



Perencanaan Pengembangan Sistem Informasi Penjadwalan Kuliah Praktikum di STMIK AMIKOM Yogyakarta

Hendra Kurniawan ^{1*}, Tristanto Ari Aji ²

¹ Program Studi Sistem Informasi, Universitas Amikom Yogyakarta, Indonesia

² Program Studi Informatika, Universitas Amikom Yogyakarta, Indonesia

* Korespondensi: hendrakurniawan@amikom.ac.id

Sitasi: Kurniawan, H.; Aji, T. A.; (2025). Perencanaan Pengembangan Sistem Informasi Penjadwalan Kuliah Praktikum di STMIK AMIKOM Yogyakarta JTIM: Jurnal Teknologi Informasi Dan Multimedia, 7(4), 710-720. <https://doi.org/10.35746/jtim.v7i4.778>

Diterima: 16-06-2025

Direvisi: 27-07-2025

Disetujui: 08-08-2025



Copyright: © 2025 oleh para penulis. Karya ini dilisensikan di bawah Creative Commons Attribution-ShareAlike 4.0 International License. (<https://creativecommons.org/licenses/by-sa/4.0/>).

Abstract: Practicum course scheduling is a complex task in higher education institutions as it involves multiple parameters such as lecturer availability, room capacity, and time slots. This process poses a significant challenge for laboratory administrators in ensuring that scheduling conflicts are avoided and that all resources are utilized optimally. This study implements a Genetic Algorithm (GA) to optimize the practicum course scheduling process at STMIK AMIKOM Yogyakarta, which has since been renamed Universitas Amikom Yogyakarta. The methodological stages include population initialization, fitness evaluation, selection using the Roulette Wheel Selection method, crossover using One Point Crossover, and mutation using Targeted Mutation. The results demonstrate that the genetic algorithm successfully produces optimal solutions by eliminating lecturer and room conflicts, while also maximizing equitable time utilization. During the iteration phase, the algorithm generated a conflict-free practicum schedule with a maximum fitness value of 167. The process terminated at the first generation after identifying two optimal chromosomes out of ten. These findings confirm that the genetic algorithm is effective in solving practicum scheduling problems and can be applied to minimize schedule clashes and improve operational efficiency in academic environments.

Keywords: algorithm; genetic; scheduling; practical; fitness value

Abstrak: Penjadwalan kuliah praktikum merupakan aktivitas yang kompleks di perguruan tinggi karena melibatkan banyak parameter seperti ketersediaan dosen, ruang, dan waktu. Proses ini menjadi tantangan bagi pihak laboratorium untuk memastikan bahwa tidak ada bentrokan jadwal dan semua sumber daya dapat dimanfaatkan secara optimal. Penelitian ini mengimplementasikan Genetic Algorithm untuk mengoptimalkan proses penjadwalan kuliah praktikum di STMIK AMIKOM Yogyakarta yang sekarang berubah nama menjadi Universitas Amikom Yogyakarta. Tahapannya mencakup inisialisasi populasi, evaluasi fitness, seleksi dengan metode Roulette Wheel Selection, crossover menggunakan One Point Crossover, dan mutasi menggunakan Targeted Mutation. Hasil penelitian menunjukkan bahwa Genetic Algorithm mampu menghasilkan solusi optimal dengan menghilangkan konflik dosen dan ruang, serta memaksimalkan pemanfaatan waktu yang merata. Pada tahap iterasi menghasilkan jadwal kuliah praktikum tanpa konflik dengan nilai fitness maksimal sebesar 167. Prosesnya berhenti pada generasi pertama setelah ditemukan 2 kromosom optimal dari 10 kromosom. Penelitian ini membuktikan bahwa Genetic Algorithm efektif dalam mengatasi masalah penjadwalan kuliah praktikum, sehingga dapat diterapkan untuk meminimalkan bentrokan jadwal dan meningkatkan efisiensi dalam operasional akademik.

Kata kunci: algoritma; genetika, penjadwalan; praktikum, nilai fitness

1. Pendahuluan

Berdasarkan Peraturan Pemerintah Republik Indonesia Nomor 60 Tahun 1999 tentang Perguruan Tinggi Pasal 56 Ayat 1 menjelaskan bahwa setiap universitas/institusi harus memiliki unsur penunjang untuk menyelenggarakan perguruan tinggi yang meliputi perpustakaan, pusat komputer, dan laboratorium/studio[1]. Laboratorium/studio berfungsi untuk mendukung kegiatan Tri Dharma Perguruan Tinggi, yang meliputi pendidikan dan pengajaran, penelitian, serta pengabdian masyarakat. Penjadwalan terhadap laboratorium/studio menjadi hal penting, yang ditujukan untuk mengatur penggunaan laboratorium/studio secara baik. Proses penjadwalan menjadi permasalahan yang cukup kompleks dan sulit diselesaikan menggunakan cara konvensional, dikarenakan membutuhkan waktu yang lama untuk mendapatkan hasil yang optimal[2].

