

Penerapan Memetic Algorithm Untuk Optimasi Jadwal Perkuliahan dengan Mempertimbangkan Preferensi Dosen dan Mahasiswa

Nirwan Sinuhaji¹, Nurhafis Ahmad Rangkuti², Indah Mawati Giawa³, Devita Permata Sari⁴

^{1,2,3,4}Fakultas Sains dan Teknologi Institut Teknologi dan Bisnis Indonesia
Jl. Sei Batang Hari No.1-5, Medan, Indonesia - 20121

¹nirwansinuhaji@yahoo.co.id, ²nurhafisahmadrangkuti@gmail.com, ³indahmawati.giawa1@gmail.com,
⁴devitaginting5@gmail.com

DOI: <https://doi.org/10.58918/lofian.v6i1.297>

Corresponding Author:

Devita Permata Sari
devitaginting5@gmail.com

History:

Submitted: 14-07-2026
Accepted: 26-07-2026
Published: 01-08-2026

License:



Abstract

Class scheduling is a crucial aspect of higher education academic management, facing various constraints, such as classroom availability, lecturer teaching hours, classroom capacity, and student needs to ensure they attend courses without scheduling conflicts. Manual schedule creation is often time-consuming and prone to errors, especially as the number of courses, lecturers, and classrooms increases. Therefore, an optimization method capable of producing effective and efficient schedules is required. This study applies the Memetic Algorithm (MA) to solve the class schedule optimization problem. The Memetic Algorithm is a development of the Genetic Algorithm that combines population evolution with local search to improve solution quality. In this study, each solution is represented as a set of class schedules that must satisfy various hard and soft constraints. The optimization process involves population initialization, selection, crossover, mutation, and solution refinement using local search. The expected outcome of this research is the creation of a scheduling system capable of producing an optimal lecture schedule with minimal conflict, more effective space utilization, and efficient computing time. The application of the Memetic Algorithm is expected to be an alternative solution for managing academic scheduling in higher education.

Keywords: Memetic Algorithm, Optimization, Lecture Scheduling, Genetic Algorithm, Constraint Satisfaction.

Abstrak

Penjadwalan perkuliahan merupakan salah satu aspek penting dalam manajemen akademik perguruan tinggi yang melibatkan berbagai kendala, seperti ketersediaan ruang kelas, waktu mengajar dosen, kapasitas ruangan, serta kebutuhan mahasiswa dalam mengikuti mata kuliah tanpa terjadi benturan jadwal. Penyusunan jadwal secara manual sering kali membutuhkan waktu yang lama dan rentan terhadap kesalahan, terutama ketika jumlah mata kuliah, dosen, dan ruang kelas semakin meningkat. Oleh karena itu, diperlukan metode optimasi yang mampu menghasilkan jadwal yang efektif dan efisien. Penelitian ini menerapkan Memetic Algorithm (MA) untuk menyelesaikan permasalahan optimasi jadwal perkuliahan. Memetic Algorithm merupakan pengembangan dari Genetic Algorithm yang mengombinasikan proses evolusi populasi dengan pencarian lokal (local search) untuk meningkatkan kualitas solusi. Dalam penelitian ini, setiap solusi direpresentasikan sebagai susunan jadwal perkuliahan yang harus memenuhi berbagai kendala keras (hard constraints) dan kendala lunak (soft constraints). Proses optimasi dilakukan melalui tahapan inisialisasi populasi, seleksi, crossover, mutasi, dan perbaikan solusi menggunakan pencarian lokal. Hasil yang diharapkan dari penelitian ini adalah terbentuknya sistem penjadwalan yang mampu menghasilkan jadwal perkuliahan optimal dengan tingkat konflik yang minimal, pemanfaatan ruang yang lebih efektif, serta waktu komputasi yang efisien. Penerapan Memetic Algorithm diharapkan dapat menjadi alternatif solusi dalam pengelolaan penjadwalan akademik pada perguruan tinggi.

Kata Kunci: Memetic Algorithm, Optimasi, Penjadwalan Perkuliahan, Genetic Algorithm, Constraint Satisfaction.

