sistem persyaratan dan juga membantu dalam mendefinisikan arsitektur sistem secara keseluruhan.

3.3 Implementasi (*Implementation*)

Pada tahap ini, sistem pertama kali dikembangkan di program kecil yang disebut unit, yang terintegrasi dalam tahap selanjutnya. Setiap unit dikembangkan dan diuji untuk fungsionalitas yang disebut sebagai unit testing, telah dikembangkan.

3.4 Pengujian (*Verification*)

Pada tahap ini, sistem dilakukan verifikasi dan pengujian apakah sistem sepenuhnya atau sebagian memenuhi persyaratan sistem, pengujian dapat dikategorikan ke dalam unit testing (dilakukan pada modul tertentu kode), sistem pengujian (untuk melihat bagaimana sistem bereaksi ketika semua modul yang terintegrasi) dan penerimaan pengujian (dilakukan dengan atau nama pelanggan untuk melihat apakah semua kebutuhan pelanggan puas).

3.5 Pemeliharaan (*Maintenance*)

Pemeliharaan (*Maintenance*), ini adalah tahap akhir dari metode waterfall. Perangkat lunak yang sudah jadi dijalankan serta dilakukan pemeliharaan. Pemeliharaan termasuk dalam memperbaiki kesalahan yang tidak ditemukan pada langkah sebelumnya.

4. PEMBAHASAN

4.1 Algoritma Genetika

Menurut Assagaf, dkk (2018), Algoritma genetika (AG) adalah suatu algoritma pencarian yang berbasis pada mekanisme seleksi alam dan genetika. Metode Algoritma Genetika ditemukan di Universitas Michigan, Amerika Serikat oleh Holland, (1975) melalui sebuah penelitian dan dipopulerkan oleh salah satu muridnya, Goldberg, (1989).

Menurut Suputra dan Pramarta (2020), Beberapa langkah-langkah yang perlu dilakukan dalam pengimplementasiannya, Berikut langkah-langkah yang perlu dilakukan:

1. Membangkitkan populasi awal dan kromosom
Proses membangkitkan sejumlah individu atau kromosom secara acak atau melalui prosedur tertentu. Ukuran untuk populasi tergantung pada masalah yang akan diselesaikan.
2. Evaluasi fungsi
Menurut Suzanti dan Mufarroha (2021), evaluasi fungsi merupakan proses untuk menghitung nilai *fitness* dari setiap kromosom yang ada dalam populasi. Tahap ini dilakukan setelah tahap populasi awal, di mana kualitas kromosom yang dihasilkan akan diuji dalam menyelesaikan masalah penjadwalan yang diberikan

3. Seleksi

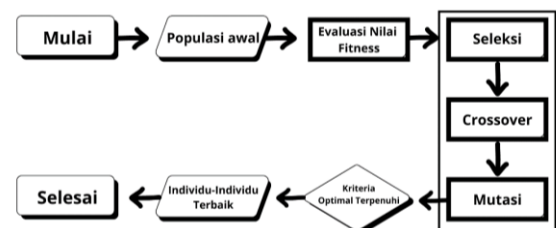
Menurut Suzanti dan Mufarroha (2021), Seleksi individu dalam populasi pada dasarnya adalah proses memilih individu dengan nilai probabilitas tinggi, yang kemungkinan besar akan dipilih pada langkah berikutnya.

4. *Crossover*

Menurut Prasetiawan (2015), pindah silang (*crossover*) adalah operator dari algoritma genetika yang melibatkan dua induk untuk membentuk kromosom baru. Pindah silang menghasilkan titik baru dalam ruang pencarian yang siap untuk diuji. Operasi ini tidak selalu dilakukan pada semua individu yang ada. Individu dipilih secara acak untuk dilakukan *crossing* dengan probabilitas *crossover* (P_c). Jika pindah silang tidak dilakukan, maka nilai dari induk akan diturunkan kepada keturunannya

5. Mutasi

Menurut Paranduk, dkk (2018), mutasi merupakan proses berubahnya komposisi gen pada kromosom itu sendiri. Peran dari proses mutasi adalah untuk menggantikan gen yang hilang dalam sebuah populasi yang diakibatkan dari proses seleksi dan *crossover*. Teknik mutasi yang digunakan yaitu dengan cara menukar dua posisi gen yang dipilih secara random yang biasa disebut dengan istilah *swapping mutation*. Pada mutasi juga terdapat *mutation rate* sebagai penentu berapa kromosom yang mengalami mutasi.



