



Perbandingan Genetic Algorithm dan Queue-Based Scheduling untuk Penjadwalan Kuliah Otomatis di Perguruan Tinggi

Muhammad Demas Adi Pratama^{*}, Wakhid Kurniawan

Fakultas Sains Teknologi dan Peternakan, Program Studi Informatika, Universitas Muhammadiyah Karanganyar, Karanganyar, Indonesia

Email: ^{1*} muhdemas2003@gmail.com, ² kurniawan.wk48@gmail.com

Email Penulis Korespondensi: muhdemas2003@gmail.com

Abstrak—Penjadwalan perkuliahan merupakan aktivitas fundamental dalam manajemen akademik, namun kompleksitas kendala ruangan dan jam mengajar sering memicu benturan jadwal ketika diproses secara konvensional. Penelitian ini bertujuan untuk merancang sistem otomatisasi penjadwalan kuliah di Universitas Muhammadiyah Karanganyar (UMUKA) dengan membandingkan efektivitas *Genetic Algorithm* (GA) dan *Queue-Based Scheduling* (QBS). Sistem dibangun menggunakan framework Laravel dan *NoSQL* Database. Eksperimen komparatif dilakukan melalui enam skenario dinamis (mulai dari 20 hingga 263 mata kuliah) yang diukur menggunakan tiga metrik: jumlah konflik akhir, nilai fitness, dan waktu komputasi. Hasil pengujian menunjukkan bahwa QBS unggul mutlak dalam kecepatan komputasi (maksimal 4,18 detik), namun gagal menghasilkan kualitas jadwal yang proporsional karena nilai fitness menetap di angka 0. Sebaliknya, GA mampu menghasilkan jadwal tanpa konflik pada beban menengah dan mengompensasi sedikit konflik pada beban masif dengan mencetak lonjakan nilai fitness yang signifikan (mencapai 462,2). Meskipun membutuhkan waktu komputasi yang relatif lebih lama (36,2 detik pada beban tertinggi), durasi tersebut tergolong sangat efisien. Kesimpulannya, GA direkomendasikan sebagai algoritma utama dalam sistem penjadwalan kuliah karena mampu melakukan optimasi global secara proporsional. Kontribusi utama penelitian ini adalah memberikan dasar empiris dan arsitektur komparatif bagi perguruan tinggi dalam memilih algoritma penjadwalan yang paling adaptif terhadap keterbatasan infrastruktur ruang dan kompleksitas *hard-constraint* lokal.

Kata Kunci: Genetic Algorithm; Queue-Based Scheduling; Penjadwalan Kuliah; Optimasi Kombinatorial; Laravel

Abstract—Course scheduling is a fundamental activity in academic management; however, the complexity of room and teaching constraints often triggers schedule conflicts when processed conventionally. This study aims to design an automated course scheduling system at Universitas Muhammadiyah Karanganyar (UMUKA) by comparing the effectiveness of the *Genetic Algorithm* (GA) and *Queue-Based Scheduling* (QBS). The system was developed using the Laravel framework and a *NoSQL* Database. Comparative experiments were conducted through six dynamic scenarios (ranging from 20 to 263 courses), measured using three metrics: the final number of conflicts, fitness value, and computational time. The results indicated that QBS was absolutely superior in computational speed (maximum 4.18 seconds) but failed to produce proportional schedule quality, as the fitness value remained at 0. Conversely, GA successfully generated conflict-free schedules in medium workloads and compensated for minor conflicts in massive workloads by achieving a significant spike in fitness value (reaching 462.2). Although requiring relatively longer computational time (36.2 seconds at the highest load), this duration remains highly efficient. In conclusion, GA is recommended as the primary algorithm for course scheduling systems due to its ability to perform global optimization proportionally. The main contribution of this research is providing empirical foundations and a comparative architecture for higher education institutions in selecting the most adaptive scheduling algorithm for local room infrastructure limitations and *hard-constraint* complexity.

Keywords: Genetic Algorithm; Queue-Based Scheduling; Course Scheduling; Combinatorial Optimization; Laravel

1. PENDAHULUAN

Penjadwalan perkuliahan merupakan aktivitas fundamental dalam manajemen akademik perguruan tinggi karena menentukan pemanfaatan sumber daya utama seperti dosen, ruang kuliah, dan waktu belajar mahasiswa. Pada banyak institusi, khususnya di negara berkembang, proses ini masih dikerjakan secara manual atau semi-manual oleh bagian administrasi, sehingga membutuhkan waktu lama, rentan kesalahan, dan sulit beradaptasi terhadap perubahan dinamis. Kompleksitas penjadwalan meningkat seiring bertambahnya jumlah program studi, kelas paralel, serta variasi preferensi dan keterbatasan dosen maupun ketersediaan ruang (Mahlous & Mahlous, 2023). Secara khusus, penugasan dosen (*professor assignment*) menjadi salah satu kendala terberat karena harus menyesuaikan kesediaan waktu mengajar dosen dengan ketersediaan ruang kelas yang sesuai kapasitas (Arratia-Martinez et al., 2021). Perkembangan kecerdasan buatan terbukti mampu memberikan solusi teknologi yang presisi, tidak hanya untuk pelayanan publik, tetapi juga sangat krusial dalam otomatisasi manajemen akademik perguruan tinggi (Tristanti et al., 2025).

Di Universitas Muhammadiyah Karanganyar (UMUKA), proses penyusunan jadwal kuliah masih bergantung pada pengolahan data manual dan penyusunan draf jadwal oleh pengelola akademik. Kondisi ini menimbulkan beberapa permasalahan berulang, antara lain konflik jadwal mata kuliah yang terjadi pada waktu dan ruang yang sama, ketidakseimbangan beban mengajar dosen, keterbatasan kapasitas ruang dibandingkan jumlah mahasiswa, serta lamanya waktu penyusunan jadwal yang dapat mencapai beberapa hari hingga minggu untuk satu semester. Setiap perubahan data, seperti penambahan mata kuliah, perubahan kapasitas kelas, ataupun penyesuaian preferensi dosen, menuntut revisi manual yang memakan waktu dan berpotensi memunculkan konflik baru (Al-Fitouri et al., 2025). Transisi dari pencatatan jadwal yang manual menuju sistem informasi terpadu telah menjadi standar utama untuk meningkatkan efisiensi operasional di lingkungan institusi pendidikan (Supriyanti et al., 2023). Oleh karena itu, riset terkait optimasi penjadwalan (*timetabling problem*) terus berkembang dan menjadi salah satu area riset prioritas di lingkungan pendidikan tinggi (Apriliyanto et al., 2026; Chen et al., 2021).



