

Optimasi Pembuatan Jadwal Perkuliahan Menggunakan Algoritma Genetika Berbasis Pendekatan Multivariat

Muhamat Abdul Rohim¹, Ferry Wiranto², Difari Afreyna Fauziah³

¹Sistem dan Teknologi Informasi, Institut Teknologi dan Sains Mandala, muhamatabdulrohim@itsm.ac.id

²Rekayasa Perangkat Lunak, Institut Teknologi dan Sains Mandala, ferry@itsm.ac.id

³Sistem dan Teknologi Informasi, Institut Teknologi dan Sains Mandala, difariafreyna@itsm.ac.id

Keywords

Course Scheduling,
Genetic Algorithm,
Schedule Optimization,
Academic System,
Scheduling Conflicts,

ABSTRACT

Course scheduling is a crucial aspect of academic management in higher education institutions. Manual scheduling processes often face various challenges, such as limited classroom availability, lecturer preferences, and uneven schedule distribution. This research aims to optimize the course scheduling process using the Genetic Algorithm (GA) to improve efficiency and reduce scheduling conflicts. The data used in this study includes 20 classrooms, 50 lecturers, and an average of 120 course schedules per semester. The system implementation was carried out using PHP programming language, following research stages that include data collection, constraint analysis, algorithm design, implementation, and result evaluation. The findings indicate that the GA-based system produces a more balanced course schedule, with a processing time of approximately 1 to 5 minutes and zero scheduling conflicts. Thus, this approach is proven to be more effective than the previously used manual method.

Kata Kunci

Penjadwalan Perkuliahan,
Algoritma Genetika,
Optimasi Jadwal,
Sistem Akademik,
Konflik Jadwal,

ABSTRAK

Penjadwalan perkuliahan merupakan salah satu aspek penting dalam manajemen akademik di perguruan tinggi. Proses penjadwalan yang dilakukan secara manual sering kali menghadapi berbagai kendala, seperti keterbatasan ruang, preferensi dosen, serta distribusi jadwal yang tidak merata. Penelitian ini bertujuan untuk mengoptimalkan proses penjadwalan perkuliahan menggunakan Algoritma Genetika (AG) agar lebih efisien dan mengurangi konflik jadwal. Data yang digunakan dalam penelitian ini meliputi 20 ruang kelas, 50 dosen, serta rata-rata 120 jadwal kuliah per semester. Implementasi sistem dilakukan menggunakan bahasa pemrograman PHP, dengan tahapan penelitian meliputi pengumpulan data, analisis kendala, perancangan algoritma, implementasi, dan evaluasi hasil. Hasil penelitian menunjukkan bahwa sistem berbasis AG mampu menghasilkan jadwal perkuliahan yang lebih merata, dengan waktu pemrosesan sekitar 1 hingga 5 menit dan tanpa adanya konflik jadwal. Dengan demikian, pendekatan ini terbukti lebih efektif dibandingkan metode manual yang sebelumnya digunakan.

Korespondensi Penulis:

Muhamat Abdul Rohim,
Institut Teknologi dan Sains Mandala, Jember
Telepon : +6285604946242
Email: muhamatabdulrohim@itsm.ac.id

Submitted : 03-03-2025; Accepted : 06-03-2025;
Published : 06-03-2025

Copyright (c) 2025 The Author (s) This article is distributed under a Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0)

1. PENDAHULUAN

Penjadwalan perkuliahan merupakan elemen krusial dalam manajemen akademik di perguruan tinggi. Sistem penjadwalan yang efektif dapat meningkatkan efisiensi dalam penggunaan ruang kelas, alokasi tenaga pengajar, serta kenyamanan mahasiswa dalam mengikuti perkuliahan. Namun, banyak institusi pendidikan masih bergantung pada metode manual dalam menyusun jadwal. Tudose [1] dalam penelitiannya menyebutkan bahwa pendekatan secara manual akan membutuhkan waktu yang lama, rawan terjadi kesalahan, dan jadwal yang dihasilkan sering kurang optimal. Dengan meningkatnya kompleksitas akademik, metode manual menjadi semakin tidak efektif dalam memenuhi kebutuhan penjadwalan yang optimal.

