

Kombinasi Crossover dan Mutasi Terbaik pada Algoritma Genetika dalam Penjadwalan Mata Kuliah

The Best Combination of Crossover and Mutation in Genetic Algorithms for Course Scheduling

Maria Karmelia Fajarlestari¹, Ignasius Boli Suban²

^{1,2}Program Studi Sistem Informasi, Fakultas Sains dan Teknologi
Universitas Pignatelli Triputra

E-mail: ¹mariakarmeliafl@gmail.com, ²ignasiusbolisuban@gmail.com

Abstrak

Pada proses penerapannya algoritma Genetika mempunyai operator *crossover* dan mutasi. Operator *crossover* mempunyai beberapa jenis dan operator mutasi dilakukan menurut besar probabilitasnya. Penggunaan *crossover* dan besar probabilitas menjadi salah satu masalah dalam penerapan algoritma Genetika karena dalam pemilihannya ditentukan secara random. Tujuan penelitian ini untuk mencari kombinasi paling baik pada jenis *crossover* dan besar probabilitas mutasi dalam memecahkan masalah penjadwalan. Kombinasi terbaik adalah kombinasi yang paling banyak menghasilkan hasil optimal. Algoritma Genetika diterapkan dalam permasalahan penjadwalan mata kuliah, kemudian hasil penerapannya dianalisis berdasarkan jenis mutasi dan besar probabilitas yang digunakan. Hasilnya dari semua kombinasi operator yang telah diuji coba untuk menyelesaikan masalah yang sama, ada satu kombinasi operator *crossover* dan mutasi yang memiliki rata-rata hasil terbaik yaitu kombinasi antara jenis *crossover* dua-titik dengan besar probabilitas mutasi 3%.

Kata kunci: Algoritma Genetika, Crossover, Mutasi, Permasalahan Penjadwalan

Abstract

In the implementation process, Genetic Algorithms employs two main operators: crossover and mutation. Crossover operators come in various types, while mutation operators are executed based on their respective probabilities. The utilization of crossover and the determination of mutation probabilities present a challenge in the application of Genetic Algorithms because they are chosen randomly. The research objective is to find the best combination of crossover types and mutation probabilities in solving scheduling problems. The best combination is defined as the one that produces the highest number of optimal results. Genetic Algorithms are applied to address course scheduling problems, and the outcomes are analyzed based on the types of mutation and the probabilities used. The results, all of the operator combinations tested for solving the same problem, there is one combination of crossover (two-point) and mutation (3% probability) that consistently yields the best average results.

Keywords: Genetic Algorithm, Crossover, Mutation, Scheduling Problem

1. PENDAHULUAN

Algoritma Genetika merupakan salah satu algoritma komputasi yang mengadopsi teori evolusi, dimana teori evolusi tersebut yang akan digunakan untuk mencari solusi dari suatu permasalahan optimasi [1,2]. Banyak permasalahan optimasi yang berhasil diselesaikan dengan algoritma ini, beberapa penelitian menggunakan Algoritma Genetika untuk menyelesaikan permasalahan jarak tempuh[3,4,5], permasalahan data citra digital [6,7,8], dan permasalahan penjadwalan juga banyak diselesaikan dengan Algoritma Genetika ini [9,10,11]. Salah satu permasalahan penjadwalan adalah permasalahan penjadwalan mata kuliah. Penjadwalan tersebut merupakan masalah yang kompleks untuk diselesaikan, banyak aspek-aspek yang perlu

dipertimbangkan dalam proses penjadwalan [12].

Dalam proses penerapannya algoritma Genetika mempunyai operator yang digunakan yaitu *crossover* dan mutasi. *Crossover* dilakukan menurut jenisnya sedangkan mutasi dilakukan berdasarkan besar probabilitasnya. Jenis *crossover* dan besar probabilitas mutasi ditentukan secara random, pemilihan jenis *crossover* dan besar probabilitas mutasi ini mempunyai pengaruh dalam hasil akhir penerapan algoritma Genetika [13,14].

Pada penelitian berjudul "Penjadwalan Mata Kuliah menggunakan Algoritma Genetika di Jurusan Sistem Informasi ITS", Algoritma Genetika digunakan untuk menyelesaikan mata kuliah, *crossover* yang digunakan adalah *crossover* dua-titik. Mutasi dilakukan opsional, yaitu ketika proses *crossover* belum menghasilkan hasil yang optimal [15]. Penelitian lain berjudul "Implementasi Algoritma Genetika untuk Aplikasi Penjadwalan Sistem Kerja Shift" mengimplementasikan algoritma genetika untuk aplikasi penjadwalan sistem kerja shift, operator Algoritma Genetika yang digunakan adalah *crossover* satu-titik tanpa menjelaskan kapan proses mutasi dilakukan [16]. Pada penelitian berjudul "Penerapan Algoritma Genetika untuk Penjadwalan Mata Pelajaran" tidak disebutkan jenis *crossover* yang digunakan, tetapi dijelaskan bahwa proses mutasi dilakukan terlebih dahulu sebelum proses *crossover* dilakukan [17]. Penelitian "A local search genetic algorithm for the job shop scheduling problem with intelligent agents" menggunakan *partially matched crossover*, yang hampir mirip dengan *crossover* dua-titik, dua titik persilangan dipilih dari kromosom secara acak dan seimbang. Kemudian gen-gen kromosom yang berada di antara titik persilangan ditukar dan menghasilkan dua kromosom baru [18].