Gambar 2 Tahapan Algoritma Genetika

4.2 Penjadwalan

Menurut Sulaeman dan Nurjaman, (2019), Penjadwalan adalah aktivitas perencanaan untuk menentukan kapan dan dimana setiap operasi sebagai bagian dari pekerjaan secara keseluruhan harus dilakukan pada sumber daya yang terbatas, serta pengalokasian sumber daya pada suatu waktu tertentu dengan memperhatikan kapasitas sumber daya yang ada.

Menurut Lau Aplonia, (2023), Penjadwalan merupakan suatu proses pemanfaatan sumber daya yang ada untuk menyelesaikan kumpulan tugas dalam jangka waktu tertentu.

4.3 Hypertext Preprocessor (PHP)

Menurut Kusuma, dkk (2017), PHP adalah *script* berbasis server yang terintegrasi dengan HTML (*server side HTML embedded scripting*). PHP dapat digunakan untuk membuat berbagai aplikasi berbasis *web*, mulai dari halaman web sederhana hingga program yang lebih kompleks dan membutuhkan koneksi database. PHP



merupakan bahasa *scripting* yang paling banyak digunakan saat ini. PHP banyak digunakan untuk membuat *website* dinamis.

4.4 MySQL

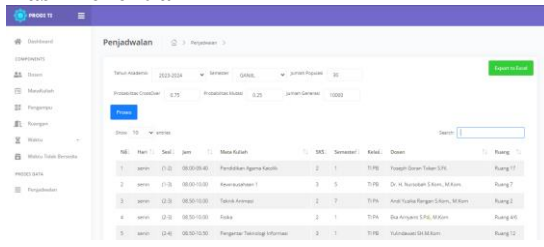
Menurut Ramadhani (2018), *MySQL* adalah perangkat lunak data berbasis server yang menggunakan sintaks *SQL* standar, memiliki basis pengguna yang besar, serta dapat memperoleh dan mengirimkan data dengan sangat cepat. *MySQL* didistribusikan di bawah lisensi GNU/GPL (*General Public License*). *MySQL* menggunakan bahasa *SQL* (*Structured Query Language*) yang menyediakan berbagai fitur dan kemampuan untuk mengakses, memanipulasi, dan menganalisis data, termasuk kemampuan untuk melakukan kueri (*query*) dan transaksi (*transaction*) data.

4.5 PhpMyAdmin

Menurut Hartati (2022), *PhpMyAdmin* adalah sebuah perangkat lunak gratis (*freeware*) yang ditulis menggunakan bahasa pemrograman PHP, yang digunakan untuk menangani administrasi *database MySQL* melalui interface *web*. *PhpMyAdmin* mendukung berbagai operasi di *database MySQL*, *Drizzle* dan *MariaDB*, seperti mengelola *database*, tabel, kolom, relasi, indeks, pengguna, izin, dan lainnya, *PhpMyAdmin* juga dapat digunakan untuk menjalankan *query MySQL*, memperbaiki, mengoptimalkan, dan melakukan perintah manajemen *database* lainnya.

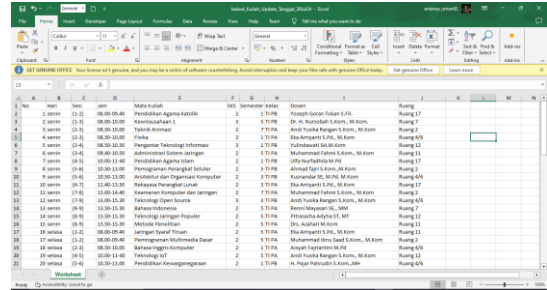
Berdasarkan uraian pendapat diatas dapat diartikan bahwa *PhpMyAdmin* digunakan untuk mengadministrasi *database MySQL* secara remote melalui *web* dengan cara visual dan mudah.