STMIK AMIKOM Yogyakarta yang sekarang bernama Universitas Amikom Yogyakarta melakukan penjadwalan laboratorium/studio untuk kuliah praktikum pada setiap awal semester. Proses penjadwalan dilakukan oleh bagian akademik menggunakan sistem informasi penjadwalan, sehingga mampu menghasilkan jadwal kuliah praktikum secara cepat. Akan tetapi, hasil penjadwalannya sering dijumpai adanya masalah berupa bentrokan dosen dan ruang. Seorang dosen pada satu waktu dapat terjadwal lebih dari 1 kelas yang berbeda. Satu ruang yang berupa laboratorium/studio juga dapat terjadwal untuk lebih dari 1 kelas pada waktu yang sama. Dampaknya, salah satu kelas terpaksa harus dibatalkan untuk melaksanakan kuliah praktikum dan bagian akademik harus melakukan penjadwalan ulang.

Penjadwalan kuliah praktikum di Universitas Amikom Yogyakarta perlu mempertimbangkan banyak aturan internal, seperti plotting dosen, fungsional dan kapasitas laboratorium/studio, serta waktu kuliah. Apabila aturan ini tidak diperhatikan dengan baik, maka akan menimbulkan masalah berupa bentrokan jadwal. Permasalahan ini dapat diminimalisir dengan melakukan optimalisasi menggunakan algoritma tertentu terhadap sistem informasi penjadwalan. Beberapa algoritma yang dapat digunakan antara lain Genetic Algorithm (GA), Ant Colony Optimization (ACO), Tabu Search (TS), Graph Coloring (GC), Particle Swarm Optimization (PSO), Simulated Annealing (SA), Greedy Algorithm (AG), dan lain-lain[3][4][5][6][7].

Genetic Algorithm (GA) merupakan salah satu jenis algoritma optimasi yang handal dan banyak digunakan dalam proses penjadwalan[8][9]. Algoritma ini digunakan untuk mendapatkan solusi optimal melalui teknik penentuan populasi awal dan inisialisasi kromosom, penentuan nilai fitness, seleksi crossover, serta mutasi[10]. Mutasi akan dilakukan sampai menghasilkan nilai fitness terbaik yang dapat dimanfaatkan untuk menentukan hasil akhir pembuatan jadwal. Apabila nilai fitness terbaik telah diperoleh, maka proses berhenti dan selesai.

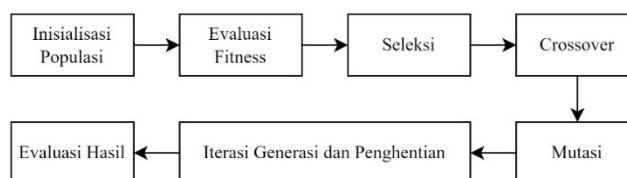
Beberapa penelitian sebelumnya menunjukkan efektivitas Genetic Algorithm dalam menyelesaikan masalah penjadwalan. Sanapiah dkk menggunakan Genetic Algorithm untuk menyusun jadwal asisten praktikum dengan mempertimbangkan keterbatasan laboratorium, jadwal mahasiswa, serta aturan bahwa asisten dapat mengajar beberapa kelas dengan batas maksimal, dan hasilnya menunjukkan solusi tanpa bentrokan dengan nilai fitness 1[11]. Handayani dkk membandingkan Greedy Algorithm, Simulated Annealing, dan Genetic Algorithm dalam penjadwalan mata kuliah, dan menemukan bahwa Genetic Algorithm memberikan solusi optimal dan tercepat dengan nilai fitness 1[12]. Sementara itu, Mardia dan Sunarto berhasil mengembangkan sistem informasi penjadwalan berbasis web pada program studi Tadris Matematika menggunakan Genetic Algorithm yang menghasilkan jadwal tanpa konflik antara ruang, waktu, dan dosen[13]. Penelitian oleh Ardiansyah dan Junianto juga menunjukkan keberhasilan penerapan Genetic Algorithm dalam penjadwalan mata pelajaran di sekolah dasar dengan parameter seperti populasi dan tingkat mutasi yang mempercepat proses penyusunan jadwal[14]. Adapun

Shofiyati dkk mengimplementasikan Genetic Algorithm untuk mengatasi bentrokan jadwal perkuliahan di Politeknik Unisma Malang dan berhasil menghilangkan konflik antara ruangan, kelas, dan dosen[15].

Berdasarkan kelima penelitian sebelumnya di atas, maka dapat disimpulkan bahwa Genetic Algorithm dapat diimplementasikan untuk melakukan penjadwalan. Algoritma tersebut mampu mencari kemungkinan-kemungkinan terbaik untuk mendapatkan hasil yang optimal tanpa adanya bentrokan jadwal. Penelitian ini bertujuan untuk mengimplementasikan Genetic Algorithm untuk penjadwalan kuliah praktikum di Universitas Amikom Yogyakarta. Sementara itu, kontribusi utamanya sebagai dasar perencanaan pengembangan sistem informasi penjadwalan kuliah praktikum menggunakan Genetic Algorithm supaya menghasilkan jadwal kuliah praktikum yang optimal, tanpa konflik dosen dan ruang, serta efisien dalam pemanfaatan waktu.