Hal-hal ini yang menjadikan latar belakang penelitian ini. Belum ada kombinasi jenis *crossover* dan besar probabilitas mutasi yang direkomendasikan dalam penerapan Algoritma Genetika untuk menyelesaikan permasalahan penjadwalan mata kuliah. Tujuan dari penelitian ini adalah membandingkan kombinasi jenis *crossover* dan besar probabilitas mutasi dalam menyelesaikan masalah penjadwalan mata kuliah dengan Algoritma Genetika. Tolok ukur perbandingannya adalah hasil optimal yang didapat dari kombinasi operator-operator yang digunakan. Dalam penerapannya akan mencari nilai optimal yang diperoleh masing-masing kombinasi jenis *crossover* dan besar probabilitas mutasi berupa jadwal kuliah yang memenuhi aspek-aspek penjadwalan. Kemudian mencari kombinasi jenis *crossover* dan besar probabilitas mutasi yang paling sesuai untuk digunakan dalam masalah penjadwalan kuliah.

2. METODE PENELITIAN

2.1 Perumusan Masalah

Tahap ini dilakukan dengan merumuskan masalah-masalah dan aturan-aturan yang ada dalam sistem penjadwalan mata kuliah yang diterapkan di Universitas Atma Jaya Yogyakarta pada Fakultas Teknik jurusan Teknik Informatika. Batasan masalah di penelitian ini adalah: 1. Mata kuliah yang dijadwalkan tidak termasuk jadwal mata kuliah praktikum, 2. Jumlah hari, sesi, ruang dan mata kuliah sudah ditentukan dari awal, 3. Jadwal mata kuliah yang baik adalah memenuhi semua kriteria atau hampir memenuhi kriteria yang sudah ditentukan sebelumnya, 4. Operator *crossover* yang digunakan adalah *crossover* dua-titik dan *crossover* empat-titik, 5. Operator mutasi atau besar probabilitas mutasi = 1%, 2%, 3%, 4%, dan 5%.

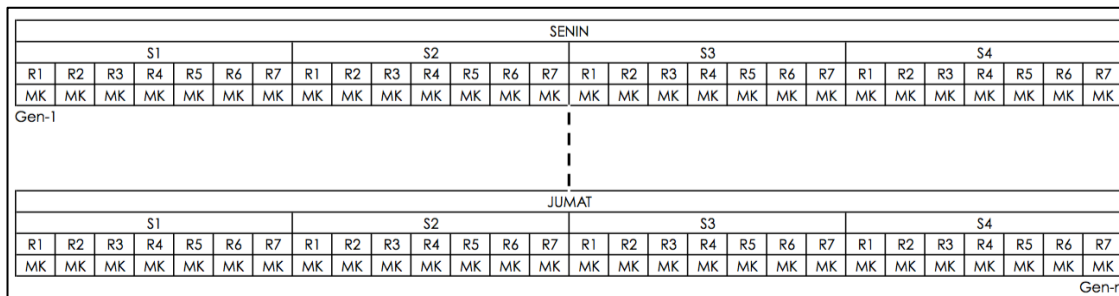
2.2 Perancangan dan Penerapan Algoritma Genetika

Tahap ini dilakukan dengan melakukan perancangan terhadap algoritma genetika. Proses algoritma Genetika adalah sebagai berikut:

2.2.1 Pembentukan dan Pemodelan Kromosom

Pembentukan kromosom dilakukan secara acak, mata kuliah (MK) ditempatkan pada setiap sesi (S), ruang (R) dan hari tertentu.

Pemodelan kromosom menggunakan pemodelan nilai. Dalam masalah penjadwalan kuliah, mata kuliah sebagai populasi dalam kromosom, maka kode dibuat untuk masing-masing data mata kuliah.



Gambar 1. Ilustrasi Pemodelan Kromosom

Ilustrasi pemodelan kromosom dapat dilihat pada gambar 1. Setiap gen pada kromosom ditempati satu hari, satu sesi, satu ruang dan satu mata kuliah yang dapat diartikan sebagai berikut:
 Gen-n = [hari; sesi; ruang; mata kuliah]

- Gen-1 = [MONDAY; 1; 1; MK]
 = [hari MONDAY; sesi 1; ruang 1; mata kuliah]
- Gen-2 = [MONDAY; 1; 2; MK]
 = [hari MONDAY; sesi 1; ruang 2; mata kuliah]
- Gen-3 = [MONDAY; 1; 3; MK]
 = [hari MONDAY; sesi 1; ruang 3; mata kuliah]

...

- Gen-139 = [FRIDAY; 4; 6; MK]
 = [hari FRIDAY; sesi 4; ruang 6; mata-kuliah]
- Gen-140 = [FRIDAY; 4; 7; MK]
 = [hari FRIDAY; sesi 4; ruang 7; mata kuliah]

2.2.2 Pembangkitan Kromosom

Sebagai populasi awal, dibangkitkan 100 kromosom dengan jumlah gen 140. Tiap-tiap gen pada kromosom ditempati oleh sesi, ruang, hari kemudian diisi mata kuliah secara acak.