Penelitian terdahulu menunjukkan bahwa penerapan algoritma optimasi dan kecerdasan buatan dapat secara signifikan meningkatkan kualitas jadwal dan menurunkan jumlah konflik dibandingkan pendekatan manual. *Genetic Algorithm* (GA) banyak digunakan untuk masalah penjadwalan kuliah karena kemampuannya mengeksplorasi ruang solusi yang besar melalui mekanisme seleksi, *crossover*, dan mutasi untuk mencari kombinasi penempatan mata kuliah, dosen, ruang, dan waktu yang memenuhi berbagai kendala. Berbagai studi melaporkan bahwa GA mampu mengurangi konflik jadwal dan waktu pembuatan jadwal secara drastis dibandingkan pendekatan tradisional. Di tingkat nasional, penerapan GA juga telah terbukti efektif dalam aplikasi penjadwalan mata kuliah, di mana sistem mampu menyelesaikan optimasi jadwal tanpa menghasilkan benturan antar kelas (Mone & Simarmata, 2021). Di sisi lain, pendekatan *Queue-Based Scheduling* yang memanfaatkan struktur antrian dan kebijakan prioritas banyak digunakan pada berbagai domain untuk mengatur urutan eksekusi tugas dengan mempertimbangkan ketersediaan sumber daya (Maspiyanti et al., 2025). Secara teknis, metode berbasis antrian ini sangat efisien dalam mengeksekusi penugasan secara sekuensial dan sering diadopsi sebagai baseline penjadwalan pada sistem yang membutuhkan respons komputasi instan (Thamizhmaran & Venkatasubramanian, 2022). Secara teknis, metode berbasis antrian ini sangat efisien dalam mengeksekusi penugasan secara sekuensial dan sering diadopsi sebagai baseline penjadwalan pada sistem yang membutuhkan respons komputasi instan (Thamizhmaran & Venkatasubramanian, 2022). Studi lain juga menunjukkan bahwa penerapan GA pada penjadwalan kuliah mampu menghasilkan jadwal yang bebas benturan dengan waktu komputasi yang masih efisien untuk skala institusi pendidikan tinggi (Mutrofin et al., 2021; Nasien & Andi, 2022).

Meskipun banyak penelitian membahas penerapan GA pada penjadwalan kuliah dan penerapan *Queue-Based Scheduling* pada konteks manajemen antrian, studi yang secara spesifik membandingkan kedua pendekatan tersebut sebagai algoritma *generate* jadwal kuliah pada konteks institusi tertentu dengan kendala lokal yang unik masih terbatas. Secara teknis, metode berbasis antrian ini sangat efisien dalam mengeksekusi penugasan secara sekuensial dan sering diadopsi sebagai baseline penjadwalan pada sistem yang membutuhkan respons komputasi instan (Juniar, 2015; Thamizhmaran & Venkatasubramanian, 2022). Sebelumnya, studi komparatif terkait optimasi penjadwalan lebih banyak membandingkan GA dengan algoritma pencarian lokal seperti *Steepest Ascent Hill Climbing*, yang menunjukkan bahwa GA cenderung lebih unggul dalam menghindari jebakan optimum lokal (Ardiyani, 2022). Berbagai studi melaporkan bahwa GA mampu mengurangi konflik jadwal dan waktu pembuatan jadwal secara drastis dibandingkan pendekatan tradisional (Paranduk et al., 2018; Sulehu & Rimalia, 2023). Di tingkat nasional, penerapan GA juga telah terbukti efektif dalam aplikasi penjadwalan mata kuliah, di mana sistem mampu menyelesaikan optimasi jadwal tanpa menghasilkan benturan antar kelas (Mone & Simarmata, 2021; Paranduk et al., 2018). Di UМУKA, kendala penjadwalan mencakup kapasitas maksimum ruang, batasan jam mengajar dosen, preferensi waktu mengajar tertentu, pembentukan kelas paralel, serta pengelompokan mata kuliah wajib dan pilihan pada slot tertentu. Selain itu, kebutuhan integrasi dengan arsitektur sistem modern berbasis web, *containerization* menggunakan *Docker*, dan *NoSQL* database menuntut rancangan algoritma yang tidak hanya akurat, tetapi juga efisien dan dapat diskalakan (Khalida et al., 2025).

Penelitian ini berangkat dari celah tersebut dengan mengembangkan sistem *generate* jadwal kuliah otomatis berbasis GA dan *Queue-Based Scheduling* yang diimplementasikan dalam lingkungan *container* dan *NoSQL* di UМУKA, serta melakukan perbandingan kinerja kedua algoritma dalam menyelesaikan masalah penjadwalan yang sama. Fokus penelitian bukan hanya pada perancangan model penjadwalan, tetapi juga pada pengukuran dan analisis perbedaan kinerja kedua algoritma dalam hal waktu komputasi, jumlah pelanggaran kendala, dan kualitas jadwal yang direpresentasikan oleh nilai fungsi *fitness* (Manuel & Mohan, 2025). Dengan demikian, penelitian ini diharapkan memberikan kontribusi praktis berupa rekomendasi algoritma yang paling sesuai untuk diterapkan di UМУKA, serta kontribusi ilmiah berupa studi komparatif yang memperkaya literatur penjadwalan kuliah berbasis optimasi.

Secara khusus, penelitian ini bertujuan merancang dan mengimplementasikan modul *generate* jadwal kuliah otomatis menggunakan GA dan *Queue-Based Scheduling* yang mengakomodasi kendala penjadwalan di UМУKA, melakukan pengujian kinerja kedua algoritma berdasarkan metrik waktu komputasi, jumlah pelanggaran kendala, dan nilai fungsi *fitness*, serta menganalisis efektivitas masing-masing algoritma untuk menentukan algoritma yang paling layak diadopsi pada sistem penjadwalan kuliah otomatis di UМУKA. Berdasarkan tujuan tersebut, kontribusi utama penelitian ini adalah menyediakan dasar empiris bagi pengambil keputusan di lingkungan perguruan tinggi dalam memilih algoritma *generate* jadwal kuliah yang sesuai dengan kebutuhan institusi dan karakteristik infrastrukturnya.

2. METODOLOGI PENELITIAN

2.1 Kerangka Dasar Penelitian

Penelitian ini termasuk dalam kategori penelitian pengembangan (research and development) yang berfokus pada pengembangan dan evaluasi sistem *generate* penjadwalan kuliah otomatis. Objek penelitian adalah proses penjadwalan kuliah reguler di Universitas Muhammadiyah Karanganyar dengan memperhatikan kendala-kendala yang berlaku di lingkungan tersebut. Penelitian menggunakan pendekatan kuantitatif melalui serangkaian eksperimen terkontrol untuk mengukur kinerja dua algoritma *generate*, yaitu *Genetic Algorithm* (GA) dan *Queue-Based Scheduling*.