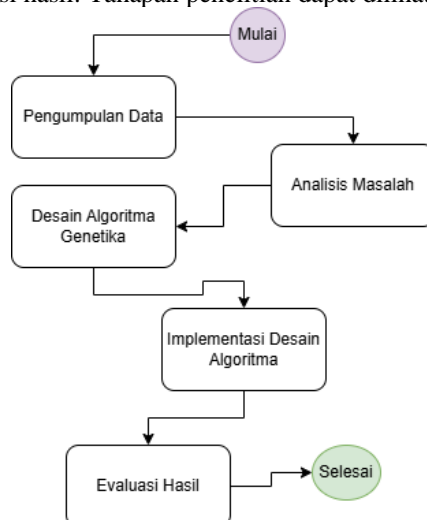
Berbagai penelitian telah berupaya mengatasi permasalahan dalam penjadwalan perkuliahan dengan menerapkan pendekatan berbasis algoritma komputasional. Salah satu metode yang banyak digunakan adalah Algoritma Genetika (AG), yang dikenal mampu menemukan solusi optimal secara heuristik. Lestari et al. [2]

mengembangkan sistem penjadwalan berbasis AG di Jurusan Teknik Elektro Universitas Riau. Studi ini menunjukkan bahwa AG dapat menghasilkan jadwal yang lebih terstruktur dengan distribusi waktu mengajar yang lebih merata bagi dosen, meskipun masih menghadapi tantangan dalam optimalisasi penggunaan ruang kuliah. Hikmawan [3] dalam penelitiannya menerapkan AG dengan teknik mutasi terbatas untuk penjadwalan di UNISMA Bekasi. Hasilnya mengindikasikan bahwa pendekatan ini lebih adaptif dalam menangani berbagai kendala dibandingkan metode manual, sehingga menghasilkan jadwal yang lebih fleksibel dan optimal. Sementara itu, penelitian oleh Wiratna et al. [4] mengembangkan aplikasi berbasis web yang memanfaatkan AG untuk menyusun jadwal perkuliahan. Hasilnya menunjukkan pengurangan signifikan dalam konflik jadwal, dengan nilai fitness mencapai 1 dalam waktu pemrosesan rata-rata 2,8 menit. Selain itu, A. Amrullah [5] mengkombinasikan AG dengan pencarian tabu dalam penelitiannya, hasilnya menunjukkan bahwa AG jadwal yang dihasilkan lebih optimal. Di sisi lain, penelitian oleh Sinaga et al. [6] yang diterapkan pada Program Studi Matematika FMIPA Universitas Sumatera Utara membuktikan bahwa AG dapat mengurangi konflik penjadwalan, meningkatkan efisiensi penggunaan ruang kuliah, serta mempermudah revisi jadwal jika diperlukan. Budi et al. [7] menyoroti pentingnya kombinasi Algoritma Genetika dengan metode lain seperti Simulated Annealing, untuk meningkatkan kualitas hasil penjadwalan perkuliahan. Sementara itu, studi yang dilakukan oleh Aulia et al. [8] membuktikan bahwa penggunaan AG dengan parameter adaptif dapat meningkatkan efisiensi dalam pencarian solusi optimal untuk sistem penjadwalan di universitas dengan skala besar. Siregar et al. [9] mengusulkan pendekatan hybrid antara AG dan Particle Swarm Optimization (PSO) sebagai solusi terhadap tantangan kompleks dalam penjadwalan akademik. Pendekatan ini terbukti mampu mengurangi waktu pemrosesan dibandingkan penggunaan AG secara mandiri. Handoko dan Prasetyo [10] juga meneliti penerapan AG dalam penjadwalan perkuliahan berbasis preferensi dosen. Hasil penelitian mereka menunjukkan bahwa metode ini dapat menghasilkan jadwal dengan tingkat kepuasan lebih tinggi dibandingkan dengan metode konvensional. Selain itu, penelitian yang dilakukan oleh Yuliana et al. [11] menunjukkan bahwa penerapan AG dalam sistem berbasis cloud computing dapat meningkatkan efisiensi dalam penjadwalan akademik, dengan hasil yang lebih baik dalam mengurangi konflik dibandingkan dengan implementasi berbasis lokal.

Meskipun berbagai penelitian telah menunjukkan efektivitas Algoritma Genetika dalam optimasi penjadwalan perkuliahan, sebagian besar masih terbatas dalam mempertimbangkan beberapa variabel spesifik, seperti ketersediaan dosen dan ruang kelas. Oleh karena itu, diperlukan pendekatan yang lebih komprehensif dengan mempertimbangkan berbagai faktor secara simultan agar dapat menghasilkan solusi penjadwalan yang lebih optimal. Penelitian ini bertujuan untuk mengoptimalkan proses penjadwalan perkuliahan menggunakan Algoritma Genetika berbasis pendekatan multivariat. Dengan memperhitungkan faktor-faktor seperti jumlah mahasiswa, preferensi dosen, kapasitas ruang, serta alokasi waktu, sistem yang dikembangkan diharapkan dapat menghasilkan jadwal yang lebih adaptif dan efisien dibandingkan metode yang telah ada. Secara keseluruhan, penerapan Algoritma Genetika dalam penjadwalan perkuliahan terbukti dapat meningkatkan efisiensi dan efektivitas dibandingkan dengan metode manual. Namun, tantangan dalam mempertimbangkan variabel yang lebih kompleks masih menjadi permasalahan yang perlu diatasi. Dengan pendekatan berbasis multivariat, penelitian ini diharapkan dapat menghasilkan solusi penjadwalan yang lebih optimal, mengurangi konflik jadwal, serta meningkatkan kualitas manajemen akademik di perguruan tinggi.