2. Bahan dan Metode

Metode penelitian menggambarkan tahapan dalam mengimplementasikan algoritma genetika untuk penjadwalan kuliah praktikum di Universitas Amikom Yogyakarta. Tahapan tersebut dirancang untuk meminimalkan terjadinya bentrokan jadwal dengan mempertimbangkan aturan dan parameter yang relevan dan hasilnya ditunjukkan pada gambar 1.



Gambar 1. Tahapan penelitian

Penelitian terdiri dari 7 tahapan, yaitu inisialisasi populasi, evaluasi fitness, seleksi, crossover, mutasi, iterasi genetika dan penghentian, dan evaluasi hasil.

2.1 Inisialisasi Populasi

Pada tahap ini akan menghasilkan sekumpulan solusi awal yang disebut populasi yang berisi sejumlah kromosom. Setiap kromosom merepresentasikan satu solusi potensial untuk masalah penjadwalan kuliah praktikum. Data yang digunakan dalam tahapan inisialisasi populasi berasal dari Fakultas Ilmu Komputer yang meliputi 3 prodi (Program Studi), yaitu Sistem Informasi, Informatika, dan Manajemen Informatika.

2.2 Evaluasi Fitness

Setiap kromosom dievaluasi menggunakan fungsi fitness untuk menilai kualitas solusi penjadwalan kuliah praktikum berdasarkan kriteria tertentu. Proses evaluasi nilai fitness akan berlangsung terus-menerus hingga kriteria penghentian terpenuhi. Hasilnya berupa nilai yang menunjukkan kualitas dari setiap solusi, di mana solusi dengan fitness yang tinggi dianggap lebih baik dibandingkan dengan fitness yang rendah.

$$Fitness = N - (C_d + C_r) \quad (1)$$

N merupakan jumlah total gen dalam kromosom yang merepresentasikan solusi penjadwalan kuliah praktikum tanpa konflik. C_d menunjukkan jumlah konflik dosen, di mana seorang dosen dijadwalkan mengajar lebih dari satu kelas pada waktu yang sama. C_r menunjukkan jumlah konflik ruang, yaitu pada saat beberapa kelas menggunakan ruang yang sama secara bersamaan.

2.3 Seleksi

Seleksi adalah proses memilih kromosom terbaik berdasarkan nilai fitness untuk dijadikan induk generasi berikutnya, di mana kromosom dengan fitness lebih tinggi memiliki peluang lebih besar untuk dipilih, karena dianggap mewakili solusi yang lebih baik. Metode seleksi yang digunakan adalah *Roulette Wheel Selection*, yang memberikan probabilitas seleksi pada kromosom sesuai proporsi fitness-nya. Perhitungan dan tahapan dalam metode *Roulette Wheel Selection*:

- a. Menghitung probabilitas seleksi

Rumusnya sebagai berikut:

$$P(i) = \frac{f(i)}{\sum_{j=1}^N f(j)} \quad (2)$$

$P(i)$ adalah probabilitas individu i terpilih, $f(i)$ adalah nilai fitness individu i , N adalah jumlah individu, dan $f(j)$ adalah total fitness populasi.

- b. Menghitung probabilitas kumulatif

Rumusnya sebagai berikut:

$$C(i) = \sum_{k=1}^i P(k) \quad (3)$$

$C(i)$ adalah probabilitas kumulatif untuk individu i dan $P(k)$ adalah probabilitas seleksi individu k .

- c. Pemilihan individu

Rumusnya sebagai berikut:

$$C(i-1) < r \leq C(i) \quad (4)$$

$C(i-1)$ adalah probabilitas kumulatif sebelumnya, $C(i)$ adalah probabilitas kumulatif individu saat ini, dan r adalah angka acak dari interval $[0, 1]$.

2.4 Crossover (Penyilangan)

Crossover merupakan proses penggabungan dua kromosom induk terbaik untuk membentuk kromosom-kromosom anak yang baru. Tujuannya untuk menciptakan kombinasi gen yang lebih baik dan mendekati solusi optimal. *Crossover* dilakukan dengan menukar sebagian gen antara dua induk, kemudian menghasilkan variasi baru lebih banyak. Metode *crossover* yang dapat digunakan antara lain *One Point Crossover* dan perhitungannya sebagai berikut:

- a. Menentukan titik potong

Rumusnya sebagai berikut:

$$t = \text{random}(1, \min(\text{length}(\text{parent1}), \text{length}(\text{parent2})) - 1) \quad (5)$$

T adalah posisi indeks titik potong, $\text{length}(\text{parent1})$ dan $\text{length}(\text{parent2})$ adalah panjang kromosom parent.

- b. Membentuk anak (*offspring*)

Rumusnya sebagai berikut:

$$\begin{aligned} \text{child1} &= \text{parent1}[1:t] + \text{parent2}[t:] \\ \text{child2} &= \text{parent2}[1:t] + \text{parent1}[t:] \end{aligned} \quad (6)$$

$parent1[:t]$ adalah gen awal dari $parent1$ hingga titik potong, $parent2[t:]$ adalah gen akhir dari $parent2$ setelah titik potong, $parent2[:t]$ adalah gen awal dari $parent2$ hingga titik potong, dan $parent1[t:]$ adalah gen akhir dari $parent1$ setelah titik potong.