Variabel bebas (independent variable) dalam penelitian ini adalah jenis algoritma *generate* yang digunakan, yaitu GA dan *Queue-Based Scheduling*. Variabel terikat (dependent variable) meliputi: (1) waktu komputasi proses *generate* jadwal, (2) jumlah pelanggaran kendala (conflict) pada jadwal yang dihasilkan, dan (3) nilai fungsi *fitness* yang

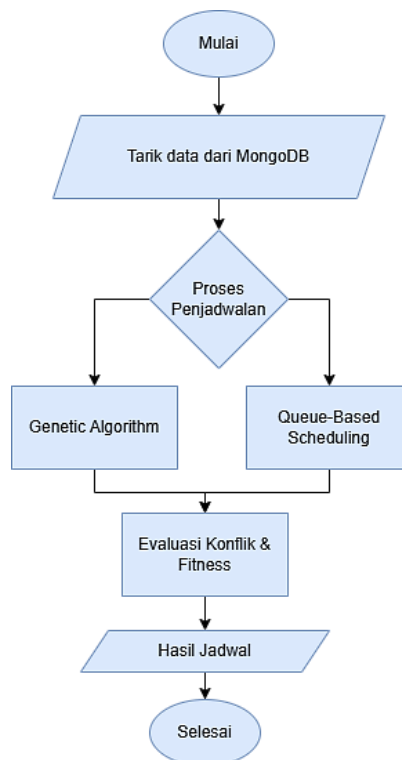
merepresentasikan kualitas jadwal secara keseluruhan. Kendala penjadwalan mencakup kapasitas maksimum ruang, tidak adanya konflik jadwal dosen, tidak adanya benturan jadwal kelas paralel, serta kepatuhan terhadap slot waktu yang tersedia.

Secara konseptual, kerangka pemikiran penelitian dapat digambarkan sebagai proses yang dimulai dari identifikasi kebutuhan penjadwalan dan kendala UMOKA, perancangan representasi kromosom dan fungsi fitness untuk GA, perancangan skema antrean dan aturan prioritas untuk *Queue-Based Scheduling*, implementasi kedua algoritma dalam sistem berbasis web yang berjalan di atas container Docker dan NoSQL database, hingga tahap eksperimen dan analisis perbandingan kinerja. Hipotesis utama penelitian adalah bahwa GA akan menunjukkan kinerja yang lebih baik dibandingkan *Queue-Based Scheduling*, baik dari sisi jumlah konflik maupun nilai fitness, dengan waktu komputasi yang masih dalam batas wajar untuk skala penjadwalan UMOKA.

2.2 Tahapan Penelitian

Tahapan penelitian dirancang secara sistematis agar proses pengembangan dan pengujian algoritma berjalan terstruktur. Tahap pertama adalah analisis kebutuhan yang meliputi pengumpulan informasi mengenai aturan penjadwalan di UMOKA, data mata kuliah, dosen, ruang, dan pola penjadwalan pada beberapa semester terakhir. Tahap kedua adalah perancangan model penjadwalan, termasuk pemodelan kendala, representasi jadwal sebagai solusi (kromosom untuk GA dan entri antrean untuk *Queue-Based Scheduling*), serta perancangan fungsi evaluasi.

Tahap ketiga adalah implementasi algoritma dalam bentuk modul generate di dalam sistem berbasis Laravel, yang dijalankan dalam lingkungan container Docker dan menggunakan NoSQL database sebagai media penyimpanan data jadwal. Penggunaan framework Laravel dipilih karena kemampuannya dalam mempercepat proses web-development dengan standar arsitektur MVC (Model-View-Controller) yang kokoh, serta terbukti andal dalam inovasi manajemen sistem informasi akademik kampus (Alwiyah et al., 2022; Subecz, 2021). Pemilihan MongoDB sebagai basis data NoSQL juga didasari oleh arsitektur penyimpanannya yang berbasis dokumen fleksibel, sehingga terbukti memberikan performa tinggi dan latensi rendah ketika mengeksekusi operasi baca-tulis data dalam jumlah masif (query performance), yang sangat krusial dalam iterasi algoritma penjadwalan (Achanta, 2024; Al Maamari & Nasar, 2025; Daly, 2021; Kausar et al., 2022; Ramadoni et al., 2021). Pada tahap ini, parameter-parameter GA seperti ukuran populasi, probabilitas crossover, dan probabilitas mutasi ditentukan berdasarkan studi literatur dan proses tuning awal, sedangkan untuk *Queue-Based Scheduling*, kebijakan prioritas dan strategi penanganan konflik didefinisikan secara eksplisit. Tahap keempat adalah pengujian dan eksperimen, di mana kedua algoritma dijalankan pada dataset simulasi dengan beberapa skenario beban penjadwalan yang berbeda, dan setiap skenario diulang beberapa kali untuk memperoleh nilai rata-rata kinerja.



Gambar 1. Arsitektur Alur Penjadwalan Sistem di UMOKA

Tahap terakhir adalah analisis data dan penarikan kesimpulan. Hasil pengukuran waktu komputasi, jumlah konflik, dan nilai fitness dari kedua algoritma dievaluasi menggunakan metode analisis komparatif untuk membandingkan efektivitas kinerja GA dan *Queue-Based Scheduling* dalam menyelesaikan masalah penjadwalan di



UMUKA. Hasil analisis ini kemudian diinterpretasikan dalam konteks kebutuhan UMUKA untuk merekomendasikan algoritma yang paling tepat diimplementasikan pada sistem penjadwalan kuliah otomatis.

2.3 Desain Eksperimen Perbandingan Algoritma

Desain eksperimen perbandingan algoritma dilakukan dengan menyusun beberapa skenario beban penjadwalan yang merepresentasikan kondisi nyata di UMUKA, misalnya skenario dengan jumlah mata kuliah sedang, tinggi, dan sangat tinggi, serta variasi jumlah kelas paralel. Untuk setiap skenario, kedua algoritma dijalankan dengan input data yang sama sehingga perbandingan kinerja dapat dilakukan secara adil.