2. METODE PENELITIAN

Metodologi penelitian ini bertujuan untuk menjelaskan langkah-langkah yang akan diambil dalam implementasi algoritma genetik untuk pembuatan jadwal perkuliahan. Proses ini melibatkan beberapa tahap, mulai dari pengumpulan data hingga evaluasi hasil. Tahapan penelitian dapat dilihat pada Gambar 1.



Gambar 1 Tahapan Penelitian

2.1 Pengumpulan Data

Pengumpulan data merupakan langkah awal yang penting dalam penelitian ini. Penelitian ini mengambil contoh data pada Institut Teknologi dan Sains Mandala, data yang diperlukan meliputi data dosen, data mahasiswa, data ruang kelas, dan data waktu yang tersedia. Data dosen berupa Nama dan Matakuliah yang diajarkan, sedangkan data mahasiswa berupa jumlah mahasiswa dalam satu kelas. Data ruang kelas berupa jumlah ruang kelas yang tersedia, kapasitas, serta fasilitas didalamnya, terakhir adalah data waktu berupa jam yang menunjukkan minimal dan maksimal perkuliahan dilaksanakan. Data ini dapat diperoleh melalui survei, wawancara, dan dokumen resmi dari institusi pendidikan.

2.2 Analisis Masalah

Setelah data terkumpul, langkah selanjutnya adalah menganalisis masalah penjadwalan. Analisis dilakukan pada kendala yang ada dalam pembuatan jadwal secara manual, dan bagaimana solusi yang dapat dilakukan untuk mengatasi kendala tersebut. Kendala yang terjadi dapat berupa batasan waktu, maupun ketersediaan dosen.

2.3 Desain Algoritma Genetika

Desain algoritma genetika meliputi beberapa komponen penting, seperti representasi kromosom, fungsi fitness, dan operator genetika seperti seleksi, persilangan, maupun mutase. Kromosom dalam hal ini - merepresentasikan jadwal perkuliahan. Setiap gen dalam kromosom akan mewakili satu mata kuliah yang dijadwalkan. Fungsi ini digunakan untuk mengevaluasi kualitas jadwal yang dihasilkan. Kriteria yang digunakan dalam fungsi fitness dapat mencakup jumlah bentrok, ketersediaan dosen, dan jumlah mahasiswa. Operator genetik yang digunakan seperti seleksi, persilangan, dan mutasi akan digunakan untuk menghasilkan generasi baru dari jadwal perkuliahan.

2.4 Implementasi Desain Algoritma

Setelah desain algoritma selesai, langkah selanjutnya adalah implementasi algoritma genetika. Proses ini meliputi proses pengkodean (*coding*) dalam bahasa pemrograman dan proses pengujian terhadap algoritma yang sudah dibuat.

2.5 Evaluasi Hasil

Setelah implementasi, hasil jadwal perkuliahan yang dihasilkan oleh algoritma genetika akan dievaluasi. Evaluasi ini mencakup tiga aspek utama, meliputi perbandingan dengan jadwal manual, dan pengukuran efisiensi waktu dan sumber daya yang digunakan dalam penyusunan jadwal.

3. HASIL DAN ANALISIS

Setiap tahapan penelitian ini memiliki peran penting dalam memastikan bahwa sistem yang dikembangkan dapat menghasilkan jadwal perkuliahan yang optimal dan sesuai dengan kebutuhan akademik. Pengumpulan data dilakukan untuk memperoleh informasi yang diperlukan dalam penyusunan jadwal, seperti daftar mata kuliah, jumlah mahasiswa, serta kapasitas ruang. Selanjutnya, analisis kendala bertujuan untuk mengidentifikasi batasan-batasan yang harus diperhitungkan dalam proses penjadwalan, seperti konflik jadwal, keterbatasan ruang, serta beban kerja dosen.