2.2.3 Perhitungan Nilai Fitness

Nilai *fitness* digunakan sebagai indikasi sebuah kromosom dikatakan baik atau buruk, kromosom dikatakan baik jika mempunyai nilai *fitness* kecil, begitu juga sebaliknya. Dalam permasalahan penjadwalan kuliah pada penelitian ini, aturan-aturan penjadwalan yang tidak boleh dilakukan dengan masing-masing nilai *fitness* yang ditunjukkan pada Tabel 1. Nilai *fitness* dikonversi dari nilai *fitness* terbesar menjadi nilai *fitness* terkecil dengan rumus sebagai berikut :

$$F' = \frac{\sum F}{F} \quad (1)$$

Keterangan :

- F' = nilai fitness setelah dikonversi
- F = nilai fitness tiap kromosom

Tabel 1 Aturan-aturan penjadwalan kuliah

NO	ATURAN	PELANGGARAN
1	Mata kuliah semester 1, 3, 5, 7 sebisa mungkin tidak berada dalam satu sesi.	Nilai fitness = +1, jika mata kuliah semester 1, 3, 5, 7 berlangsung dalam satu sesi.
2	Mata kuliah semester 2, 4, 6, 8 7 sebisa mungkin tidak berada dalam satu sesi.	Nilai fitness = +1, jika mata kuliah semester 2, 4, 6, 8 7 berlangsung dalam satu sesi.

3	Mata kuliah pilihan tidak boleh berada dalam hari dan sesi yang sama	Nilai fitness = +1, jika mata kuliah pilihan berada dalam hari dan sesi yang sama
4	Semua dosen tetap tidak boleh mengajar pada hari senin sesi 3	Nilai fitness = +1, jika ada dosen tetap yang mengajar pada hari senin sesi 3
5	Semua jadwal mengajar dosen harus sesuai dengan permintaan jadwal dosen	Nilai fitness = +1, jika ada jadwal mengajar dosen yang tidak sesuai dengan permintaan jadwal dosen

2.2.4 Seleksi Kromosom

Berdasarkan nilai *fitness* pada proses sebelumnya, seleksi kromosom dilakukan dengan menggunakan metode Roda Roulette. Kromosom mempunyai kemungkinan terpilih lebih besar sebagai jika nilai *fitness* yang diperoleh baik, kromosom terpilih akan dijadikan kromosom orang tua untuk proses reproduksi. Dari total nilai *fitness* dalam populasi, selanjutnya akan dihitung probabilitas dari tiap kromosom pada populasi. Perhitungan probabilitas dirumuskan sebagai berikut:

$$Pk = \frac{t}{s} \times 100 \quad (2)$$

Keterangan :

Pk = besar probabilitas kromosom/individu

t = nilai *fitness* kromosom/individu

s = total nilai *fitness* kromosom/individu

Proses seleksi dilakukan dengan menggambarkan masing-masing kromosom menjadi sebuah Roda Roulette berdasarkan besar probabilitasnya, kemudian menentukan satu titik sebelum Roda Roulette diputar dan berhenti dengan sendirinya pada satu kromosom terpilih.

2.2.5 Proses Reproduksi

Proses reproduksi dilakukan dengan melakukan *crossover* dan operasi mutasi pada kromosom secara bergantian pada setiap iterasi. Tujuan dari proses ini adalah untuk menghasilkan kromosom/individu baru yang diharapkan bisa lebih baik dari kromosom sebelumnya/kromosom orang tua. Dalam penelitian ini proses reproduksi dilakukan terus-menerus dan hanya akan berhenti jika menemukan kondisi berikut: 1. Proses sudah berlangsung sebanyak jumlah iterasi, jumlah iterasi sudah ditentukan sebelumnya, 2. Kromosom baru hasil reproduksi mempunyai nilai optimal. Hasil akhir dalam proses reproduksi nantinya akan menghasilkan kromosom terbaik.

Operasi *crossover* dilakukan melalui persilangan dua kromosom yaitu dua kromosom yang sebelumnya sudah dipilih sebagai kromosom orang tua. Dalam setiap iterasi dilakukan proses *crossover* ini dengan dengan syarat tidak adanya proses mutasi dalam iterasi yang sama.