Untuk GA, setiap skenario dijalankan dengan jumlah generasi maksimum tertentu (misalnya N generasi) dan ukuran populasi M individu. Pada setiap generasi, nilai fitness terbaik dan rata-rata dicatat, serta jumlah pelanggaran kendala pada solusi terbaik. Untuk *Queue-Based Scheduling*, algoritma dijalankan satu kali per skenario dengan kebijakan prioritas yang sama, dan hasil jadwal yang dihasilkan dievaluasi berdasarkan jumlah konflik dan nilai fungsi evaluasi yang setara dengan fitness pada GA.

Metode pengukuran waktu komputasi dilakukan dengan mencatat durasi eksekusi proses generate dari awal hingga diperoleh jadwal akhir untuk masing-masing algoritma. Seluruh eksperimen dijalankan pada lingkungan perangkat keras dan konfigurasi sistem yang sama untuk menghindari bias performa. Data hasil eksperimen kemudian direkap dalam bentuk tabel dan grafik untuk dianalisis lebih lanjut.

3. HASIL DAN PEMBAHASAN

3.1 Implementasi Metode

Sebelum menjabarkan hasil komputasi, tahapan esensial dalam penelitian ini adalah menerjemahkan aturan akademik ke dalam model komputasi yang representatif. Sistem penjadwalan ini mengimplementasikan dua pendekatan algoritma yang berbeda secara fundamental, yakni Genetic Algorithm (GA) sebagai metode heuristik adaptif dan Queue-Based Scheduling (QBS) sebagai metode sekuensial deterministik sebagai dasar perbandingan. Keputusan untuk mengadopsi GA didasarkan pada kemampuannya untuk mengeksplorasi ruang solusi yang luas dan menemukan solusi optimal atau mendekati optimal dalam masalah penjadwalan yang kompleks dan seringkali memiliki banyak batasan. Sementara itu, QBS dipilih sebagai *baseline* karena kesederhanaan implementasinya dan kemampuannya dalam menghasilkan jadwal secara cepat, meskipun potensi optimalitasnya terbatas.

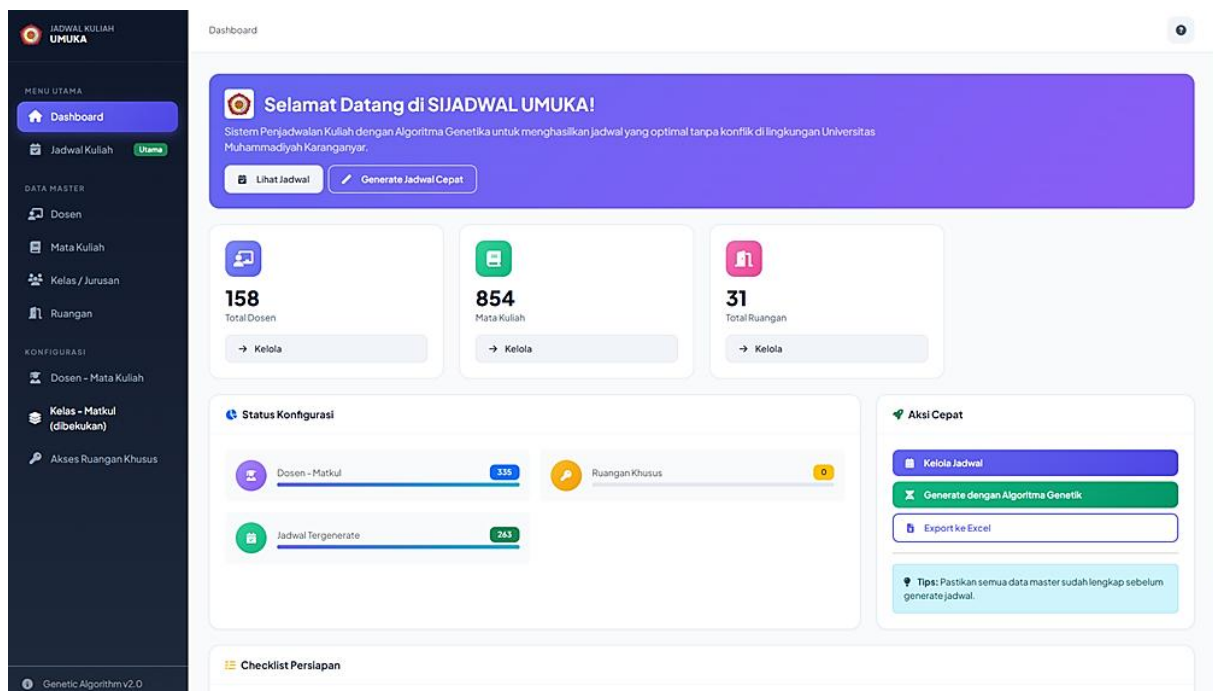
Pada implementasi Genetic Algorithm (GA), jadwal direpresentasikan secara inovatif sebagai struktur kromosom, di mana setiap kromosom merupakan kumpulan gen yang mengkodekan informasi krusial untuk setiap sesi perkuliahan: Mata Kuliah, Dosen yang ditugaskan, Kelas atau Ruangan yang dialokasikan, serta Waktu penyelenggaraan. Kualitas setiap kromosom (jadwal) dievaluasi secara kuantitatif menggunakan fungsi *Fitness* yang dirancang secara cermat. Nilai *fitness* ini merefleksikan seberapa baik jadwal tersebut memenuhi batasan-batasan yang ditetapkan. Evaluasi ini mencakup penalti yang signifikan untuk pelanggaran batasan keras (*hard constraints*), seperti konflik penugasan dosen pada waktu yang bersamaan atau ketidaksesuaian kapasitas ruang dengan jumlah mahasiswa. Selain itu, *fitness function* juga mempertimbangkan penalti yang lebih ringan untuk pelanggaran batasan lunak (*soft constraints*), yang bertujuan untuk meningkatkan efisiensi dan kenyamanan, misalnya meminimalkan jeda mengajar yang terlalu lama bagi dosen. Proses evolusi algoritma berlangsung melalui siklus iteratif yang terdiri dari mekanisme *Selection* (menggunakan metode Roulette Wheel untuk probabilitas pemilihan individu), *Crossover* (pertukaran informasi genetik antar kromosom), dan *Mutation* (perubahan acak pada gen) hingga tercapai kondisi terminasi yang telah ditentukan, baik berupa batas maksimum generasi maupun ditemukannya jadwal yang sepenuhnya bebas dari bentrokan (*conflict-free*).