Pada tahap desain Algoritma Genetika, dilakukan perancangan mekanisme algoritma yang mencakup representasi kromosom, fungsi fitness, serta operator seleksi, *crossover*, dan mutasi yang digunakan. Setelah desain selesai, algoritma diterapkan dalam tahap implementasi desain, di mana sistem mulai dijalankan untuk menghasilkan jadwal perkuliahan berdasarkan parameter yang telah ditentukan. Tahap terakhir adalah evaluasi hasil, yang bertujuan untuk mengukur kinerja algoritma dalam menghasilkan jadwal yang optimal serta membandingkannya dengan metode penjadwalan manual.

3.1 Pengumpulan Data.

Tahap pengumpulan data dilakukan untuk mendapatkan informasi yang diperlukan dalam proses penjadwalan perkuliahan. Data yang dikumpulkan mencakup informasi mengenai ruang perkuliahan, daftar dosen, jadwal waktu perkuliahan, serta jumlah kelas dan jumlah mahasiswa per kelas. Terdapat 20 ruang perkuliahan yang digunakan dalam proses penjadwalan. Setiap ruang memiliki kapasitas yang berbeda serta terletak di lantai yang berbeda pula. Rincian mengenai kapasitas dan lokasi masing-masing ruang dapat dilihat pada Tabel 1.

Tabel 1 Daftar Ruang Kuliah

No	Nama Ruang	Kapasitas	Lokasi
1	Ruang A2.1	45	Lt.2
2	Ruang A2.2	45	Lt.2
3	Ruang A2.3	45	Lt.2
4	Ruang A2.5	50	Lt.2
5	Ruang A3.1	65	Lt.3
6	Ruang A3.2	65	Lt.3

No	Nama Ruang	Kapasitas	Lokasi
7	Ruang A3.3	65	Lt.3
8	Ruang A3.4	45	Lt.3
9	Ruang A3.5	45	Lt.3
10	Ruang A3.6	45	Lt.3
11	Ruang A3.7	45	Lt.3
12	Ruang A3.8	45	Lt.3
13	Ruang A3.9	45	Lt.3
14	Ruang A3.10	45	Lt.3
15	Ruang A3.11	45	Lt.3
16	Ruang A3.12	45	Lt.3
17	Ruang A3.13	45	Lt.3
18	Lab. Komputer	16	Lt.2
19	Lab Bahasa	25	Lt.3
20	Aula	150	Lt.2

Selain itu, jumlah dosen pengampu mata kuliah dalam sistem ini sebanyak 50 orang. Setiap dosen memiliki jadwal mengajar yang berbeda-beda, bergantung pada jumlah mata kuliah yang diampu. Waktu perkuliahan dalam sistem ini dibagi menjadi 6 sesi per hari, dimulai dari pukul 07.00 hingga 23.00. Pembagian jadwal waktu ini dilakukan agar distribusi perkuliahan lebih merata dan dapat mengoptimalkan penggunaan ruang serta tenaga pengajar. Detail pembagian waktu perkuliahan dapat dilihat pada Tabel 2.

Tabel 2 Jadwal Kuliah

Jam Ke-	Waktu
1	07.00 s/d 09.30
2	09.30 s/d 12.00
3	13.00 s/d 15.30
4	15.30 s/d 18.00
5	18.00 s/d 20.30
6	20.30 s/d 23.00

Secara keseluruhan, dalam satu semester terdapat rata-rata 120 jadwal kuliah yang harus dijadwalkan. Setiap kelas memiliki jumlah peserta yang bervariasi, dengan rata-rata antara 30 hingga 45 mahasiswa per kelas. Data ini menjadi dasar dalam proses penjadwalan untuk memastikan distribusi yang optimal antara dosen, ruang kuliah, dan waktu perkuliahan sehingga tidak terjadi bentrokan jadwal maupun ketidakseimbangan beban kerja.

3.2 Analisis Masalah

Proses penyusunan jadwal perkuliahan yang dilakukan oleh bagian akademik memiliki beberapa kendala utama yang dihadapi. Kendala-kendala ini perlu dianalisis secara mendalam agar sistem penjadwalan yang dikembangkan mampu memberikan solusi yang optimal dan sesuai dengan kebutuhan. Salah satu kendala utama adalah keterbatasan ruang kelas. Jumlah ruang perkuliahan yang tersedia sebanyak 20 ruangan, harus dilakukan distribusi jadwal yang efisien agar tidak terjadi tumpang tindih penggunaan ruangan. Kapasitas setiap ruangan yang berbeda-beda juga menjadi tantangan tersendiri dalam menyesuaikan jumlah mahasiswa per kelas dengan kapasitas maksimal ruangan yang tersedia. Jika distribusi tidak dilakukan dengan baik, dapat terjadi ketimpangan dalam penggunaan ruangan, di mana beberapa ruang menjadi terlalu padat sementara yang lain kurang dimanfaatkan secara optimal.