Mutasi dilakukan dengan menukar gen atau bagian dari suatu kromosom orang tua. hasilnya adalah satu kromosom baru. Operasi mutasi dilakukan menyesuaikan dengan besaran probabilitas, probabilitas mutasi yang akan menentukan kapan mutasi akan dilakukan dan berapa kali proses mutasi akan dilakukan. Jumlah mutasi yang akan dilakukan dihitung dengan rumus:

$$Jumlah\ mutasi = \frac{probabilitas\ mutasi}{100} \times jumlah\ iterasi \quad (3)$$

Iterasi mutasi atau kapan operasi mutasi dilakukan yaitu mencari kelipatan ke berapa iterasi yang akan dilakukan proses mutasi, yang dihitung dengan rumus:

$$Iterasi\ mutasi = \frac{jumlah\ iterasi}{jumlah\ mutasi} \quad (4)$$

2.2.6 Update Generasi

Update generasi dilakukan dengan tujuan mengganti kromosom terburuk menjadi kromosom yang lebih baik dalam populasi. Proses ini dilakukan setelah proses reproduksi,

kromosom hasil reproduksi akan mempunyai nilai *fitness* baru, ada dua kemungkinan nilai *fitness* yang diperoleh yaitu bisa lebih baik dari nilai *fitness* sebelumnya atau lebih buruk. *Update* generasi hanya akan dilakukan jika kromosom baru yang dihasilkan mempunyai nilai *fitness* lebih baik dari nilai *fitness* kromosom paling buruk dalam populasi. Jika kromosom yang dihasilkan mempunyai nilai *fitness* lebih buruk dari nilai *fitness* kromosom paling buruk dalam populasi maka proses ini tidak dilakukan dan kembali pada proses reproduksi kromosom. Sampai dengan proses reproduksi yang menghasilkan kromosom yang lebih baik. Hasil dari tahap update generasi ini adalah adanya populasi baru, yaitu populasi yang lebih baik untuk melanjutkan proses reproduksi.

2.3 Implementasi

Tahap ini dilakukan dengan membangun sistem sederhana berdasarkan perancangan dan penerapan algoritma Genetika menggunakan bahasa pemrograman Java.

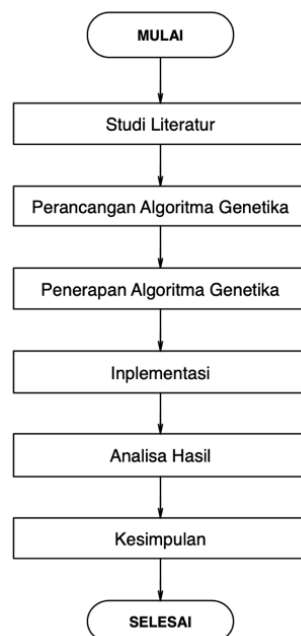
2.4 Analisa Hasil dan Kesimpulan

Tahap analisa hasil dilakukan dengan eksekusi sistem yang telah dibangun kemudian menganalisa hasil, hasil di analisa dari masing-masing nilai *fitness* yang diperoleh dari percobaan yang dilakukan menggunakan sistem penjadwalan yang meliputi semua kombinasi jenis *crossover* dan besar probabilitas mutasi.

2.5 Kesimpulan

Tahap kesimpulan dilakukan dengan menyimpulkan kombinasi *crossover* dan mutasi mana yang mempunyai hasil terbaik.

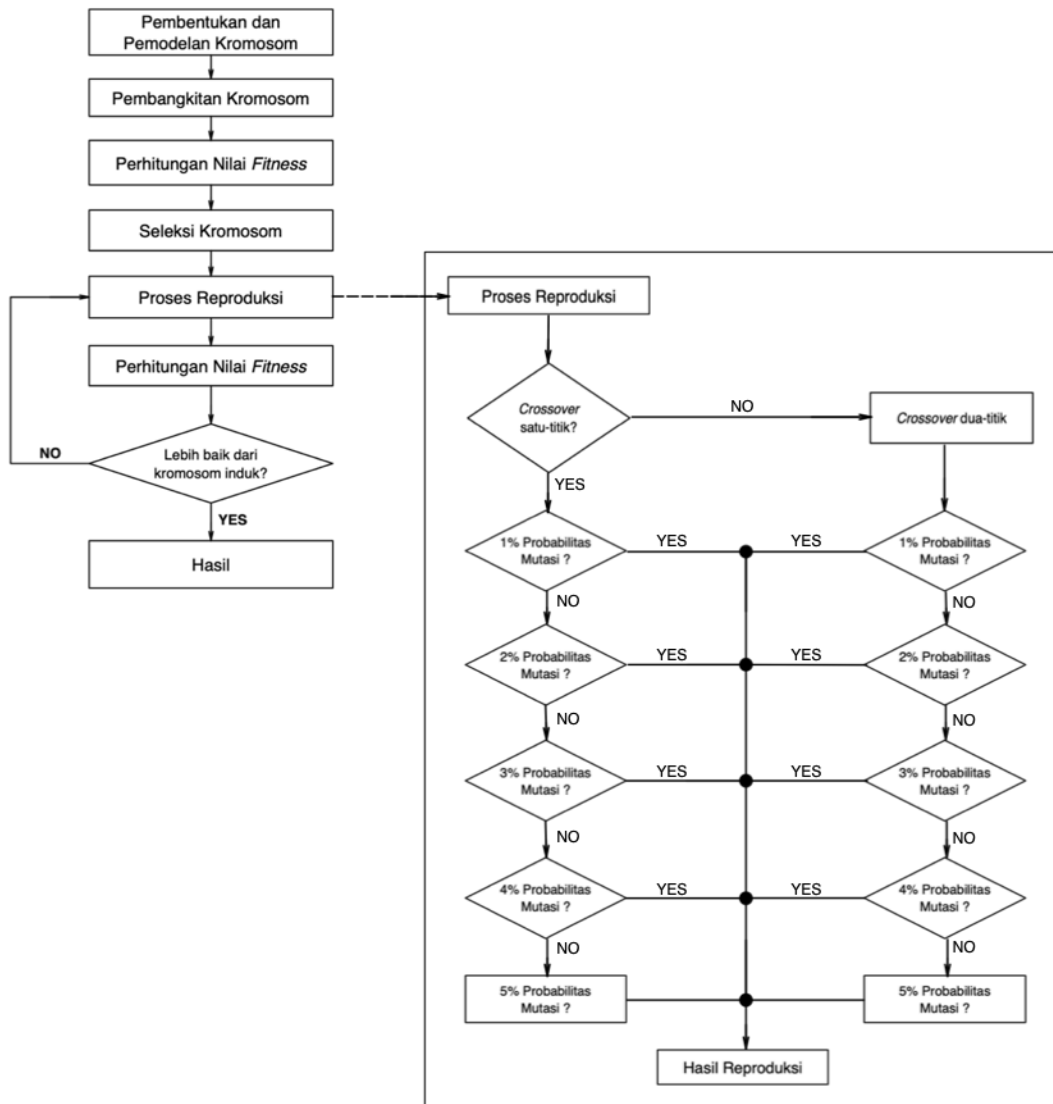
Metode penelitian dilakukan dalam beberapa tahap penelitian yang dapat dilihat pada Gambar 5. Diagram Alir Penelitian menggambarkan alur dari setiap proses penelitian yang telah dijelaskan sebelumnya. Proses diawali dengan Studi Literatur, kemudian merumuskan masalah yang akan diselesaikan dengan Algoritma Genetika. Setelah itu Algoritma Genetika dirancang sesuai dengan masalah yang akan diselesaikan. Selanjutnya rancangan Algoritma Genetika diterapkan dengan data-data yang digunakan dalam proses penjadwalan dan diimplementasikan dalam program. Kemudian proses analisa hasil dan terakhir menarik kesimpulan.