Sementara itu, pada implementasi Queue-Based Scheduling (QBS), algoritma dibangun berdasarkan logika sederhana First-Come-First-Served (FCFS) yang diterapkan pada daftar mata kuliah. Daftar mata kuliah yang telah diurutkan (misalnya berdasarkan prioritas atau urutan pendaftaran) dimasukkan ke dalam sebuah antrean (*queue*). Sistem kemudian secara iteratif melakukan pencarian ruang dan waktu kosong pertama yang tersedia dalam basis data yang terstruktur. Setelah ditemukan kombinasi ruang dan waktu yang cocok, mata kuliah tersebut langsung ditempatkan tanpa melalui proses evaluasi mendalam terhadap *soft constraints* atau pertimbangan kenyamanan jadwal secara holistik. Pendekatan ini menjamin bahwa setiap mata kuliah akan mendapatkan alokasi, namun tidak memberikan jaminan optimalitas dalam hal pemenuhan kriteria-kriteria tambahan yang seringkali krusial dalam konteks akademik, seperti keseimbangan beban mengajar dosen atau efisiensi penggunaan fasilitas.

3.2 Implementasi Antarmuka Sistem

Prototipe penjadwalan kuliah otomatis, yang dinamakan SIJADWAL UMUKA, berhasil dibangun dan diimplementasikan dengan memanfaatkan antarmuka berbasis web responsif. Pendekatan ini secara strategis dirancang untuk mengabstraksi kompleksitas inheren dari perhitungan algoritma optimasi seperti Genetic Algorithm (GA) dan metode Queue-Based Scheduling, sehingga menyajikan solusi yang lebih mudah diakses oleh pengguna. Pada halaman awal, pengguna disambut oleh sebuah dasbor utama yang terintegrasi secara informatif, merangkum keseluruhan data master yang krusial untuk proses penjadwalan. Seperti yang dapat diamati pada Gambar 2, sistem telah berhasil mengagregasi dan memvalidasi data awal yang meliputi 158 Dosen, 854 Mata Kuliah, dan 31 Ruangan, yang semuanya

siap untuk diproses lebih lanjut. Selain itu, tampilan dasbor ini juga secara ringkas menampilkan keluaran awal dari sistem, yaitu ringkasan 263 jadwal perkuliahan yang telah berhasil di-*generate* oleh algoritma, memberikan gambaran kuantitatif awal mengenai kapabilitas sistem dalam menyelesaikan permasalahan penjadwalan yang kompleks.



Gambar 2. Antarmuka Dasbor Utama

Alih-alih menampilkan seluruh form pengisian data secara terpisah, fungsionalitas Manajemen Dosen, Mata Kuliah, dan Ruangan diintegrasikan ke dalam modul Master Data. Integrasi ini merupakan temuan kunci yang menunjukkan pergeseran dari pendekatan modular yang terfragmentasi menuju platform manajemen data yang terpusat dan holistik dalam sistem penjadwalan akademik. Pada modul ini, sistem memisahkan klasifikasi parameter, seperti Ruang Umum dan Ruang Khusus (Laboratorium), serta mengakomodasi beban SKS dan NIP dosen. Pemisahan dan klasifikasi parameter ini bukan sekadar kategorisasi, melainkan fondasi logis yang memungkinkan pemodelan kendala yang kompleks untuk algoritma penjadwalan.

Relasi antar parameter ini, seperti ketersediaan ruang khusus untuk mata kuliah praktikum atau kesesuaian beban SKS dosen dengan jadwal yang diusulkan, menjadi variabel krusial (*hard constraint*) yang esensial bagi kinerja algoritma. Konsep *hard constraint* dalam penjadwalan merujuk pada aturan yang mutlak harus dipenuhi agar jadwal yang dihasilkan valid dan dapat dieksekusi. Sebagai contoh, algoritma dirancang untuk secara inheren mencegah alokasi kelas praktikum ke ruang teori reguler, atau memastikan bahwa seorang dosen tidak dijadwalkan mengajar di dua tempat berbeda pada waktu yang bersamaan. Pendekatan ini secara signifikan mengurangi potensi kesalahan manual yang sering terjadi pada sistem penjadwalan tradisional yang terpisah. Melalui antarmuka yang terintegrasi ini, administrator tidak hanya dapat mengeksekusi proses penjadwalan secara otomatis (*generate*), tetapi juga memperoleh visibilitas yang superior terhadap jadwal akhir yang telah bebas konflik. Fitur tambahan berupa peta persebaran waktu mengajar dosen secara visual memberikan dimensi pemahaman yang lebih mendalam mengenai beban kerja dan alokasi sumber daya dosen. Kemampuan untuk melihat secara visual distribusi jam mengajar dosen memungkinkan administrator untuk mengidentifikasi potensi kelebihan beban atau kesenjangan dalam utilisasi dosen secara proaktif, yang sebelumnya sulit dilakukan dengan metode manual atau sistem yang terfragmentasi.

Temuan ini mengindikasikan bahwa integrasi data master dan visualisasi informasi, sebagaimana diimplementasikan dalam modul Master Data, berkontribusi pada peningkatan efisiensi dan efektivitas dalam proses penjadwalan akademik. Kemampuan sistem untuk secara cerdas mengelola *hard constraints* berdasarkan relasi parameter yang terdefinisi dengan baik, serta menyajikan hasil dalam format yang mudah diinterpretasikan, menandakan kemajuan signifikan dibandingkan metode konvensional. Hal ini sejalan dengan penelitian sebelumnya yang menekankan pentingnya basis data yang terintegrasi dan antarmuka pengguna yang intuitif untuk mendukung pengambilan keputusan yang kompleks dalam manajemen akademik. Implementasi ini tidak hanya menyederhanakan tugas administratif tetapi juga berkontribusi pada optimalisasi alokasi sumber daya akademik dan peningkatan kepuasan dosen serta mahasiswa melalui jadwal yang lebih terstruktur dan bebas masalah.

3.3 Hasil Pengujian Genetic Algorithm

engujian kinerja dilakukan menggunakan skenario peningkatan beban mata kuliah (dari 20 hingga 263 MK) untuk mengukur skalabilitas dan ketahanan kedua algoritma.



Mekanisme evolusioner pada GA diterapkan untuk melakukan pencarian solusi global (global search) agar persebaran jadwal dosen dan ketersediaan ruang menjadi optimal. Hasil dari enam skenario pengujian GA disajikan pada Tabel 1.