4.6 Hasil Penelitian



Gambar 3. Implementasi Tampilan Penjadwalan

Gambar diatas adalah form *Generate* jadwal dimana proses perhitungan algoritma akan dijalankan, dengan meng-input kan data-data sebelumnya ke dalam sistem. Sebelum jadwal di *Generate* sebelumnya harus mengisikan jumlah populasi yang dibangkitkan, jumlah generasi, probabilitas *crossover* dan probabilitas mutasi.



Gambar 4. Implementasi Hasil Generate Jadwal.

Setelah hasil dari generate atau optimasi maka akan keluar hasil dari penjadwalan algoritma genetika seperti yang diperlihatkan pada gambar 4. Setelah mendapatkan hasilnya data penjadwalan yang baru bisa kita export ke excel.

4.7 Uji Coba Algoritma Genetika

Untuk mengetahui waktu dan akurasi terbaik yang bisa diraih maka harus dilakukan uji coba sistem terlebih dahulu dengan melakukan 5x (lima kali) percobaan untuk setiap nilai variabel yang diinputkan sehingga diketahui rata-rata estimasi waktu dan nilai *fitness*-nya. Untuk percobaan pertama yang dilakukan adalah mencari nilai variabel terbaik yang sesuai dengan hard constraints dengan langkah pertama mencari nilai variabel maksimal generasi. Adapun hasil pengujian dari nilai variabel maksimal generasi menggunakan inputan user berdasarkan hard constraints yang dapat dilihat pada tabel 1.

Tabel 1. Hasil Pengujian Dari Nilai Variabel Maksimal Generasi Berdasarkan Hard constraints

No	Jml Pop Dibangkitkan	Maks Generasi	Prob. Crossover (pc)	Prob. mutasi (pm)	Estimasi Wkt	Keterangan
1	30	1000	75	25	4 m	Tidak Optimal [√]
2	30	5000	75	25	37s	Opsi Optimal
3	30	10000	75	25	41s	Opsi Optimal

Berdasarkan analisis dari Tabel 1, nilai variabel maksimal generasi yang memberikan hasil *fitness* terbaik adalah 10.000 generasi dengan estimasi waktu 41 detik, dan opsi optimal dengan keseimbangan terbaik antara hasil *fitness* dan waktu adalah 5.000 generasi dengan waktu 37 detik. Estimasi waktu terbaik yang dapat dicapai untuk opsi optimal adalah 30 detik. Setelah menentukan nilai maksimal generasi yang terbaik, langkah selanjutnya adalah menyesuaikan jumlah populasi yang dibangkitkan berdasarkan hasil pengujian yang disesuaikan dengan hard constraints dari inputan pengguna, seperti yang ditunjukkan dalam Tabel 2.

Tabel 2. Hasil Pengujian Dari Nilai Variabel Jumlah Populasi Dibangkitkan Berdasarkan Hard constraints

No	Jml Pop Dibangkitkan	Maks Generasi	Prob. Crossover (pc)	Pro. mutasi (pm)	Estimasi Wkt	Keterangan
1	20	10000	75	25	117.1 detik.	Opsi Optimal
2	30	10000	75	25	32 detik.	Opsi optimal

Berdasarkan Tabel 2 maka diketahui bahwa nilai variabel jumlah populasi dibangkitkan agar mendapatkan nilai *fitness* dan estimasi waktu terbaik adalah 30, dimana estimasi waktu terbaik yang bisa diraih dengan catatan opsi optimal adalah 32 detik. Setelah mengetahui nilai variabel jumlah populasi dibangkitkan yang terbaik, yang harus dilakukan adalah mencari nilai probabilitas *crossover* (pc). Adapun hasil pengujian dari nilai variabel probabilitas *crossover* (pc) menggunakan inputan user berdasarkan hard constraints yang dapat dilihat pada Tabel 3 berikut ini:

Tabel 3. Hasil Pengujian Dari Nilai Variabel Probabilitas Crossover (PC) Berdasarkan Hard Constraints

No	Jml Pop Dibangkitkan	Maks Generasi	Prob. Crossover (pc)	Prob. mutasi (pm)	Estimasi Wkt	Keterangan
1	30	10000	70	25	42 detik.	Opsi Optimal
2	30	10000	75	25	45 detik	Opsi Optimal
3	30	10000	80	25	42 detik.	Opsi Optimal