Gambar 5. Diagram Alir Penelitian

Proses perancangan Algoritma genetika dapat dilihat pada gambar 6, yang membedakan dengan proses Algoritma Genetika pada penelitian-penelitian sebelumnya adalah pada penelitian

ini proses reproduksi dapat diatur sesuai jenis crossover dan besar probabilitas mutasi yang ingin digunakan. Analisa Hasil pada penelitian ini berfokus pada proses reproduksi dalam Algoritma Genetika, proses reproduksi dilakukan dengan menjalankan semua kemungkinan kombinasi jenis *crossover* dan besar probabilitas mutasi. Hasil beberapa percobaan dari masing-masing kombinasi *crossover* dan mutasi akan dianalisa, hasil akhir dari proses tersebut nantinya yang akan menjadi kontribusi dari penelitian ini.



Gambar 6. Proses Perancangan Algoritma Genetika

3. HASIL DAN PEMBAHASAN

Pengujian algoritma Genetika dilakukan dengan menjalankan program untuk masing-masing kombinasi dan pengaturan operator untuk masing-masing pengujian.

3.1 Jumlah Iterasi

Jumlah iterasi yang digunakan dalam pengujian ini sebanyak 100 kali.

3.2 Jumlah Populasi

Jumlah kromosom dalam populasi sebanyak 10 kromosom.

3.3 Pengujian Tiap Metode

Metode 1

Jenis *Crossover* = *Crossover* dua-titik

Probabilitas mutasi = 1%

Jumlah iterasi = $1/100 * 1000 = 10$

Iterasi Mutasi = $1000/10 = 100$

Jadi operator mutasi dilakukan sebanyak 10 kali pada kelipatan iterasi 100, sisanya setiap iterasi dilakukan *crossover* dua-titik.

Metode 2

Jenis *Crossover* = *Crossover* dua-titik

Probabilitas mutasi = 2%

Jumlah iterasi = $2/100 * 1000 = 20$

Iterasi Mutasi = $1000/20 = 50$

Jadi operator mutasi dilakukan sebanyak 20 kali pada kelipatan iterasi 50, sisanya setiap iterasi dilakukan *crossover* dua-titik.

Metode 3

Jenis *Crossover* = *Crossover* dua-titik

Probabilitas mutasi = 3%

Jumlah iterasi = $3/100 * 1000 = 30$

Iterasi Mutasi = $1000/30 = 33$

Jadi operator mutasi dilakukan sebanyak 30 kali pada kelipatan iterasi 33, sisanya setiap iterasi dilakukan *crossover* dua-titik.

Metode 4

Jenis *Crossover* = *Crossover* dua-titik

Probabilitas mutasi = 4%

Jumlah iterasi = $4/100 * 1000 = 40$

Iterasi Mutasi = $1000/40 = 25$

Jadi operator mutasi dilakukan sebanyak 40 kali pada kelipatan iterasi 25, sisanya setiap iterasi dilakukan *crossover* dua-titik.