Tabel 1. Hasil Pengujian *Genetic Algorithm* (GA)

Skenario	Jumlah MK	Konflik Akhir	Nilai <i>Fitness</i> Terbaik	Waktu Komputasi (detik)
1	20	0	0	2,0988
2	50	0	1	4,8449
3	100	0	10,4	11,6774
4	150	6	77,2	19,9254
5	200	14	171,6	27,8998
6	263	41	462,20	36,2066

Berdasarkan analisis data yang disajikan pada Tabel 1, Algoritma Genetika (GA) menunjukkan kapabilitas yang signifikan dalam menuntaskan masalah penjadwalan tanpa adanya pelanggaran konflik (nilai 0 konflik) pada skenario beban kerja ringan hingga menengah. Skenario 1 hingga 3, yang merepresentasikan kondisi beban ringan (misalnya, jumlah mata kuliah (MK) yang sedikit, ketersediaan ruang yang melimpah), berhasil diselesaikan secara absolut oleh GA, mengindikasikan bahwa parameter dan operator genetika yang diterapkan efektif dalam menemukan solusi optimal dalam ruang pencarian yang relatif tidak kompleks. Namun, seiring dengan peningkatan beban kerja secara masif, khususnya pada skenario 4 hingga 6 yang mencakup 150 hingga 263 mata kuliah, algoritma mulai mendeteksi adanya benturan, yang termanifestasi dalam bentuk konflik matriks ruang dan waktu (dengan kisaran 6 hingga 41 konflik). Fenomena ini perlu dipahami bukan sebagai indikasi kegagalan mutlak dari algoritma dalam menemukan solusi, melainkan sebagai cerminan inheren dari sifat heuristik GA yang beroperasi dengan prinsip optimasi multi-objektif. GA secara aktif berusaha menjaga keseimbangan antara kualitas populasi jadwal secara keseluruhan dan pemenuhan setiap kendala, yang dalam kasus beban tinggi, seringkali memerlukan kompromi.

Lebih lanjut, peningkatan jumlah konflik pada beban kerja yang lebih berat tidak serta-merta menurunkan kualitas solusi secara keseluruhan, sebagaimana dibuktikan oleh metrik *Fitness* Terbaik yang mengalami peningkatan eksponensial yang dramatis. Data menunjukkan lonjakan signifikan dari nilai nominal 1,00 (kemungkinan merepresentasikan kondisi awal atau standar referensi) hingga mencapai 462,20 pada skala maksimum. Hal ini mengindikasikan bahwa, meskipun terjadi pelanggaran lokal (konflik), GA terus menerus melakukan eksplorasi dan eksploitasi ruang pencarian melalui mekanisme mutasi dan crossover untuk mengidentifikasi solusi yang secara global memberikan nilai "kebugaran" tertinggi. GA secara proaktif berupaya mengisolasi dan mengelola sisa kendala yang muncul akibat kepadatan ruang dan waktu yang tinggi. Mekanisme mutasi, dalam konteks ini, berfungsi untuk memperkenalkan variasi genetik yang dapat membantu keluar dari *local optima* dan mengeksplorasi area solusi yang belum terjamah, memastikan bahwa sebagian besar jadwal dalam populasi yang dihasilkan tetap berada pada tingkat optimasi yang tinggi meskipun terdapat *bottleneck* yang tak terhindarkan.

Analisis mendalam terhadap temuan ini selaras dengan literatur terkini yang membahas keterbatasan algoritma heuristik dalam menyelesaikan masalah optimasi kompleks dengan kendala ketat dan saling terkait. Teori mengenai trade-off antara kecepatan konvergensi dan kualitas solusi pada algoritma metaheuristik, seperti yang dibahas dalam review algoritma metaheuristik klasik dan adaptif (Ge et al., 2025), menjelaskan bahwa intensifikasi pencarian pada solusi yang sudah cukup baik dapat mengarah pada penemuan konflik dalam kondisi beban ekstrem. Kondisi "bottleneck" akibat rasio kepadatan ruangan dan waktu yang tinggi merupakan tantangan klasik dalam penjadwalan, di mana ruang solusi layak (*feasible region*) menjadi sangat terbatas, sebagaimana ditunjukkan dalam studi genetic algorithm untuk scheduling dengan kendala resource dinamis. GA, melalui desainnya, berusaha mencari solusi paling mendekati optimal dalam ruang terbatas tersebut, bahkan jika mengkompromikan pemenuhan absolut batasan untuk *fitness* keseluruhan lebih tinggi. Oleh karena itu, deteksi konflik pada beban kerja masif bukan indikasi kegagalan algoritma, melainkan manifestasi perjuangannya mencari keseimbangan terbaik di bawah tekanan kendala signifikan, sesuai prinsip optimasi multi-objektif dalam metaheuristik modern.

3.4 Hasil Pengujian Queue-Based Scheduling

Sebagai algoritma pembanding berjenis deterministik sekuensial, *Queue-Based Scheduling* (QBS) diuji dengan urutan prioritas antrean berdasarkan pembacaan data awal dan ketersediaan slot. Hasil evaluasinya ditampilkan pada Tabel 2.

Tabel 2. Hasil Pengujian *Queue-Based Scheduling* (QBS)

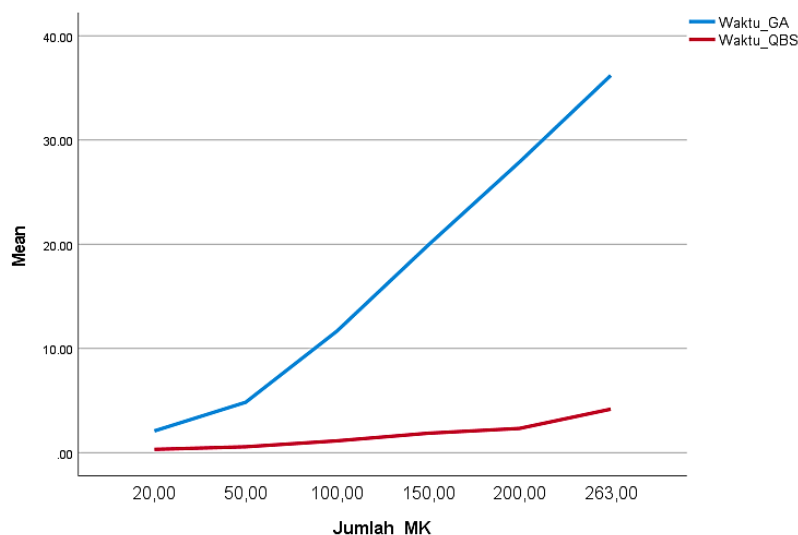
Skenario	Jumlah MK	Konflik Akhir	<i>Fitness</i> Terbaik	Waktu Komputasi (detik)
1	20	0	0	0,3221
2	50	0	0	0,5729
3	100	0	0	1,1407
4	150	0	0	1,8711
5	200	0	0	2,3382
6	263	0	0	4,1804