Berdasarkan hasil pengujian nilai variabel probabilitas *crossover* (pc) dengan jumlah populasi 30, maksimal generasi 10000, dan probabilitas mutasi 25%, ditemukan bahwa probabilitas *crossover* 70%, 75%, dan 80% semuanya merupakan opsi optimal dengan estimasi waktu masing-masing 42 detik, 45 detik, dan 42 detik. Dari ketiga opsi ini, probabilitas *crossover* 70% dan 80% memberikan estimasi waktu terbaik yaitu 42 detik. Oleh karena itu, disarankan menggunakan probabilitas *crossover* 70% atau 80% untuk mencapai hasil optimal dengan efisiensi waktu yang baik. Setelah mengetahui nilai variabel probabilitas *crossover* (pc) yang terbaik, yang harus dilakukan adalah mencari nilai probabilitas mutasi (pm).

Tabel 4. Hasil Pengujian Dari Nilai Variabel Probabilitas Mutasi (PM) Berdasarkan Hard

No	Jml Pop Dibangkitkan	Maks Generasi	Prob. Crossover (pc)	Pro. mutasi (pm)	Estimasi Wkt	Keterangan
1	30	10000	80	15	35 detik.	Opsi Optimal
2	30	10000	80	20	100 detik	Opsi Optimal

No	Jml Pop Dibangkitkan	Maks Generasi	Prob. Crossover (pc)	Pro. mutasi (pm)	Estimasi Wkt	Keterangan
3	30	10000	80	25	42 detik.	Opsi Optimal

Berdasarkan Tabel 4 yang memaparkan hasil pengujian nilai variabel probabilitas mutasi (pm) dengan jumlah populasi 30, maksimal generasi 10000, dan probabilitas *crossover* 80%, ditemukan bahwa semua opsi dengan probabilitas mutasi 15%, 20%, dan 25% adalah opsi optimal. Namun, estimasi waktu terbaik diperoleh pada probabilitas mutasi 15% dengan waktu 35 detik, diikuti oleh probabilitas mutasi 25% dengan waktu 42 detik, dan yang paling lambat adalah 20% dengan waktu 100 detik. Oleh karena itu, disarankan menggunakan probabilitas mutasi 15% untuk mencapai hasil optimal dengan efisiensi waktu terbaik.

4.8 Pengujian

4.8.1 Blackbox Testing

Pada penelitian ini, black box testing digunakan untuk mengevaluasi fungsionalitas dan user interface pengguna. Pengujian ini berguna untuk memastikan bahwa sistem telah berhasil beroperasi sebagaimana diharapkan oleh penulis. Berikut selengkapnya tentang black box testing:

Tabel 4. Pengujian Blackbox

No.	Menu	Fungsi dan Fitur	Hasil yang Diharapkan	Hasil Pengujian	
				Berhasil	Gagal
1	Login	Memvalidasi keberhasilan login pengguna	Pengguna berhasil login	√	
2	Dashboard	Menampilkan informasi jumlah dosen, jumlah mata kuliah, jumlah ruangan.	Informasi jumlah dosen, jumlah mata kuliah dan jumlah ruangan ditampilkan sesuai dengan data yang tersedia berhasil ditampilkan	√	



No.	Menu	Fungsi dan Fitur	Hasil yang Diharapkan	Hasil Pengujian	
				Berhasil	Gagal
3	Data Dosen	Menampilkan data dosen, prosen CRUD data dosen	Informasi data dosen dan proses CRUD berhasil dijalankan dan ditampilkan	√	
4	Data Matakuliah	Menampilkan data mata kuliah dan proses CRUD data mata kuliah	Informasi data mata kuliah dan proses CRUD berhasil dijalankan dan ditampilkan	√	
5	Data Pengampu	Menampilkan data pengampu dan proses CRUD data pengampu	Informasi data pengampu dan proses CRUD berhasil dijalankan dan ditampilkan	√	
6	Data Ruangan	Menampilkan data ruangan dan proses CRUD data ruangan	Informasi data ruangan dan proses CRUD berhasil dijalankan dan ditampilkan	√	