Selain itu, kendala lainnya adalah preferensi dosen dalam mengajar. Beberapa dosen memiliki batasan atau preferensi tertentu dalam penjadwalan, seperti tidak ingin mengajar pada jam-jam tertentu atau tidak ingin mengajar di lantai tertentu. Hal ini dapat disebabkan oleh berbagai faktor, seperti kondisi kesehatan, efektivitas pengajaran pada waktu tertentu, atau aksesibilitas lokasi ruang kuliah. Preferensi ini harus dipertimbangkan dalam sistem penjadwalan agar tidak mengganggu kenyamanan dan produktivitas tenaga pengajar. Kombinasi dari kedua kendala ini menjadikan proses penjadwalan semakin kompleks. Jika tidak dikelola dengan baik, dapat terjadi konflik penjadwalan, seperti bentroknya waktu mengajar dosen, ketidakseimbangan beban mengajar, serta ketidakcocokan antara kapasitas ruangan dan jumlah mahasiswa dalam satu kelas. Oleh karena itu, diperlukan pendekatan yang lebih sistematis dan komputasional dalam penyusunan jadwal, sehingga kendala-kendala ini dapat diminimalkan dan jadwal yang dihasilkan lebih optimal serta sesuai dengan kebutuhan akademik.

3.3 Desain Algoritma Genetika

Penyelesaian masalah penjadwalan perkuliahan dengan keterbatasan ruang kelas dan preferensi dosen, dapat menggunakan Algoritma Genetika (AG) sebagai metode optimasi. AG merupakan teknik pencarian berbasis heuristik yang meniru proses evolusi alami untuk menemukan solusi optimal. Dalam penelitian ini, desain AG akan disesuaikan dengan kebutuhan penjadwalan yang mencakup 20 ruang kelas, 50 dosen, 6 slot waktu per hari, dan rata-rata 120 jadwal per semester.

Bagian pertama dari AG adalah representasi kromosom, setiap solusi dalam AG direpresentasikan sebagai kromosom, yang terdiri dari sejumlah gen. Dalam konteks penjadwalan ini, setiap gen merepresentasikan satu jadwal kuliah yang terdiri dari atribut Kode Matakuliah, Kode Dosen, Hari dan Waktu, Kode Ruangan, Jumlah Mahasiswa. Kromosom yang terbentuk akan berjumlah 120, sesuai dengan rata-rata jumlah jadwal kuliah per-semester. Langkah selanjutnya adalah membangkitkan populasi awal secara acak, dengan memastikan bahwa ruangan yang digunakan cukup untuk menampung jumlah mahasiswa, tidak terjadi bentrokan antar jadwal, dan memperhatikan preferensi dosen terkait waktu dan lantai tempat mengajar.

Selanjutnya, fungsi *fitness* digunakan untuk mengevaluasi kualitas Solusi yang dihasilkan. Beberapa faktor yang akan diperhatikan dalam fungsi *fitness* adalah tidak adanya bentrokan, distribusi jadwal yang merata dimana dosen tidak memiliki jadwal yang terlalu padat atau terlalu jarang, preferensi dosen, dan pemanfaatan kapasitas ruangan yang optimal. Fungsi *fitness* yang digunakan dalam penelitian ini dapat dilihat pada rumus (1).

$$fitness = \frac{1}{1+(B1+B2+B3+B4)} \quad (1)$$

Keterangan:

B1: Jumlah bentrokan ruangan

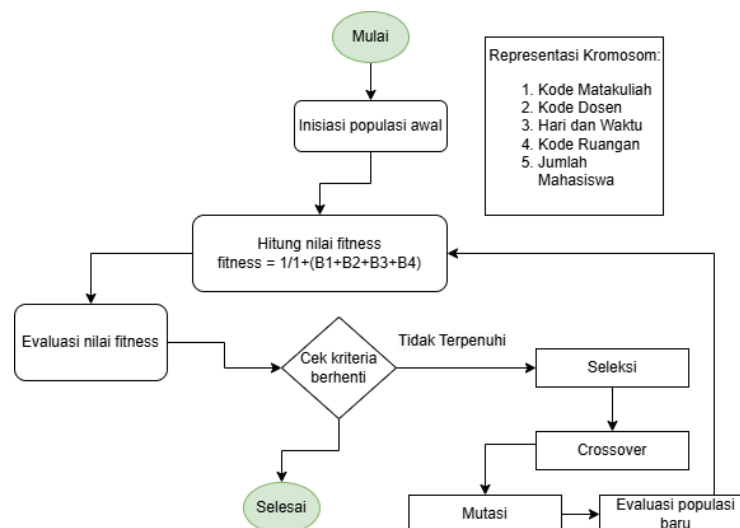
B2: Jumlah bentrokan jadwal dosen

B3: Jumlah pelanggaran preferensi dosen

B4: Jumlah kelas dengan pemanfaatan ruang yang optimal

Selanjutnya, operator AG yang digunakan dalam penelitian ini adalah seleksi, *crossover*, mutasi, dan Elitisme. Seleksi menggunakan metode *Roulette Wheel Selection* untuk memilih individu terbaik dengan probabilitas berdasarkan nilai *fitness*, sedangkan *crossover* menggunakan metode *One-Point Crossover* dengan probabilitas *crossover* 0.8, di mana dua jadwal akan saling bertukar sebagian gen. Selanjutnya operator mutasi dilakukan dengan mengganti slot waktu atau ruangan secara acak dengan probabilitas mutasi 0.1 untuk mencegah konvergensi premature, dan terakhir Elitisme digunakan jika ada beberapa individu terbaik dari generasi sebelumnya akan langsung masuk ke generasi berikutnya tanpa perubahan untuk menjaga kualitas solusi.

Proses evolusi untuk menemukan solusi akan berakhir jika sudah tidak ada bentrokan jadwal, atau sudah ada 1000 generasi yang terbentuk, atau tidak ada perubahan signifikan dalam 10 iterasi secara berturut-turut. Secara keseluruhan desain algoritma genetika yang dibuat dapat dilihat pada Gambar 2.



Gambar 2 Desain Algoritma Genetika

3.4 Implementasi Desain Algoritma

Pada tahap ini, desain algoritma akan di implementasikan pada bahasa pemrograman PHP. Berdasarkan desain yang sudah dijelaskan pada sub-bab 3.3, maka langkah awal implementasinya adalah dengan membuat kelas dengan empat atribut utama, yaitu `$room` mewakili ruangan, `$lecturer` mewakili dosen, `$timeslots` mewakili waktu, dan `$course` mewakili matakuliah. Implementasi kode pemrogramannya dapat dilihat pada Gambar 3.

```
public function __construct($rooms, $lecturers, $timeSlots, $courses) {
    $this->rooms = $rooms;
    $this->lecturers = $lecturers;
    $this->timeSlots = $timeSlots;
    $this->courses = $courses;
}
```

Gambar 3 Implementasi Kromosom

Selanjutnya adalah implementasi proses inisiasi populasi awal. Inisiasi populasi awal menggunakan parameter `$size` sebagai jumlah populasi yang dibentuk. `$size` akan mewakili berapa jumlah jadwal yang akan dibentuk dan dianggap sebagai populasi. Implementasi fungsi inisiasi populasi dapat dilihat pada Gambar 4.

```
private function initializePopulation($size) {
    for ($i = 0; $i < $size; $i++) {
        $schedule = [];
        foreach ($this->courses as $course) {
            $schedule[] = [
                'course' => $course,
                'room' => $this->rooms[array_rand($this->rooms)],
                'lecturer' => $this->lecturers[array_rand($this->lecturers)],
                'time' => $this->timeSlots[array_rand($this->timeSlots)]
            ];
        }
        $this->population[] = $schedule;
    }
}
```

Gambar 4 Implementasi fungsi inisiasi populasi awal

Langkah selanjutnya adalah mengimplementasikan fungsi menghitung nilai *fitness*. Fungsi ini bertugas mencari jumlah konflik/bentrok jadwal dari populasi yang sudah dibentuk sebelumnya. `$schedule` mewakili populasi/jadwal yang ada, setelah jumlah konflik ditemukan, kemudian dihitung nilai *fitness* yang dari populasi tersebut. Kode pemrograman fungsi *fitness* dapat dilihat pada Gambar 5.

```
private function calculateFitness($schedule) {
    $conflicts = 0;
    $timeCheck = [];
    foreach ($schedule as $class) {
        $key = $class['room'] . '-' . $class['time'];
        if (isset($timeCheck[$key])) {
            $conflicts++;
        }
        $timeCheck[$key] = true;
    }
    return 1 / (1 + $conflicts);
}
```