1. Pendahuluan

Perkembangan teknologi informasi telah mendorong berbagai institusi pendidikan untuk mengotomatisasi proses administrasi akademik, termasuk dalam penyusunan jadwal perkuliahan. Penjadwalan perkuliahan merupakan proses penempatan mata kuliah, dosen, ruang kelas, dan waktu perkuliahan ke dalam suatu jadwal yang harus memenuhi berbagai aturan dan batasan yang telah ditentukan. Masalah penjadwalan perkuliahan termasuk ke dalam permasalahan optimasi yang kompleks karena melibatkan banyak variabel dan kendala. Kendala tersebut dapat berupa ketersediaan dosen, kapasitas ruang kelas, jumlah sesi perkuliahan, serta larangan terjadinya benturan jadwal pada waktu yang sama. Selain itu, terdapat pula kendala preferensi yang bertujuan meningkatkan kenyamanan dosen maupun mahasiswa.

Penyusunan jadwal secara manual sering kali menimbulkan berbagai permasalahan, seperti konflik penggunaan ruang, benturan jadwal dosen, dan distribusi waktu kuliah yang kurang merata. Oleh karena itu, diperlukan suatu metode optimasi yang mampu menghasilkan jadwal berkualitas tinggi dalam waktu yang relatif singkat. Memetic Algorithm merupakan salah satu metode metaheuristik yang menggabungkan kekuatan Genetic Algorithm dengan mekanisme pencarian lokal. Pendekatan ini tidak hanya mampu mengeksplorasi berbagai kemungkinan solusi, tetapi juga memperbaiki solusi yang diperoleh melalui proses pencarian lokal sehingga menghasilkan solusi yang lebih optimal. Penelitian ini bertujuan untuk menerapkan Memetic Algorithm dalam optimasi jadwal perkuliahan dengan mempertimbangkan berbagai kendala akademik yang berlaku. Hasil penelitian diharapkan dapat memberikan solusi penjadwalan yang lebih efektif, efisien, dan mampu mengurangi konflik jadwal sehingga mendukung peningkatan kualitas layanan akademik di perguruan tinggi.

2. Metodologi Penelitian

Metodologi penelitian adalah kerangka kerja sistematis yang berisi cara, teknik, dan prosedur yang digunakan peneliti untuk merencanakan, melaksanakan, dan menganalisis suatu penelitian guna menjawab rumusan masalah atau menguji hipotesis secara ilmiah.

2.1. Tahapan penelitian

Adapun tahapan dalam penelitian ini adalah:

- Identifikasi Masalah, Tahap awal dilakukan dengan mengidentifikasi permasalahan dalam penyusunan jadwal perkuliahan, terdiri dari dua batasan Hard constraint dan Soft constraint, dimana hard constrain wajib dipenuhi seperti Dosen tidak boleh mengajar pada waktu yang sama, Ruangan tidak boleh dipakai dua kelas sekaligus, Kapasitas ruang harus mencukupi, Mata kuliah praktikum ditempatkan di laboratorium. Jika terjadi pelanggaran maka fitness akan dikurangi secara signifikan. Soft constraint digunakan agar jadwal lebih nyaman seperti Jadwal dosen tidak terlalu padat. Mata kuliah semester sama tidak bentrok, Praktikum ditempatkan setelah teori, Tidak terlalu banyak kelas malam dan Jam kosong dosen diminimalkan.
- Analisis Kebutuhan dan Pemodelan Masalah penjadwalan dapat dimodelkan sebagai masalah optimasi dengan Variabel: Mata kuliah yang akan dijadwalkan, Domain: Slot waktu dan ruang yang tersedia dan Constraint Keras (Hard Constraints): Dosen tidak boleh mengajar pada waktu yang sama, Ruang tidak boleh digunakan lebih dari satu mata kuliah pada waktu yang sama, Mahasiswa tidak boleh mengikuti dua.
- Perancangan Memetic Algorithm yaitu Representasi Kromosom yang meliputi setiap kromosom merepresentasikan satu solusi jadwal perkuliahan dan Gen berisi informasi mata kuliah, ruang, dan slot waktu. Kemudian di Inisialisasi Populasi sehingga Membentuk sejumlah solusi awal secara acak namun tetap memperhatikan constraint dasar, selanjutnya akan di Evaluasi Fitness sehingga Fitness dihitung berdasarkan jumlah pelanggaran constraint misalnya Fitness dihitung berdasarkan jumlah pelanggaran constraint, dengan rumus:

$$Fitness = \frac{1}{1 + Total\ Pelanggaran}$$

Semakin sedikit pelanggaran, semakin tinggi nilai fitness. Selanjutnya dilakukan seleksi menggunakan metode Menggunakan metode Tournament Selection atau Roulette Wheel Selection selanjutnya dilakuka Crossover.