2.5 Mutasi

Mutasi merupakan proses perubahan acak pada beberapa gen dalam kromosom yang bertujuan menjaga keragaman genetik. Mutasi dapat dilakukan dengan memilih gen yang menyebabkan pelanggaran atau konflik dalam kromosom, seperti jadwal yang konflik. Metode mutasi yang digunakan adalah targeted mutation dan perhitungannya sebagai berikut:

a. Mengidentifikasi gen bermasalah

Gen bermasalah adalah gen yang menyebabkan konflik dosen atau ruang. Cara mengidentifikasinya dengan memeriksa apakah ada 2 gen ($g1, g2$) yang berupa hari dan jam yang sama, dosen atau ruang yang sama.

$$Konflik(g1, g2) = \begin{cases} 1, & \text{jika } g1.hari = g2.hari \wedge g1.jam = g2.jam \wedge \\ & (g1.dosen = g2.dosen \vee g1.ruang = g2.ruang) \\ 0, & \text{lainnya} \end{cases} \quad (7)$$

b. Memilih hari dan jam baru

Apabila ditemukan konflik pada suatu gen (g), maka akan dipilih hari dan jam baru secara acak dari daftar yang tersedia.

$$g.hari_baru, g.jam_baru = random.choice(hari), random.choice(jam) \quad (8)$$

c. Memvalidasi hari dan jam baru

Hari dan jam baru dapat diterima, apabila tidak ada gen lain (g_k) dalam kromosom yang memiliki hari dan jam yang sama dengan dosen atau ruang yang sama.

$$\begin{aligned} Validasi(g, g_{baru}) &= \text{tidak ada } g_k \text{ samadengan: } g_k.hari = g_{baru}.hari \wedge g_k.jam \\ &= g_{baru}.jam \wedge (g_k.dosen = g_{baru}.dosen \vee g_k.ruang \\ &= g_{baru}.ruang). \end{aligned} \quad (9)$$

d. Mutasi pada populasi

Setiap gen dalam kromosom memiliki peluang atau probabilitas mutasi (P_m). Mutasi hanya dilakukan pada gen yang menyebabkan konflik.

$$P_m = random(0,1), \text{ jika } P_m < P_{mutasi}, \text{ gendimutasi.} \quad (10)$$

Mutasi dapat mencegah modifikasi yang berlebihan pada jadwal yang sudah baik.

2.6 Iterasi Generasi dan Penghentian

Tahap Iterasi Generasi dan Penghentian merupakan proses berulang di mana populasi baru yang dihasilkan dari seleksi, *crossover*, dan mutasi dievaluasi dan diseleksi lagi dalam siklus yang berkelanjutan. Seluruh kromosom pada setiap generasi akan dievaluasi menggunakan fungsi fitness untuk menilai kualitas solusi yang dihasilkan. Kromosom-kromosom terbaik dipilih untuk diturunkan ke generasi berikutnya, sementara hasil dari *crossover* dan mutasi untuk memperkaya populasi dengan solusi yang lebih bervariasi. Proses ini akan diulang hingga memenuhi kriteria penghentian yang telah ditetapkan.

2.7 Evaluasi Hasil

Evaluasi Hasil merupakan proses alisis terhadap solusi terbaik yang telah dihasilkan untuk memastikan kualitas dan kecocokannya dengan tujuan optimasi. Proses ini mencakup pengecekan apakah solusi tersebut memenuhi semua batasan dan kriteria yang ditetapkan, seperti alokasi waktu, sumber daya, dan preferensi tertentu dalam penjadwalan kuliah praktikum.

3. Hasil

Hasil dan pembahasan menyajikan hasil penelitian yang diperoleh dari implementasi algoritma genetika untuk penjadwalan kuliah praktikum di Universitas Amikom Yogyakarta, serta membahas secara mendalam mengenai temuannya. Bagian ini berisi hasil dan pembahasan dari setiap tahapan dalam metode penelitian.

3.1. Inisialisasi Populasi

Setiap kromosom dalam populasi merepresentasikan solusi jadwal yang terdiri dari alokasi dosen, mata kuliah, kelas, ruang, hari, dan jam. Struktur kromosom dirancang dalam format array, di mana setiap gen dalam kromosom mewakili satu jadwal kuliah praktikum.

prodi	dosen	nama	kelas	ruang	hari	jam
-------	-------	------	-------	-------	------	-----

Gambar 2. Desain gen

Setiap gen dalam kromosom menggambarkan satu sesi kuliah praktikum yang terdiri dari beberapa komponen utama, yaitu: prodi (program studi seperti Sistem Informasi, Informatika, dll.), dosen (pengampu praktikum), nama mata kuliah (misalnya Pengantar Teknologi Informasi, Struktur Data, dll.), ruang (laboratorium atau studio sebagai tempat praktikum), kelas (kelompok mahasiswa berdasarkan angkatan), serta hari dan jam (waktu pelaksanaan kuliah praktikum yang berlangsung dari Senin hingga Jumat dengan lima slot waktu mulai pukul 07.00 hingga 17.10).

Setiap kromosom dalam tahap inisialisasi populasi dihasilkan secara acak, namun pembangkitannya harus memperhatikan aturan dan batasan yang ditetapkan oleh pihak perguruan tinggi. Aturan dan batasan tersebut ditunjukkan pada tabel 1.