No.	Menu	Fungsi dan Fitur	Hasil yang Diharapkan	Hasil Pengujian	
				Berhasil	Gagal
7	Data Jam	Menampilkan data jam dan proses CRUD data jam	Informasi data jam dan proses CRUD berhasil dijalankan dan ditampilkan	√	
8	Data Hari	Menampilkan data hari dan proses CRUD data hari	Informasi data hari dan proses CRUD berhasil dijalankan dan ditampilkan	√	
9	Waktu Tidak Bersedia	Memilih waktu tidak bersedia dosen	Dapat memilih waktu tidak bersedia dosen dan menyimpan data	√	
10	Penjadwalan	Menampilkan jadwal perkuliahan dan <i>export</i> jadwal	Data jadwal dapat ditampilkan dan diexport ke excel	√	

4.8.2 Pengujian White Box

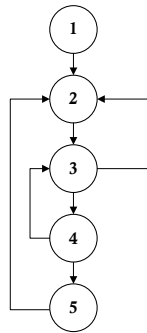
Pada penelitian ini *whitebox testing* digunakan untuk menguji dan menganalisis kode program. Berikut selengkapnya tentang *whitebox testing*:

```

111 // fungsi untuk inialisasi populasi individu secara acak
112 public function Inialisasi()
113 {
114     // Menghitung jumlah pengampu, jam, hari, ruang besar, dan ruang kecil
115     $jumlah_pengampu = count($this->pengampu);
116     $jumlah_jam = count($this->jam);
117     $jumlah_hari = count($this->hari);
118     $jumlah_ruang_besar = count($this->ruangbesar);
119     $jumlah_ruang_kecil = count($this->ruangkecil);
120
121     // Membuat populasi individu secara acak
122     for ($i = 0; $i < $this->populasi; $i++) {
123         $indiv = array();
124         $indiv['pengampu'] = $i;
125         $indiv['jam'] = mt_rand(0, $jumlah_jam - 50); // Memastikan jam secara acak berdasarkan 50
126         $indiv['hari'] = mt_rand(0, $jumlah_hari - 1); // Memastikan hari secara acak
127         $indiv['ruang'] => ($this->ruangbesar[$i] > 400 ? $this->ruangbesar[mt_rand(0, $jumlah_ruang_besar - 1)] : $this->ruangkecil[mt_rand(0, $jumlah_ruang_kecil - 1)]); // Memastikan ruang berdasarkan jumlah mahasiswa
128     }
129 }
130
131 }

```

Gambar 5 Inialisasi dan Representasi Kromosom



Gambar 6. Flowgraph Inisialisasi dan Representasi Kromosom

Berdasarkan *flow graph* pada gambar 4.13, diketahui bahwa terdapat 5 *node* (*n*) dan 4 *edge* (*e*). Sehingga nilai CC adalah $5 - 4 + 2 = 3$. Berikut merupakan jalur independen yang didapatkan:

1. Path 1 = 1-2-3-4-3-4
2. Path 2 = 1-2-3-4-3-2-3-4
3. Path 3 = 1-2-3-4-3-2-5

Tabel 5. Test Case Inisialisasi dan Representasi Kromosom

Kondisi	Hasil yang Diharapkan	Keterangan
Menghitung jumlah pengampu, jam, hari, ruang besar, dan ruang kecil	Sistem berhasil menghitung jumlah pengampu, jam, hari, ruang besar, dan ruang kecil	Berhasil
Perulangan populasi atau individu dan perulangan pengampu.	Sistem berhasil melakukan perulangan individu atau populasi dan perulangan pengampu	Berhasil
Proses inisialisasi dan representasi kromosom secara acak dengan mengambil data waktu, hari dan ruang	Sistem berhasil melakukan Proses inisialisasi dan representasi kromosom secara acak dengan mengambil data waktu, hari dan ruang	Berhasil

5. KESIMPULAN

Dari semua uraian dan pembahasan maka diambil beberapa kesimpulan sebagai berikut:

1. Dari hasil pengujian sistem dapat memberikan kemudahan dan kecepatan kepada user atau Program Studi Teknik Informatika (*Admin*) dalam proses pembuatan atau penyusunan jadwal untuk perkuliahan, yaitu hanya diperlukan waktu sekitar 40 detik sampai 3 menit dibandingkan dengan proses manual yang memerlukan waktu yang lama.
2. Sistem dapat menghasilkan jadwal perkuliahan, sehingga mendapatkan kombinasi terbaik antara

mata kuliah, dosen yang mengajar, mahasiswa yang mengambil mata kuliah, dan ruangan yang tersedia.