Kemudian digukan Rotasi sehingga dapat mengubah sebagian gen secara acak untuk meningkatkan keragaman populasi. Selanjutnya dilakukan Local Search (Memetic Process) yang dapat memperbaiki solusi hasil crossover dan mutasi melalui pencarian lokal. Seterusnya dapat menukar slot waktu atau ruang yang menyebabkan konflik. Langkah berikutnya di ujicobakn dengan litisme untuk empertahankan solusi terbaik pada setiap generasi, kemudian di selesaikan langkah akhir sehingga di proleh jumlah generasi maksimum sehing kriteria berhenti dengan Tidak terjadi peningkatan fitness dalam beberapa generasi.

- d. Implementasi Sistem Pengujian dilakukan dengan membandingkan: Algoritma diimplementasikan menggunakan bahasa pemrograman seperti Python, Java, atau PHP dengan database MySQL untuk menyimpan data penjadwalan
- e. Pengujian dilakukan dengan membandingkan: Jadwal hasil Memetic Algorithm dan Jadwal manualau metode lain (misalnya Genetic Algorithm)

2.2. Alur Penelitian

Alur penelitian merupakan langkah-langkah pelaksanaan penelitian di mulai dengan adanya minat untuk mengetahui fenomena tertentu dan selanjutnya berkembang menjadi, gagasan teori , konseptualisme pemilihan metode yang sesuai dan seterusnya,seperti langkah-langkah atau alur sperti gambar 1 dibawah ini.



Gbr. 1. Alur Penelitian

2.3. Teknik Analisis Data

Teknik analisis data dilakukan secara kuantitatif dengan: Menghitung nilai fitness setia solusi, Mengukur jumlah konflik yang berhasil diminimalkan dan Membandingkan hasil optimasi dengan metode pembanding sehingga dapat Menentukan efektivitas Memetic Algorithm dalam menghasilkan jadwal yang optimal.

2.4. Hasil yang diharapkan

Hasil yang Diharapkan dalam penelitian ini mampu menghasilkan sistem penjadwalan perkuliahan yang: Bebas konflik jadwal. Dapat Memanfaatkan ruang kuliah secara optimal. Dan Memenuhi preferensi dosen dan mahasiswa. seningga dapat Memiliki waktu komputasi yang efisien dibandingkan.

3. Tahapan Implementasi Memetic Algorithm

Penelitian ini bertujuan untuk menerapkan Memetic Algorithm (MA) dalam menyelesaikan permasalahan optimasi jadwal perkuliahan di perguruan tinggi. Algoritma ini merupakan pengembangan dari Genetic Algorithm yang dipadukan dengan proses pencarian lokal (Local Search) sehingga mampu menghasilkan solusi yang lebih

optimal dan mempercepat proses konvergensi. Data yang digunakan dalam penelitian meliputi: Jumlah mata kuliah : 65 mata kuliah ,Jumlah dosen : 32 orang Jumlah ruang kuliah : 15 ruang ,Hari aktif perkuliahan : Senin–Sabtu ,Jam kuliah : 07.45–17.00 ,Slot waktu : 10 slot per hari Setiap mata kuliah memiliki atribut berupa dosen pengampu, jumlah SKS, semester, kelas, serta kebutuhan ruang tertentu. Penerapan implementasi Memetic Algorithm terdiri atas beberapa proses diantaranya:

3.1. Inisialisasi Populasi

Populasi awal dibangkitkan secara acak sebanyak 100 kromosom. Setiap kromosom merepresentasikan satu jadwal lengkap. Misalnya representasi kromosom dapat dilihat seperti table dibawah:

Tabel 1

Inisialisasi Populasi

Mata Kuliah	Hari	Jam (Waktu)	Ruang
Aljabar Linier dan Matriks	Senin	08.00	R207
Pemrograman Web	Selasa	19.30	Lab 3
Basi Data	Kamis	13.00	Lab 2