Tabel 1. Aturan dan batasan dalam penjadwalan

Aturan dan Batasan	Deskripsi
Dosen, Mata Kuliah	Setiap dosen harus dijadwalkan pada mata kuliah yang ditetapkan oleh prodi
Dosen dan Kelas	Setiap dosen harus dijadwalkan pada kelas yang ditetapkan oleh prodi
Mata Kuliah, Ruang, dan Waktu	Setiap mata kuliah dialokasikan pada ruang dan waktu oleh laboran untuk menyesuaikan kebutuhan software dan hardware-nya
Konflik Dosen	Seorang dosen tidak boleh mengajar lebih dari satu kelas pada waktu yang sama
Konflik Ruang	Satu ruangan tidak boleh digunakan untuk lebih dari satu kelas pada waktu yang sama

Proses pembuatan populasi diawali dengan iterasi sesuai dengan plotting dosen yang dilakukan oleh prodi. Setiap mata kuliah menghasilkan gen yang dibentuk berdasarkan jumlah kelas yang diampu oleh dosen. Pada penelitian ini menggunakan data dari Fakultas Ilmu Komputer yang berupa 82 item mata kuliah dengan alokasi dosen dan kelas yang telah ditentukan.

Hari dan jam dipilih secara acak, serta jumlah gen dalam satu kromosom ditentukan oleh total mata kuliah yang perlu dijadwalkan. Setiap gen mewakili satu elemen jadwal seperti ditunjukkan pada tabel 2.

Tabel 2. Hasil pembuatan gen

No	Prodi	Dosen	Nama Matkul	Kelas	Ruang	Ruang	Jam
1	Sistem Informasi	Hendra Kurniawan, M.Kom	Pengantar Teknologi Informasi	21-S1SI-01	Laboratorium 7.3.1	Senin	10.40-12.20
2	Sistem Informasi	Hendra Kurniawan, M.Kom	Pengantar Teknologi Informasi	21-S1SI-02	Laboratorium 7.3.1	Selasa	07.00-08.40
3	Sistem Informasi	Hendra Kurniawan, M.Kom	Pengantar Teknologi Informasi	21-S1SI-03	Laboratorium 7.3.1	Rabu	13.20-15.00
...	...	...	...	...	...	...	...
167	Manajemen Informatika	Yuli Astuti, M.Kom	Pascaproduksi Multimedia	20-D3MI-Pil1	Laboratorium 2.3.3	Rabu	10.40-12.20

Proses inisialisasi populasi menghasilkan 10 kromosom (solusi awal) dan total gen dalam populasi berjumlah 1670. Setiap kromosom terdiri dari 167 gen yang merepresentasikan 167 jadwal kuliah praktikum.

3.2. Evaluasi Fitness

Evaluasi fitness bertujuan untuk menilai kualitas kromosom yang dihasilkan dalam populasi berdasarkan sejumlah kriteria (aturan dan batasan) yang telah ditentukan. Nilai fitness dihitung dari jumlah pelanggaran yang terjadi. Hasilnya ditunjukkan pada tabel 3.

Tabel 3. Evaluasi nilai fitness

Kromosom	Pelanggaran		Nilai Fitness	Jumlah Gen
	1	2		
1	12	94	61	167
2	22	122	23	167
3	12	68	87	167
4	4	100	63	167
5	16	108	43	167
6	12	94	61	167
7	14	94	59	167
8	18	104	45	167
9	10	94	63	167
10	14	94	59	167
Total	134	972	564	1670

Nilai fitness yang tinggi menunjukkan bahwa jumlah pelanggaran yang terjadi relatif kecil. Misalnya, kromosom 3 memiliki nilai fitness tertinggi, yaitu 87, dengan pelanggaran berupa konflik dosen 12 dan konflik ruang 68. Sebaliknya, kromosom 2 memiliki nilai fitness terendah, yaitu 23, dengan pelanggaran konflik dosen 22 dan konflik ruang 122.

3.3. Seleksi

Seleksi menggunakan metode *Roulette Wheel Selection* untuk memilih individu (kromosom) dari populasi guna melanjutkan ke tahap berikutnya, seperti *crossover* dan *mutasi*. Pemilihan dilakukan berdasarkan tingkat fitness-nya, di mana individu dengan fitness tinggi memiliki peluang lebih besar untuk dipilih, sehingga populasi akan menghasilkan solusi yang lebih baik. Hasilnya ditunjukkan pada tabel 4.

Tabel 4. Hasil seleksi menggunakan metode *Roulette Wheel Selection*

Kromosom	Fitness Sebelum Seleksi	Fitness Setelah Seleksi
1	61	45
2	23	63
3	87	61

Kromosom	Fitness Sebelum Seleksi	Fitness Setelah Seleksi
4	63	59
5	43	87
6	61	87
7	59	59
8	45	63
9	63	61
10	59	61
Total	564	646

Beberapa kromosom mengalami peningkatan fitness, seperti kromosom 2 dan 5. Di sisi lain, beberapa kromosom mengalami penurunan fitness, seperti kromosom 1 dan 3. Secara umum metode *Roulette Wheel Selection* menghasilkan peningkatan total fitness populasi dari 564 menjadi 646. Kondisi ini menunjukkan adanya mekanisme meningkatkan keragaman genetik untuk mencari solusi yang optimal.