Tabel 2 sekilas memang menampilkan pencapaian "0 konflik" dengan waktu komputasi yang sangat impresif, di mana rata-rata durasi pemrosesan untuk 263 Mata Kuliah (MK) hanya berkisar maksimal 4,18 detik. Namun, dalam konteks analisis hasil dan pembahasan pada Bab 4 ini, pandangan awal tersebut memerlukan interpretasi yang jauh lebih kritis dan mendalam. Ketidadaan konflik yang teramati pada Tabel 2, jika hanya dilihat secara permukaan, berpotensi menyesatkan secara akademik karena sifatnya yang semu dan prematur. Hal ini sangat mungkin terjadi mengingat karakteristik fundamental dari algoritma QBS (Queue-Based Scheduling) yang digunakan, yaitu berbasis prinsip *First-Come-First-Served* (FCFS). Pendekatan FCFS secara inheren menginstruksikan agar jadwal langsung diinjeksikan ke dalam slot waktu kosong pertama yang tersedia tanpa melakukan analisis silang (*cross-validation*) terhadap konsekuensi jangka panjangnya terhadap parameter-parameter penjadwalan lainnya, terutama *soft constraints*.

Lebih lanjut, investigasi mendalam terhadap proses penjadwalan yang dihasilkan oleh algoritma QBS mengungkapkan bahwa parameter *soft constraints*, yang krusial untuk optimalisasi kualitas penjadwalan seperti keseimbangan beban dosen, alokasi ruang yang efisien, dan distribusi kelas mahasiswa yang merata, secara fundamental diabaikan. Konfirmasi empiris atas kelalaian ini terlihat jelas dari Nilai Fitness yang secara konsisten mutlak berada di angka 0. Dalam teori penjadwalan, Nilai Fitness yang nol mengindikasikan bahwa solusi yang dihasilkan tidak memenuhi kriteria optimasi apa pun yang telah ditetapkan, bahkan jika kriteria tersebut bersifat lunak. Dengan kata lain, meskipun algoritma berhasil menghindari tabrakan jadwal (konflik keras), ia gagal mencapai tujuan penjadwalan yang lebih holistik dan meminimalkan ketidaknyamanan atau inefisiensi (konflik lunak). Keadaan ini, meskipun secara teknis tidak menimbulkan bentrokan kelas, menimbulkan konsekuensi praktis yang signifikan.

Implikasi praktis dari dominasi hard constraints atas soft constraints pada algoritma QBS ini terwujud dalam pola luaran jadwal yang cenderung menumpuk secara mekanis di awal minggu perkuliahan. Distribusi kelas yang tidak merata ini secara langsung berkontribusi pada peningkatan idle time yang signifikan bagi dosen, di mana terdapat jeda waktu panjang antara sesi mengajar mereka, yang berpotensi mengurangi produktivitas dan efisiensi waktu kerja. Selain itu, penumpukan kelas-kelas berat atau mata kuliah yang menuntut konsentrasi tinggi bagi mahasiswa dalam satu atau dua hari di awal minggu dapat memicu kelelahan kognitif (*cognitive fatigue*), menurunkan efektivitas pembelajaran, dan menciptakan pengalaman perkuliahan yang kurang optimal secara keseluruhan. Fenomena ini menggarisbawahi kebutuhan mendesak untuk tidak hanya mengejar efisiensi komputasi dalam algoritma penjadwalan, tetapi juga memberikan perhatian setara pada kualitas dan keberlanjutan jadwal yang dihasilkan, selaras dengan prinsip-prinsip optimalisasi penjadwalan yang dikemukakan oleh studi terkini tentang penanganan constraint dan optimasi jadwal perkuliahan (Fatchurrochman et al., 2023).

3.4 Perbandingan Kinerja GA dan Queue-Based Scheduling



Gambar 9. Grafik Perbandingan Waktu Komputasi GA vs Queue-Based Scheduling

Integrasi data pengujian dari kedua algoritma memberikan landasan evaluasi yang jelas. *Queue-Based Scheduling* unggul signifikan dari segi metrik waktu karena tidak memerlukan manipulasi kromosom dan evaluasi rekursif. Metode sekuensial ini cocok jika institusi hanya sekadar ingin "memasukkan jadwal ke kotak kosong" tanpa memedulikan rasionalitas akademik operasional.

Sebaliknya, *Genetic Algorithm* membuktikan supremasinya sebagai algoritma cerdas yang berorientasi pada kualitas. Kendati waktu komputasi pada skenario puncak memakan waktu 36,2 detik, durasi ini dinilai amat sangat efisien untuk sebuah otomasi sistem tingkat fakultas, yang mana bila dilakukan secara manual (konvensional) dapat menghabiskan waktu berhari-hari. Pertambahan waktu komputasi pada GA merupakan investasi yang wajar untuk mendapatkan jadwal dengan nilai fitness 462,20. Implikasinya bagi pihak program studi sangat krusial: GA menjamin distribusi beban kelas yang proporsional antar-hari, menjembatani irisan waktu istirahat (jeda sholat/makan) yang manusiawi, dan mencegah jadwal berlubang yang merugikan produktivitas dosen. Keberadaan 41 konflik minor murni



disebabkan oleh batasan fisik ruang kelas UMUKA terhadap jumlah SKS yang diproses, yang penyelesaiannya dapat difasilitasi melalui revisi intervensi admin setelah proses *generate*. Oleh karena itu, penelitian ini menyimpulkan bahwa implementasi algoritma GA terbukti lebih aplikatif, berkualitas tinggi, dan relevan dengan kebutuhan manajerial akademik di institusi tersebut dibandingkan metode sekuensial QBS.

4. KESIMPULAN

Penelitian ini berhasil menjawab permasalahan optimasi penjadwalan akademik di Universitas Muhammadiyah Karanganyar (UMUKA) melalui implementasi dan komparasi dua pendekatan algoritma, yakni *Genetic Algorithm* (GA) dan *Queue-Based Scheduling* (QBS). Berdasarkan pengujian pada enam skenario beban mata kuliah riil, QBS terbukti unggul secara absolut dalam hal kecepatan komputasi, namun gagal memenuhi standar kualitas operasional perguruan tinggi karena tidak melakukan optimasi *soft constraints* (nilai fitness statis di angka 0), yang berakibat pada penumpukan jadwal secara tidak proporsional. Sebaliknya, GA terbukti menjadi solusi yang paling efektif dan rasional untuk sistem penjadwalan otomatis; algoritma ini mampu menihilkan konflik (*hard constraints*) pada beban skala ringan hingga menengah, serta memberikan optimasi persebaran jadwal yang sangat baik (lonjakan nilai fitness yang tinggi) pada skala beban maksimal (263 mata kuliah) dengan waktu komputasi yang masih sangat efisien untuk level fakultas. Oleh karena itu, penelitian ini merekomendasikan GA sebagai mesin utama (*core engine*) penjadwalan otomatis di UMUKA untuk menggantikan metode manual, sementara QBS cukup difungsikan sebagai metode alternatif untuk pengujian basis data terdistribusi secara cepat. Keterbatasan penelitian ini terletak pada penggunaan dataset tunggal institusi dan belum maksimalnya eksplorasi mutasi parameter GA. Untuk pengembangan selanjutnya, disarankan melakukan pengujian kombinasi algoritma *hybrid* serta mengukur performanya pada perguruan tinggi lain yang memiliki karakteristik keterbatasan ruang yang berbeda.