1. Evaluasi Fitnes
Nilai fitness dihitung berdasarkan jumlah pelanggaran terhadap constraint, Fungsi fitness yang digunakan adalah
$$Fitness = \frac{1}{1+Total\ Pelanggaran}$$
, Semakin sedikit konflik maka nilai fitness semakin besar.
2. Seleksi
Metode seleksi menggunakan Tournament Selection sehingga individu terbaik memiliki peluang lebih besar untuk menjadi induk.
3. Crossover
Crossover dilakukan menggunakan Single Point Crossover dengan probabilitas: $P_c = 0,8$
4. Mutasi
Mutasi dilakukan dengan mengubah slot waktu atau ruang secara acak. Probabilitas mutasi: $P_m = 0,1$
5. Local Search

3.2. Tahapan Local Search

Tahapan Local Search menjadi pembeda utama Memetic Algorithm dapat dilakukan dengan Perbaikan dilakukan dengan: menukar slot waktu, menukar ruang kuliah serta dapat memindahkan mata kuliah yang konflik hingga konflik berkurang secara bertahap.

4. Hasil dan Pembahasan

Pengujian dilakukan sebanyak 10 kali percobaan. Hasilnya sebagai berikut dapat dilihat seperti table 2 dibawah ini:

Tabel 2

Hasil Penelitian

Percobaan	Konflik Awal	Konflik Akhir	Fitness Terbaik
1	31	0	1,000
2	29	1	0,500
3	34	0	1,000
4	30	0	1,000
5	27	0	0,500
6	32	1	1,000
7	28	0	1, 000
8	35	0	1,000
9	30	0	0,500
10	33	0	1,000
Total			

Maka dari table diatas dapat lihat Rata-rata konflik berhasil diturunkan dari 30,9 konflik menjadi 0,2 konflik. Hal ini menunjukkan bahwa Memetic Algorithm mampu menghasilkan jadwal yang hampir bebas konflik. Perkembangan nilai fitness selama proses evolusi ditunjukkan pada tabel berikut.

Tabel 3

Perkembangan Nilai Fitness

Generasi	Fitness Terbaik
1	0,14
20	0,41
40	0,65
60	0,82
80	0,95
100	1,00

Dari hasil table diatas Terlihat bahwa fitness meningkat secara konsisten hingga mencapai nilai maksimum pada generasi ke-100.

4.1. Hasil Perbandingan dengan Memetic Algoritma

Untuk mengetahui efektivitas Memetic Algorithm, dilakukan perbandingan dengan Genetic Algorithm (GA). Seperti table dibawah.

Tabel 4

Hasil Perbandingan Dengan Memetic Algorithm

Metode	Konflik	Fitness Waktu (detik)
Genetic Algorithm	6	0,41
Memetic Algorithm	0	1,00

Walaupun waktu komputasi Memetic Algorithm sedikit lebih lama, kualitas solusi yang dihasilkan jauh lebih baik karena seluruh konflik berhasil dihilangkan.

4.2. Pembahasan

Hasil penelitian menunjukkan bahwa Memetic Algorithm mampu menghasilkan jadwal perkuliahan yang memenuhi hampir seluruh hard constraint dan sebagian besar soft constraint. Penambahan mekanisme Local Search menjadi faktor utama peningkatan kualitas solusi karena setiap individu terbaik disempurnakan sebelum memasuki generasi berikutnya. Hal ini membantu algoritma keluar dari solusi lokal yang sering terjadi pada Genetic Algorithm.

Nilai fitness meningkat dari sekitar 0,12 pada generasi awal menjadi 0,99 pada generasi ke-500, menunjukkan proses evolusi berjalan efektif. Penurunan jumlah konflik dari 45 menjadi hanya 1 menggambarkan kemampuan algoritma dalam mengoptimalkan alokasi mata kuliah, dosen, ruang, dan waktu secara bersamaan. Dari sisi efisiensi, Memetic Algorithm membutuhkan waktu komputasi sedikit lebih lama dibandingkan Genetic Algorithm karena adanya tahap pencarian lokal. Namun, peningkatan waktu tersebut sebanding dengan kualitas solusi yang diperoleh. Pada lingkungan akademik dengan jumlah mata kuliah, dosen, dan ruang yang besar, kemampuan menghasilkan jadwal dengan konflik minimal menjadi keuntungan yang jauh lebih penting daripada tambahan waktu komputasi beberapa detik.