3.4. Crossover (Penyilangan)

Proses *crossover* menggunakan metode *One Point Crossover* untuk menghasilkan jadwal baru yang lebih baik dengan cara menggabungkan informasi dari dua jadwal (kromosom) yang sudah ada. Metode ini menggabungkan dua individu (*parent*) untuk menghasilkan keturunan (*offspring*). Hasilnya ditunjukkan pada tabel 5.

Tabel 5. Hasil *crossover* menggunakan metode *One Point Crossover*

Kromosom	Fitness Sebelum Crossover	Fitness Setelah Crossover
1	45	51
2	63	59
3	61	49
4	59	63
5	87	87
6	87	87
7	59	69
8	63	49
9	61	65
10	61	57
Total	646	636

Secara umum, metode *One Point Crossover* menghasilkan penurunan total fitness populasi. Sebelum penyilangan, total fitnessnya adalah 646, kemudian setelah penyilangan menurun menjadi 636. Hal ini berarti terjadi penurunan 10 poin atau sekitar 1.55%. Penurunan ini diakibatkan oleh karakteristik probabilistik *crossover* yang memungkinkan kromosom dengan fitness rendah tetap terpilih untuk menjaga keberagaman populasi dan mencegah konvergensi dini pada solusi suboptimal. Kromosom terbaik, yaitu kromosom 5 dan 6 dengan nilai fitness 87 berhasil dipertahankan, sehingga solusi optimal tetap ada dalam populasi. Beberapa kromosom mengalami peningkatan fitness, seperti kromosom 1, 7, dan 9. Akan tetapi, beberapa kromosom ada yang mengalami penurunan fitness, seperti kromosom 8.

3.5. Mutasi

Tahap mutasi dilakukan menggunakan *targeted mutation* untuk memperbaiki solusi yang mengandung konflik, baik konflik dosen atau ruang, serta meningkatkan kualitas jadwal kuliah praktikum. Hasilnya ditunjukkan pada tabel 6.

Tabel 6. Hasil mutasi menggunakan metode *targeted mutation*

Kromosom	Fitness Sebelum Mutasi	Fitness Setelah Mutasi
1	51	145
2	59	155
3	49	141
4	63	145
5	87	167
6	87	167
7	69	167
8	49	167
9	65	167
10	57	167
Total	636	1588

Total fitness populasi mengalami peningkatan signifikan dari 636 menjadi 1588. Hal ini menunjukkan bahwa proses mutasi menggunakan metode *targeted mutation* berhasil mengatasi konflik yang ada pada jadwal kuliah praktikum, sehingga meningkatkan kualitas solusi secara keseluruhan. Beberapa kromosom mencapai nilai fitness optimal sebesar 167, seperti pada kromosom 5, 6, 7, 8, 9, dan 10.

3.6. Iterasi Generasi dan Penghentian

Iterasi Generasi dan Penghentian bertujuan untuk memperbaiki kualitas solusi secara terus-menerus hingga menemukan jadwal kuliah praktikum yang optimal. Hasilnya ditunjukkan pada tabel 7.

Tabel 7. Hasil iterasi generasi dan penghentian

Kromosom	Konflik Gen	Nilai Fitness
1	9	157
2	6	161
3	3	163
4	26	141
5	6	161
6	15	151
7	20	147
8	8	159
9	0	167
10	0	167
Total	93	1574

Dari total 10 kromosom dalam populasi terdapat 2 kromosom (kromosom 9 dan 10) memiliki konflik gen 0. Hal ini menunjukkan bahwa kedua kromosom tersebut adalah solusi optimal tanpa adanya bentrokan dosen atau ruang. Nilai fitness maksimum yang dicapai adalah 167 sesuai dengan jumlah total gen yang harus dijadwalkan tanpa konflik. Dengan demikian, dapat disimpulkan bahwa tahap iterasi telah berhasil menemukan solusi optimal (fitness = 167), sehingga proses algoritma genetika dihentikan.

4. Pembahasan

Pada penelitian sebelumnya menerapkan Genetic Algorithm untuk menyusun jadwal pelajaran di SD Kreatif Muhammadiyah 2 Bontang[16]. Hasilnya memperoleh tingkat fitness rata-rata 0,5 hingga 1 tergantung skema pengujian, sehingga cocok digunakan dalam skala kecil. Sementara itu, penelitian yang dilakukan di Universitas Amikom Yogyakarta menerapkan Genetic Algorithm dalam skala yang lebih kompleks untuk penjadwalan kuliah praktikum dengan melibatkan banyak parameter seperti dosen, ruang, dan

waktu, serta menggunakan pendekatan terstruktur seperti metode *Roulette Wheel Selection*, *One Point Crossover*, dan *Targeted Mutation*. Hasilnya menunjukkan nilai fitness maksimal sebesar 167 tanpa konflik dan solusi optimal ditemukan hanya dalam satu generasi.

Hasil penerapan algoritma genetika dalam penjadwalan kuliah praktikum ditunjukkan seperti pada tabel 8.