Metode 5

Jenis *Crossover* = *Crossover* dua-titik

Probabilitas mutasi = 5%

Jumlah iterasi = $5/100 * 1000 = 50$

Iterasi Mutasi = $1000/50 = 20$

Jadi operator mutasi dilakukan sebanyak 50 kali pada kelipatan iterasi 20, sisanya setiap iterasi dilakukan *crossover* dua-titik.

Metode 6

Jenis *Crossover* = *Crossover* empat-titik

Probabilitas mutasi = 1%

Jumlah iterasi = $1/100 * 1000 = 10$

Iterasi Mutasi = $1000/10 = 100$

Jadi operator mutasi dilakukan sebanyak 10 kali pada kelipatan iterasi 100, sisanya setiap iterasi dilakukan *crossover* empat-titik.

Metode 7

Jenis *Crossover* = *Crossover* empat-titik

Probabilitas mutasi = 2%

Jumlah iterasi = $2/100 * 1000 = 20$
 Iterasi Mutasi = $1000/20 = 50$
 Jadi operator mutasi dilakukan sebanyak 20 kali pada kelipatan iterasi 50, sisanya setiap iterasi dilakukan *crossover* empat-titik.

Metode 8

Jenis *Crossover* = *Crossover* empat-titik
 Probabilitas mutasi = 3%
 Jumlah iterasi = $3/100 * 1000 = 30$
 Iterasi Mutasi = $1000/30 = 33$
 Jadi operator mutasi dilakukan sebanyak 30 kali pada kelipatan iterasi 33, sisanya setiap iterasi dilakukan *crossover* empat-titik.

Metode 9

Jenis *Crossover* = *Crossover* empat-titik
 Probabilitas mutasi = 4%
 Jumlah iterasi = $4/100 * 1000 = 40$
 Iterasi Mutasi = $1000/40 = 25$
 Jadi operator mutasi dilakukan sebanyak 40 kali pada kelipatan iterasi 25, sisanya setiap iterasi dilakukan *crossover* empat-titik.

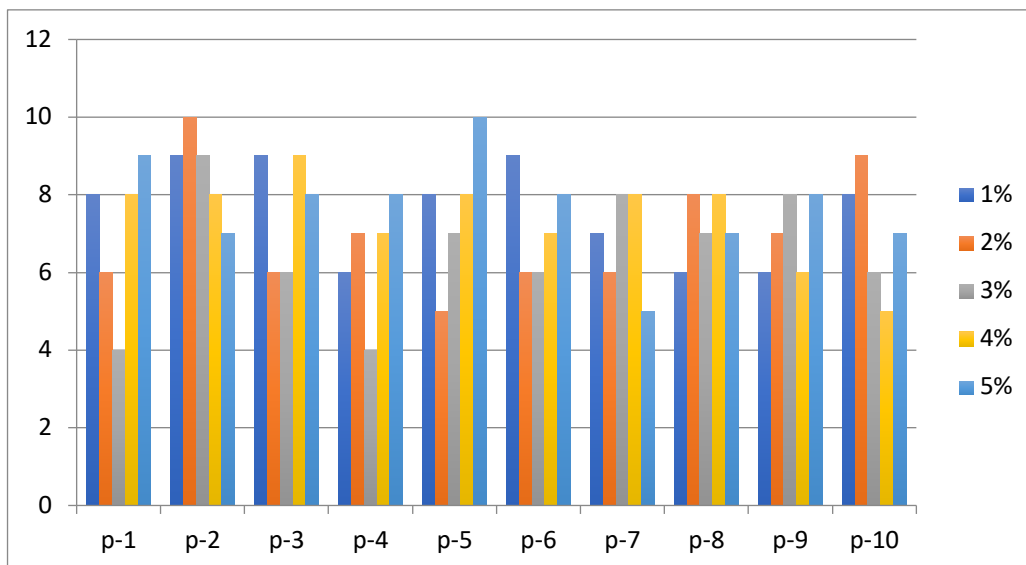
Metode 10

Jenis *Crossover* = *Crossover* empat-titik
 Probabilitas mutasi = 5%
 Jumlah iterasi = $5/100 * 1000 = 50$
 Iterasi Mutasi = $1000/50 = 20$
 Jadi operator mutasi dilakukan sebanyak 50 kali pada kelipatan iterasi 20, sisanya setiap iterasi dilakukan *crossover* empat-titik.

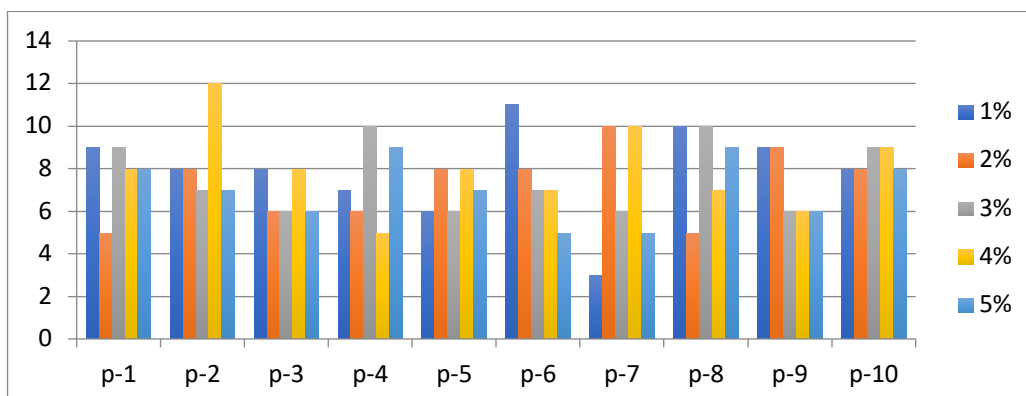
Program dijalankan sebanyak 10 kali untuk masing-masing metode pengujian, pada tabel 2 menjelaskan hasil nilai *fitness* yang diperoleh dari masing-masing kombinasi jenis *crossover* dan besar probabilitas mutasi.