3. Dari uji coba algoritma genetika, diperoleh bahwa nilai maksimal generasi terbaik adalah 10.000 dengan estimasi waktu 41 detik, sementara opsi optimal dengan keseimbangan terbaik adalah 5.000 generasi dengan waktu 37 detik. Nilai jumlah populasi optimal adalah 30, dengan estimasi waktu terbaik 32 detik. Untuk probabilitas *crossover* (*pc*), nilai optimal ditemukan pada 70% dan 80% dengan waktu 42 detik. Sedangkan untuk probabilitas mutasi (*pm*), nilai optimal adalah 15% dengan waktu 35 detik. Kombinasi terbaik untuk mencapai efisiensi waktu dan nilai *fitness* optimal adalah dengan jumlah populasi 30, probabilitas *crossover* 70% atau 80%, dan probabilitas mutasi 15%.

6. SARAN

Berdasarkan kesimpulan yang telah diuraikan, terdapat beberapa saran yang dapat diberikan untuk pengembangan lebih lanjut dan penerapan hasil penelitian ini:

1. Disarankan untuk penelitian selanjutnya dapat digabungkan dengan metode lain untuk memberikan hasil penjadwalan kuliah yang dapat menerapkan penjadwalan yang optimal.
2. Disarankan untuk kedepannya penelitian ini dapat mengikuti format jam perkuliahan yang digunakan oleh program studi teknik informatika.
3. Disarankan untuk kedepannya penelitian ini dapat menambah aturan jam sesi istirahat, sehingga tidak ada perkuliahan di jam istirahat.

7. DAFTAR PUSTAKA

- Assagaf, A., Ibrahim, A., & Suranto, C. (2018). *Membangun Sistem Informasi Penjadwalan Dengan Metode Algoritma Genetika Pada Laboratorium Teknik Informatika Universitas Muhammadiyah Maluku Utara*.
- Hartati, E. (2022). *Sistem Informasi Transaksi Gudang Berbasis Website Pada Cv. Asyura*. 3(1).
- Kusuma, A. S., Gede, I., & Putra, S. E. (2017). Rancang Bangun Sistem Penjadwalan Wawancara Mahasiswa Baru Stmik Stikom Indonesia. *Jurnal Ilmiah Teknologi Informasi Asia*, 11(2).
- Lau Aplonia, E. (2023). Efektivitas Dan Efisiensi Melalui Pemanfaatan Runout Time(Rot) Method. *Jurnal Exchall*, Vol.5, No. 1.
- Paranduk, L., Indriani, A., Hafid, M., & Suprianto. (2018). Sistem Informasi Penjadwalan Mata Kuliah Menggunakan Algoritma Genetika Berbasis Web. In *Seminar Nasional Aplikasi Teknologi Informasi (Snati)*.



- Prasetiawan, E. (2015). *Aplikasi Penjadwalan Mata Pelajaran Pada Smk Negeri 1 Tenggarong Menggunakan Algoritma Genetika Berbasis Intranet*.
- Putu Gede Hendra Suputra, I., & Rai Adi Pramatha, C. (2020). Rekomendasi Rute Perjalanan Wisata Berbasis Web Menggunakan Algoritma Genetika. *Jurnal Ilmu Komputer*, 13(1), 21. <https://doi.org/10.24843/Jik.2020.V13.I01.P03>
- Ramadhani, I. A. (2018). Pengembangan Sistem Informasi Penjadwalan Mata Kuliah Berbasis Web Di Fakultas Teknik Universitas Negeri Makassar. *Jurnal Pendidikan*, 6(2).
- Setiawati Sulaeman, F., & Firaz Fakhri Nurjaman, M. (2019). Aplikasi Penjadwalan Dan Booking Online Menggunakan Teknologi Android Webview. *Media Jurnal Informatika*, 11(2). <http://jurnal.unsur.ac.id/mjinformatika>
- Suzanti, I. O., & Mufarroha, F. A. (2021). Implementasi Relevant Feedback Menggunakan Algoritma Genetika Pada Dokumen Bahasa Indonesia Implementation Of Relevant Feedback Using Genetic Algorithm In Indonesian Documents. In *Jurnal Ilmu Pengetahuan Dan Teknologi Komunikasi* (Vol. 23, Issue 2).