Tabel 8. Hasil penjawalan kuliah praktikum

No	Prodi	Dosen	Nama Matkul	Kelas	Ruang	Hari	Jam
1	Sistem Informasi	Hendra Kurniawan, M.Kom	Pengantar Teknologi Informasi	21-S1SI-01	Laboratorium 7.3.1	Senin	10.40-12.20
2	Sistem Informasi	Hendra Kurniawan, M.Kom	Pengantar Teknologi Informasi	21-S1SI-02	Laboratorium 7.3.1	Senin	08.50-10.30
3	Sistem Informasi	Hendra Kurniawan, M.Kom	Pengantar Teknologi Informasi	21-S1SI-03	Laboratorium 7.3.1	Jumat	08.50-10.30
...	...	...	...	...	...	...	...
167	Manajemen Informatika	Yuli Astuti, M.Kom	Pascaproduksi Multimedia	20-D3MI-Pil1	Laboratorium 2.3.3	Jumat	07.00-08.40

Hasil penjadwalan telah melalui beberapa tahap pengecekan yang meliputi konflik dosen, konflik ruang, dan pemanfaatan waktu. Pada aspek konflik dosen menunjukkan bahwa setiap dosen hanya mengajar satu kelas pada waktu tertentu. Sebagai contoh, Hendra Kurniawan, M.Kom mengajar tiga kelas dengan waktu yang berbeda, yaitu Senin jam 10.40-12.20 dan 08.50-10.30, serta Jumat jam 08.50-10.30, sehingga tidak terjadi bentrokan jadwal. Dari segi konflik ruang, setiap laboratorium hanya digunakan untuk satu kelas pada waktu tertentu, seperti Laboratorium 7.3.1 yang dijadwalkan untuk beberapa kelas pada slot waktu yang berbeda. Pemanfaatan waktu juga merata yang tersebar mulai dari hari Senin s/d Jumat pada jam 07.00-08.40, 08.50-10.30, dan seterusnya.

Genetic Algorithm memiliki keunggulan dalam menemukan solusi melalui mekanisme seleksi, *crossover*, dan mutasi. Dibandingkan dengan *Simulated Annealing* (SA) dan *Tabu Search* (TS), *Genetic Algorithm* lebih fleksibel dalam menangani adanya banyak parameter dan solusi yang dihasilkan bervariasi. Sementara itu, dibandingkan dengan *Ant Colony Optimization* (ACO) dan *Particle Swarm Optimization* (PSO), *Genetic Algorithm* unggul dalam melakukan penjadwalan yang kompleks, karena dapat mengakomodasi banyak aturan dan batasan. Namun, *Genetic Algorithm* memiliki keterbatasan dalam proses evolusi yang memerlukan waktu lama. Nilai fitness yang digunakan dalam evolusi juga hanya mengukur jumlah konflik, tanpa mempertimbangkan faktor lain seperti preferensi dosen, kapasitas ruang, atau kebutuhan khusus dari kelas tertentu. Karena itu, meskipun secara sistem sudah optimal, tetap diperlukan penyesuaian agar jadwal yang dihasilkan bisa diterapkan dengan baik di dunia nyata.

5. Kesimpulan

Implementasi *Genetic Algorithm* dalam penjadwalan kuliah praktikum di Universitas Amikom Yogyakarta terbukti efektif dalam mengatasi konflik jadwal dosen dan ruang, serta menghasilkan jadwal yang optimal dan bebas bentrok dengan nilai fitness maksimal sebesar 167. Pendekatan ini mengurangi intervensi manual dan meningkatkan efisiensi operasional akademik khususnya penjadwalan kuliah praktikum. Metode *targeted mutation* terbukti mampu mempertahankan kualitas solusi yang dihasilkan. Akan tetapi, penelitian ini masih sebatas pada penjadwalan kuliah praktikum dan belum mencakup

aktivitas akademik lainnya, seperti penjadwalan kuliah teori dan peminjaman laboratorium/studio. Ke depan, metode *Genetic Algorithm* dapat diimplementasikan untuk pengembangan sistem informasi penjadwalan kuliah praktikum yang terintegrasi dengan sistem ERP (*Enterprise Resource Planning*) perguruan tinggi guna mendukung pengambilan keputusan yang lebih baik.

Ucapan Terima Kasih: Penulis mengucapkan terima kasih kepada Lembaga Penelitian dan Pengabdian Masyarakat (LPPM) Universitas Amikom Yogyakarta atas dukungan pendanaan yang diberikan, sehingga penelitian ini dapat terlaksana dengan baik. Selain itu, ucapan terima kasih yang sebesar-besarnya juga disampaikan kepada pihak laboratorium Universitas Amikom Yogyakarta yang telah memberikan informasi dan data terkait sistem penjadwalan kuliah praktikum. Semoga hasil penelitian ini dapat memberikan manfaat untuk pengembangan sistem penjadwalan akademik yang lebih efisien dan optimal.