Tabel 2 Hasil Nilai *Fitness* Terbaik

Metode	Jenis <i>Crossover</i>	Probabilitas Mutasi	Nilai <i>Fitness</i> Terbaik										Rata-rata nilai <i>Fitness</i>
			P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	
1	<i>Crossover</i> dua-titik	1%	2	5	4	1	3	0	2	2	2	2	2,3
2		2%	5	0	4	0	4	4	0	3	4	6	3
3		3%	2	4	2	2	2	5	2	2	0	4	2,5
4		4%	3	3	3	0	2	2	2	0	1	2	1,8
5		5%	2	4	0	2	0	4	3	0	0	0	1,5
6	<i>Crossover</i> empat-titik	1%	2	3	2	4	4	2	3	5	2	2	2,9
7		2%	3	4	3	5	0	2	2	2	4	5	3
8		3%	3	2	0	2	4	3	2	5	4	2	2,7
9		4%	4	0	2	2	4	3	4	4	2	1	2,6
10		5%	2	2	2	4	6	1	4	2	4	4	3,1



Gambar 7. Grafik Hasil Nilai *Fitness* Terbaik dengan *Crossover* Dua-titik



Gambar 8. Grafik Hasil Nilai *Fitness* Terbaik dengan *Crossover* Empat-titik

Dari hasil nilai *fitness* terbaik pada tabel 2 dan grafik hasil nilai *fitness* terbaik pada gambar 7 dan gambar 8, untuk masing-masing kombinasi jenis *crossover* dan besar probabilitas mutasi mempunyai hasil nilai *fitness* yang random, tidak ada kombinasi keduanya yang mempunyai nilai *fitness* selalu baik. Namun dari rerata total nilai *fitness* yang dihasilkan dalam 10 kali percobaan, kombinasi *crossover* dua titik dengan probabilitas mutasi 3% mempunyai rata-rata *fitness* terendah dibandingkan kombinasi lainnya. Maka dapat disimpulkan bahwa kombinasi tersebut adalah kombinasi *crossover* dan mutasi terbaik.

4. KESIMPULAN

Beberapa penelitian sebelumnya *crossover* ditentukan secara random menggunakan jenis *crossover* satu-titik atau dua-titik, beberapa penelitian menggabungkan aturan lain untuk menunjang proses *crossover*. Proses mutasi juga ditentukan secara acak. Pada penelitian ini Algoritma Genetika diterapkan untuk menyelesaikan masalah penjadwalan mata kuliah dengan menerapkan semua kemungkinan kombinasi *crossover* dan mutasi. Masing-masing kombinasi jenis *crossover* dan besar probabilitas mutasi hampir mempunyai hasil nilai *fitness* yang random, namun dari banyak percobaan dan analisa yang dilakukan, kombinasi jenis *crossover* dua-titik dan besar probabilitas mutasi 3% mempunyai rata-rata hasil nilai *fitness* lebih baik dibandingkan

dengan kombinasi lainnya. Maka dapat disimpulkan bahwa pada penelitian ini penulis memberikan suatu gagasan atau rekomendasi untuk Algoritma Genetika dalam penyelesaian masalah penjadwalan, yaitu dengan menggunakan jenis *crossover* dua-titik dan besar probabilitas mutasi 3%. Kombinasi tersebut menjadi kombinasi terbaik dalam menyelesaikan masalah penjadwalan mata kuliah dengan Algoritma Genetika.

5. SARAN

Analisis kombinasi *crossover* dan mutasi dapat dilanjutkan dengan mengkombinasikan operator lainya seperti jenis seleksi, cara pembangkitan kromosom, atau jumlah kromosom dalam proses evolusi.