Gambar 5 Implementasi fungsi fitness

Langkah terakhir adalah mengimplementasikan fungsi seleksi, *crossover*, dan mutasi. Proses seleksi, *crossover*, dan mutasi dijalankan selama kondisi pemberhentian belum terpenuhi. Jika kondisi pemberhentian sudah terpenuhi, maka populasi terakhir itu yang akan menjadi jadwal perkuliahan. Implementasi kode pemrograman proses ini dapat dilihat pada Gambar 6.

```
private function selectParents() {
    usort($this->population, function ($a, $b) {
        return $this->calculateFitness($b) <=> $this->calculateFitness($a);
    });
    return array_slice($this->population, 0, 2);
}

private function crossover($parent1, $parent2) {
    $child = [];
    for ($i = 0; $i < count($parent1); $i++) {
        $child[] = ($i % 2 == 0) ? $parent1[$i] : $parent2[$i];
    }
    return $child;
}

private function mutate(&$schedule) {
    if (rand(0, 100) / 100 < $this->mutationRate) {
        $index = array_rand($schedule);
        $schedule[$index]['room'] = $this->rooms[array_rand($this->rooms)];
    }
}
```

Gambar 6 Implementasi proses seleksi, crossover, dan mutasi

3.5 Evaluasi Hasil

Setelah implementasi Algoritma Genetika (AG) untuk proses penjadwalan perkuliahan, hasil yang diperoleh menunjukkan peningkatan signifikan dibandingkan dengan metode manual. Salah satu indikator keberhasilan adalah waktu pemrosesan yang jauh lebih cepat. Berdasarkan estimasi dari kode yang telah diimplementasikan, proses pembentukan jadwal dengan AG memerlukan waktu sekitar 1 hingga 5 menit, tergantung pada kompleksitas jadwal dan jumlah iterasi yang dibutuhkan hingga solusi optimal tercapai. Waktu ini jauh lebih efisien dibandingkan dengan penyusunan jadwal secara manual, yang umumnya membutuhkan beberapa jam hingga beberapa hari untuk menghasilkan jadwal yang optimal. Hasil ini menunjukkan kesamaan dengan hasil penelitian yang dilakukan oleh Sur Liyan et al [12] yang menyatakan bahwa penerapan AG mempercepat penyusunan jadwal yang ada.

Selain efisiensi waktu, sistem penjadwalan yang dikembangkan juga berhasil menghilangkan seluruh konflik jadwal. Dari hasil evaluasi, jumlah jadwal yang mengalami bentrokan, baik karena tabrakan waktu perkuliahan, penggunaan ruang yang berlebihan, maupun ketidaksesuaian preferensi dosen, berhasil dikurangi hingga mencapai nol konflik. Hal ini menunjukkan bahwa pendekatan berbasis Algoritma Genetika tidak hanya mempercepat proses penjadwalan, tetapi juga menghasilkan jadwal yang lebih merata dan optimal, dengan distribusi mata kuliah yang lebih seimbang bagi dosen dan pemanfaatan ruang yang lebih efisien. Dengan hasil ini, dapat disimpulkan bahwa penggunaan Algoritma Genetika dalam sistem penjadwalan perkuliahan memberikan solusi yang lebih baik dibandingkan dengan metode manual. Tidak hanya dari segi kecepatan, tetapi juga dari segi kualitas jadwal yang dihasilkan, yang bebas dari konflik serta lebih sesuai dengan kebutuhan akademik di perguruan tinggi.

4. KESIMPULAN

Penelitian ini bertujuan untuk mengoptimalkan proses penjadwalan perkuliahan menggunakan Algoritma Genetika (AG) dengan mempertimbangkan keterbatasan ruang, preferensi dosen, serta distribusi jadwal yang lebih merata. Berdasarkan hasil implementasi dan evaluasi, sistem yang dikembangkan mampu menyusun jadwal secara lebih efisien dibandingkan dengan metode manual. Proses penjadwalan yang sebelumnya memerlukan waktu beberapa jam hingga beberapa hari kini dapat diselesaikan dalam waktu 1 hingga 5 menit, dengan hasil yang lebih optimal. Selain itu, sistem ini berhasil menghilangkan seluruh konflik jadwal, baik dari sisi tabrakan waktu, penggunaan ruang, maupun preferensi dosen, sehingga menghasilkan jadwal yang lebih terstruktur dan adil.