Referensi

- [1] P. R. Indonesia, *Peraturan Pemerintah Republik Indonesia Nomor 60 Tahun 1999 Tentang Pendidikan Tinggi*. 1999.
- [2] A. Rochman, "Penjadwalan Kuliah Menggunakan Metode Constraints Programming dan Simulated Annealing," *Semin. Nas. Apl. Teknol. Inf. (SNATI 2012)*, 2012, <https://journal.uui.ac.id/Snati/article/view/2916>.
- [3] S. Mauluddin, I. Ikbal, and A. Nursikuwagus, "Optimasi Aplikasi Penjadwalan Kuliah Menggunakan Algoritma Genetik," *J. Resti*, vol. 2, no. 3, pp. 792–799, 2018, <https://doi.org/10.29207/resti.v2i3.597>.
- [4] A. Nugroho, W. Priatna, and I. Romli, "Implementasi Algoritma Genetika untuk Optimasi Penjadwalan Mata Kuliah," *J. Penelit. Tek. Inform.*, vol. 1, no. 2, pp. 188–194, 2018, <https://doi.org/10.34012/jutikomp.v1i2.238>.
- [5] A. R. Juwita, A. R. Pratama, and T. Triono, "Penjadwalan Perkuliahan menggunakan Algoritma Particle Swarm Optimization (PSO)," *J. Sisfotek Glob.*, vol. 10, no. 1, pp. 18–26, 2020, <https://dx.doi.org/10.38101/sisfotek.v10i1.270>.
- [6] Wiktasaria and J. E. Suseno, "Metode Simulated Annealing untuk Optimasi Penjadwalan Perkuliahan Perguruan Tinggi," *J. Sist. Inf. Bisnis*, vol. 6, no. 2, pp. 133–143, 2016, <https://doi.org/10.21456/vol6iss2pp133-143>.
- [7] Y. M. Khader, Y. I. Nurhasanah, and A. D. Kartika, "Penjadwalan Matakuliah Menggunakan Algoritma Greedy (Studi Kasus Penjadwalan Semester Ganjil 2017-2018 Informatika Itenas)," *J. Ilm. Teknol. Inf. Terap.*, vol. 4, no. 3, pp. 207–213, 2018, <https://doi.org/10.33197/jitter.vol4.iss3.2018.168>.
- [8] D. Oktarina and A. Hajjah, "Perancangan Sistem Penjadwalan Seminar Proposal dan Sidang Skripsi dengan Metode Algoritma Genetika," *J. Inf. Syst. Informatics Eng.*, vol. 3, no. 1, pp. 32–40, 2019, <https://doi.org/10.35145/joisie.v3i1.421>.
- [9] A. Josi, "Implementasi Algoritma Genetika pada Aplikasi Penjadwalan Perkuliahan Berbasis Web dengan mengadopsi model Waterfall (Studi Kasus: STMIK Prabumulih)," *J. Inform. J. Pengemb. IT*, vol. 2, no. 2, pp. 77–83, 2017, <https://doi.org/10.30591/jpit.v2i2.517>.
- [10] Y. Elva, "Sistem Penjadwalan Mata Pelajaran Menggunakan Algoritma Genetika," *J. Teknol. Inf.*, vol. 3, no. 1, pp. 49–57, 2019, <https://doi.org/10.36294/jurti.v3i1.687>.
- [11] M. D. S. Sanapiah, A. Rofi'ah, H. D. Jayanti, A. G. Arliana, and V. N. Wijyaningrum, "Penyusunan Jadwal Asisten Praktikum Menggunakan Algoritma Genetika," *Sist. J. Sist. Inf.*, vol. 8, no. 2, pp. 282–287, 2019, <https://doi.org/10.32520/stmsi.v8i2.501>.
- [12] T. Handayani, D. H. Fudholi, and S. Rani, "Kajian Algoritma Optimasi Penjadwalan Mata Kuliah," *PETIR J. Pengkaj. dan Penerapan Tek. Inform.*, vol. 13, no. 2, pp. 212–222, 2020, <https://doi.org/10.33322/petir.v13i2.1027>.
- [13] A. Mardia and Sunarto, "Metode Algoritma Genetika untuk Penyusunan Jadwal Perkuliahan Program Studi Tadris Matematika UIN Sulthan Thaha Saifuddin Jambi," *AXIOM J. Pendidik. dan Mat.*, vol. 10, no. 2, pp. 143–157, 2021, <http://dx.doi.org/10.30821/axiom.v10i2.10336>.
- [14] H. Ardiansyah and M. B. S. Junianto, "Penerapan Algoritma Genetika untuk Penjadwalan Mata Pelajaran," *J. Media Inform. Budidarma*, vol. 6, no. 1, pp. 329–336, 2022, <https://doi.org/10.30865/mib.v6i1.3418>.
- [15] N. Shofiyati, M. U. Musthofa, and A. M. Putra, "Implementasi Algoritma Genetika dalam Eliminasi Bentrokan Jadwal Perkuliahan di Politeknik Unisma Malang," *J. INSTEK Inform. dan Teknol.*, vol. 6, no. 1, 2021, <https://doi.org/10.24252/instek.v6i1.16587>.
- [16] L. A. Pangestu, S. H. Suryawan, and A. J. Latipah, "Penerapan Algoritma Genetika Dalam Penjadwalan Mata Pelajaran," *J. Inform.*, vol. 10, no. 2, pp. 194–205, 2023, <https://doi.org/10.31294/inf.v10i2.16701>.