DAFTAR PUSTAKA

- [1] S. D. Immanuel and U. Kr. Chakraborty, "Genetic Algorithm: An Approach on Optimization," in *2019 International Conference on Communication and Electronics Systems (ICCES)*, IEEE, Jul. 2019, pp. 701–708. doi: 10.1109/ICCES45898.2019.9002372.
- [2] I. Sumadireja, C. Prianto, and M. H. K. Saputra, *Optimasi nilai pendapatan menggunakan algoritma genetika*. Bandung: Kreatif Industri Nusantara, 2020.
- [3] D. A. Suprayogi and W. F. Mahmudy, "Penerapan Algoritma Genetika Traveling Salesman Problem with Time Window: Studi Kasus Rute Antar Jemput Laundry," *Jurnal Buana Informatika*, vol. 6, no. 2, May 2015, doi: 10.24002/jbi.v6i2.407.
- [4] T. N. Suharsono and M. R. Saddat, "Penentuan Optimalisasi TSP (Travelling Salesman Problem) Distribusi Barang Menggunakan Algoritma Genetika di Buka Mata Adv," in *SENTER 2017: Seminar Nasional Teknik Elektro 2017*, Bandung, Dec. 2017, pp. 326–335.
- [5] N. K. Nissaa, Farikhinb, and B. Surarso, "Analisis Pengaruh Operator Genetik pada Algoritma Genetika dan Penerapannya pada Traveling Salesman Problem (TSP)," in *PRISMA, Prosiding Seminar Nasional Matematika*, Semarang, 2020, pp. 1–7.
- [6] E. Supomo, A. Sunyoto, and M. P. Kurniawan, "Optimasi Peletakan Watermark pada Citra Digital Menggunakan Algoritma Genetika," *COREAI: Jurnal Kecerdasan Buatan, Komputasi dan Teknologi Informasi*, vol. 2, no. 2, pp. 11–17, Feb. 2022, doi: 10.33650/coreai.v2i2.3216.
- [7] A. Riski, A. Saiful Rizal, and A. Kamsyakawuni, "Pengamanan Citra Dengan Operator Algoritma Genetika," *Fountain of Informatics Journal*, vol. 4, no. 1, p. 13, May 2019, doi: 10.21111/fij.v4i1.2906.
- [8] I. A. Pardosi, P. Sirait, K. -, S. Goh, and R. Chandra, "Perbaikan Citra Gelap dan Pembesaran Objek Citra Menggunakan Gradient Based Low-Light Image Enhancement dan Rational Ball Cubic B-Spline With Genetic Algorithm," *Jurnal SIFO Mikroskil*, vol. 20, no. 2, Oct. 2019, doi: 10.55601/jsm.v20i2.674.
- [9] W. Cuiyu, L. Yang, and L. Xinyu, "Solving flexible job shop scheduling problem by a multi-swarm collaborative genetic algorithm," *Journal of Systems Engineering and Electronics*, vol. 32, no. 2, pp. 261–271, Apr. 2021, doi: 10.23919/JSEE.2021.000023.
- [10] W. Xu, H. Y. Sun, A. L. Awaga, Y. Yan, and Y. J. Cui, "Optimization approaches for solving production scheduling problem: A brief overview and a case study for hybrid flow shop using genetic algorithms," *Advances in Production Engineering & Management*, vol. 17, no. 1, pp. 45–56, Mar. 2022, doi: 10.14743/apem2022.1.420.
- [11] E. Sugiarto, S. Winarno, and A. Fahmi, "PENJADWALAN PERKULIAHAN OTOMATIS BERBASIS FUZZY LOGIC DAN GENETIC ALGORITHM PADA UNIVERSITAS DIAN NUSWANTORO," *Techno.COM*, vol. 14, no. 4, Nov. 2015.
- [12] M. Ridwan, "Prototype Sistem Pendukung Keputusan Untuk Penetapan Jadwal Kuliah Menggunakan Algoritma Genetika," *Systemic: Information System and Informatics Journal*, vol. 2, no. 2, pp. 9–18, Dec. 2016, doi: 10.29080/systemic.v2i2.109.
- [13] B. Alhijawi and A. Awajan, "Genetic algorithms: theory, genetic operators, solutions, and applications," *Evol Intell*, Feb. 2023, doi: 10.1007/s12065-023-00822-6.

- [14] S. Katoch, S. S. Chauhan, and V. Kumar, "A review on genetic algorithm: past, present, and future," *Multimed Tools Appl*, vol. 80, no. 5, pp. 8091–8126, Feb. 2021, doi: 10.1007/s11042-020-10139-6.
- [15] W. A. Puspaningrum, A. Djunaidy, and R. A. Vinarti, "PENJADWALAN MATA KULIAH MENGGUNAKAN ALGORITMA GENETIKA DI JURUSAN SISTEM INFORMASI ITS," *JURNAL TEKNIK POMITS*, vol. 21, no. 1, 2013.
- [16] W. Priatna, J. Warta, and D. Sulistiyo, "Implementasi Algoritma Genetika untuk Aplikasi Penjadwalan Sistem Kerja Shift," *Techno.Com*, vol. 22, no. 1, pp. 235–246, Feb. 2023, doi: 10.33633/tc.v22i1.7049.
- [17] H. Ardiansyah and M. B. S. Junianto, "Penerapan Algoritma Genetika untuk Penjadwalan Mata Pelajaran," *JURNAL MEDIA INFORMATIKA BUDIDARMA*, vol. 6, no. 1, p. 329, Jan. 2022, doi: 10.30865/mib.v6i1.3418.
- [18] L. Asadzadeh, "A local search genetic algorithm for the job shop scheduling problem with intelligent agents," *Comput Ind Eng*, vol. 85, pp. 376–383, Jul. 2015, doi: 10.1016/j.cie.2015.04.006.