Meskipun sistem ini telah menunjukkan hasil yang signifikan, masih terdapat peluang untuk pengembangan lebih lanjut. Salah satu saran untuk penelitian selanjutnya adalah mengintegrasikan teknik optimasi lain, seperti Particle Swarm Optimization (PSO) atau Simulated Annealing, guna meningkatkan efisiensi pencarian solusi optimal. Selain itu, pengembangan sistem berbasis web atau cloud computing dapat meningkatkan aksesibilitas dan kemudahan dalam pengelolaan jadwal. Dengan pengembangan lebih lanjut, diharapkan sistem ini dapat diterapkan secara lebih luas dan fleksibel untuk mendukung manajemen akademik yang lebih baik di berbagai institusi pendidikan.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada UPTTI Institut Teknologi dan Sains Mandala yang telah menyediakan data referensi yang diperlukan dalam penelitian ini. Ucapan terima kasih juga disampaikan kepada Kepala Laboratorium Institut Teknologi dan Sains Mandala atas dukungan fasilitas laboratorium yang diberikan, sehingga penelitian ini dapat berjalan dengan baik. Selain itu, penulis juga berterima kasih kepada Bagian Akademik Institut Teknologi dan Sains Mandala yang telah menjadi mitra utama dalam diskusi dan komunikasi terkait sistem penjadwalan perkuliahan. Dukungan dan kontribusi dari berbagai pihak ini sangat membantu dalam penyelesaian penelitian ini.

REFERENSI

- [1] C. Tudose, "Optimizing the Scheduling of Teaching Activities in a Faculty," *Appl. Sci.*, vol. 14, no. 20, p. 9554, Oct. 2022. [Online]. Available: <https://www.mdpi.com/2076-3417/14/20/9554>.
- [2] A. Lestari, E. Susilo, dan A. Anhar, "Sistem Penjadwalan Perkuliahan Menggunakan Algoritma Genetika Pada Jurusan Teknik Elektro Universitas Riau," *Jurnal Teknologi Informatika dan Komputer*, vol. 9, no. 2, 2023.
- [3] S. Hikmawan, "Algoritma Genetika dengan Mutasi Terbatas untuk Penjadwalan Perkuliahan," *Jurnal Kajian Ilmiah*, vol. 21, no. 2, 2021.
- [4] R. E. Wiratna, A. L. Nurlaili, dan A. M. Rizki, "Pembuatan Aplikasi Penjadwalan Mata Kuliah Menggunakan Algoritma Genetika," *Jurnal Teknologi dan Manajemen*, vol. 4, no. 1, 2023.
- [5] A. Amrulloh, and E. I. Sela, "Optimasi proses penjadwalan mata kuliah menggunakan algoritme genetika dan pencarian tabu," *Jurnal Teknologi dan Sistem Komputer*, vol. 9, no. 3, pp. 157-166, Jun. 2021. <https://doi.org/10.14710/jtsiskom.2021.14137>
- [6] H. D. P. Sinaga, P. Gultom, S. Syahriol, dan S. Suyanto, "Implementasi Algoritma Genetika untuk Penjadwalan Perkuliahan (Studi Kasus: Program Studi Sarjana Matematika di FMIPA Universitas Sumatera Utara)," *Innovative: Journal Of Social Science Research*, vol. 4, no. 4, 2024.
- [7] B. Budi, R. Sari, dan D. Kusuma, "Kombinasi Algoritma Genetika dan Simulated Annealing untuk Optimasi Penjadwalan Perkuliahan," *Jurnal Sistem Informasi*, vol. 11, no. 3, 2022.
- [8] A. Aulia, F. Ramadhan, dan R. Widodo, "Optimasi Parameter Adaptif dalam Algoritma Genetika untuk Penjadwalan Perkuliahan Skala Besar," *Jurnal Teknologi Informasi dan Komputer*, vol. 5, no. 1, 2021.
- [9] S. Siregar, M. Wibowo, dan T. Gunawan, "Hybrid Genetic Algorithm and Particle Swarm Optimization for University Timetabling," *International Journal of Computer Science and Information Security*, vol. 18, no. 4, 2022.
- [10] A. Handoko dan R. Prasetyo, "Implementasi Algoritma Genetika untuk Penjadwalan Perkuliahan Berbasis Preferensi Dosen," *Jurnal Informatika*, vol. 9, no. 2, 2023.
- [11] Y. Yuliana, H. Purnama, dan L. Fadillah, "Cloud-Based Genetic Algorithm for Academic Scheduling Optimization," *Journal of Advanced Computing Research*, vol. 7, no. 2, 2024.
- [12] Sur Liyan, et al, "Lecture Scheduling Using Genetic Algorithm", *Journal of Intelligent Software Systems*, Vol.3, No.2, 2024, pp.7-12, 10.26798/jiss.v3i2.1501