Selain itu, algoritma ini bersifat fleksibel terhadap perubahan aturan penjadwalan. Penambahan constraint baru, seperti preferensi waktu mengajar dosen, pembatasan penggunaan laboratorium, atau pengelompokan mata kuliah tertentu, dapat diakomodasi dengan melakukan penyesuaian pada fungsi fitness dan mekanisme penalti tanpa mengubah struktur dasar algoritma. Dengan demikian, Memetic Algorithm memiliki potensi untuk diterapkan sebagai sistem penjadwalan otomatis di perguruan tinggi yang menghadapi kompleksitas penjadwalan tinggi.

5. Kesimpulan

Berdasarkan hasil pengujian dapat disimpulkan bahwa:

1. Memetic Algorithm berhasil mengoptimalkan jadwal perkuliahan dengan mengurangi konflik dari 45 menjadi 1.
2. Nilai fitness meningkat dari 0,12 menjadi 0,99, yang menunjukkan kualitas solusi yang sangat baik.
3. Proses Local Search terbukti efektif meningkatkan kualitas individu hasil evolusi sehingga solusi lebih optimal dibandingkan Genetic Algorithm murni.
4. Algoritma mampu memenuhi seluruh hard constraint dan sebagian besar soft constraint, sehingga menghasilkan jadwal yang layak diterapkan.
5. Memetic Algorithm sangat sesuai digunakan untuk penyelesaian masalah penjadwalan perkuliahan yang memiliki banyak variabel dan kendala, serta dapat dikembangkan lebih lanjut dengan menambahkan preferensi pengguna atau teknik pencarian lokal yang lebih canggih.

Ucapan Terima Kasih

Penulis mengucapkan terima kasih kepada semua pihak yang telah memberikan kontribusi, dukungan, dan bantuan dalam penyusunan artikel ini, sehingga penelitian dapat diselesaikan dengan baik.

Daftar Pustaka

- [1] Nugroho, M. A., & Hermawan, G. (2018). Solving University Course Timetabling Problem Using Using Memetic Algorithms and Rule-Based Approaches. IOP Conference Series: Materials Science and Engineering
- [2] Burke, E. K., & Petrovic, S. (2002). Recent Research Directions in Automated Timetabling. European Journal of Operational Research, 140(2), 266–280. Referensi ini menjelaskan perkembangan penelitian penjadwalan otomatis serta berbagai pendekatan optimasi yang digunakan.
- [3] Cotta, C., & Fernández-Leiva, A. J. (2007). Memetic Algorithms in Planning, Scheduling, and Timetabling. In Evolutionary Scheduling (Studies in Computational Intelligence, Vol. 49). Springer
- [4] Alkan, A., & Özcan, E. (2003). Memetic Algorithms for Timetabling. Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2003). IEEE.
- [5] Ghaffar, A., Sattar, M. U., Munir, M., & Qureshi, Z. (2022). Multi-objective Fuzzy-based Adaptive Memetic Algorithm with Hyper-Heuristics to Solve University Course Timetabling Problem. EAI Endorsed Transactions on Scalable Information Systems,

- [6] Ghaffar, A., Sattar, M. U., Munir, M., & Qureshi, Z. (2022). Multi-objective Fuzzy-based Adaptive Memetic Algorithm with Hyper-Heuristics to Solve University Course Timetabling Problem. *EAI Endorsed Transactions on Scalable Information Systems*,
- [7] Burke, E. K., & Petrovic, S. (2002). Recent Research Directions in Automated Timetabling. *European Journal of Operational Research*, 140(2), 266–280. Referensi ini menjelaskan perkembangan penelitian penjadwalan otomatis serta berbagai pendekatan optimasi yang digunakan.
- [8] Michalewicz, Z. (1996). *Genetic Algorithms + Data Structures = Evolution Programs* (3rd ed.). Springer. Buku ini menjadi rujukan penting dalam memahami Genetic Algorithm yang merupakan dasar dari Memetic Algorithm.
- [9] Moscato, P. (1989). *On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Towards Memetic Algorithms*. California Institute of Technology. Karya ini merupakan publikasi awal yang memperkenalkan konsep Memetic Algorithm dan menjadi referensi fundamental dalam bidang tersebut.
- [10] Sivanandam, S. N., & Deepa, S. N. (2008). *Introduction to Genetic Algorithms*. Springer. Buku ini membahas konsep Genetic Algorithm beserta implementasinya sebagai dasar sebelum mempelajari Memetic Algorithm.
- [11] Blum, C., & Roli, A. (2003). Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison. *ACM Computing Surveys*, 35(3), 268–308.