REFERENCES

- Achanta, P. R. D. (2024). Optimizing Hybrid Cloud Database Architecture: Integrating SQL Server and MongoDB in Azure Environments. *International Journal of Scientific Research and Management (IJSRM)*, 12(12), 1815–1826. <https://doi.org/10.18535/ijssrm/v12i12.ec05>
- Al Maamari, S. R. S., & Nasar, M. (2025). A Comparative Analysis of NoSQL and SQL Databases: Performance, Consistency, and Suitability for Modern Applications with a Focus on IoT. *East Journal of Computer Science*, 1(2), 10–15. <https://doi.org/10.63496/ejcs.vol1.iss2.76>
- Al-Fitouri, A. M., Agaal, A., Ali, M., & Bugrin, H. A. (2025). A Genetic Algorithm Approach for University Course Timetabling: A Case Study at the Faculty of IT, Sebha University. *Sebha University Conference Proceedings*, 04(3). <https://doi.org/10.51984/SUCP.V4I3.4003>
- Alwiyah, A., Sayyida, S., Sunarya, P. A., & Apriliasari, D. (2022). Inovasi Manajemen Pengajaran Judul Kuliah Kerja Praktek (KKP) Berbasis Laravel Framework. *Technomedia Journal*, 7(2), 168–180. <https://doi.org/10.33050/tmj.v7i2.1902>
- Apriliyanto, E., Iriandi Putra, R., & Kurniawan, W. (2026). PIMAS Pemberdayaan Masyarakat melalui Implementasi Sistem Pembayaran QRIS Berbasis Biometrik dan AI-IoT di Kabupaten Sukoharjo. *Jurnal Pengabdian Masyarakat (PIMAS)*, 5(1). <https://doi.org/10.35960/pimas.v1i2.2161>
- Ardiyani, L. P. S. (2022). Perbandingan Algoritma Genetika dengan Algoritma Steepest Ascent Hill Climbing untuk Optimasi Penjadwalan Kuliah. *Jurnal Nasional Pendidikan Teknik Informatika (JANAPATI)*, 11(1), 63. <https://doi.org/10.23887/janapati.v11i1.43172>
- Arratia-Martinez, N. M., Maya-Padron, C., & Avila-Torres, P. A. (2021). University Course Timetabling Problem with Professor Assignment. *Mathematical Problems in Engineering*, 2021. <https://doi.org/10.1155/2021/6617177>
- Chen, M. C., Sze, S. N., Goh, S. L., Sabar, N. R., & Kendall, G. (2021). A Survey of University Course Timetabling Problem: Perspectives, Trends and Opportunities. *IEEE Access*, 9, 106515–106529. <https://doi.org/10.1109/ACCESS.2021.3100613>
- Daly, D. (2021). *Performance Engineering and Database Development at MongoDB*. 129–129. <https://doi.org/10.1145/3447545.3451199>
- Fatchurrochman, F., Afandi, A. N., Arifin, M. Z., & Mahmudy, W. F. (2023). Algoritma Penanganan Constraint pada Persoalan Penjadwalan Perkuliahan Universitas di Lingkungan Pendidikan Tinggi Keagamaan Islam (PTKI). *Jurnal Edukasi Dan Penelitian Informatika (JEPIN)*, 9(2), 331. <https://doi.org/10.26418/jp.v9i2.64546>
- Ge, Z., Ji, C., & Zhang, B. (2025). A Review of Classic Metaheuristic Optimization Algorithms. *Proceedings of the 2025 6th International Conference on Computer Information and Big Data Applications*, 303–311. <https://doi.org/10.1145/3746709.3746761>
- Juniar, A. (2015). Penerapan Algoritma Greedy pada Penjadwalan Produksi Single-Stage dengan Parallel Machine di Industri Konveksi. *Jurnal Sifo Mikroskil*, 16(2), 1–5.
- Kausar, M. A., Nasar, M., & Soosaimanickam, A. (2022). A Study of Performance and Comparison of NoSQL Databases: MongoDB, Cassandra, and Redis Using YCSB. *Indian Journal Of Science And Technology*, 15(31), 1532–1540. <https://doi.org/10.17485/IJST/v15i31.1352>



- Khalida, R., Ondo, S., Tua, P., Budi, D., Sulistiowati, S., & Setiawati, S. (2025). Implementation of Genetic Algorithm for Automatic Course Scheduling Optimization. *International Journal of Knowledin Databases*, 05(02), 218–231. <https://doi.org/10.30983/knowbase.v5i2.10260>
- Mahlous, A. R., & Mahlous, H. (2023). Student timetabling genetic algorithm accounting for student preferences. *PeerJ Computer Science*, 9. <https://doi.org/10.7717/peerj-cs.1200>
- Manuel, B., & Mohan, V. (2025). *Timetable Generator Using Genetic Algorithm and Constraint Satisfaction Problem*. www.ijfmr.com
- Maspiyanti, F., Gatc, J., Rezeki, S., Nursari, C., & Murtako, A. (2025). *Course Timetabling using Genetic Algorithm and Fuzzy Cross-Over*. JOIV : International Journal on Informatics Visualization, 9(5), doi: 10.62527/joiv.9.5.3196
- Mone, F., & Simarmata, J. E. (2021). Application of Genetic Algorithm in Scheduling Subjects. *Barekeng*, 15(4), 615–628. <https://doi.org/10.30598/barekengvol15iss4pp615-628>
- Mutrofin, S., Zulfa, I., & Satyareni, D. H. (2021). Implementasi Algoritma Genetika untuk Otomatisasi Sistem Penjadwalan pada Lembaga Bimbingan Belajar. *G-Tech*, 5(1), 2021. doi: 10.33379/gtech.v5i1.1178
- Nasien, D., & Andi, A. (2022). Optimization of Genetic Algorithm in Courses Scheduling. *IT Journal Research and Development*, 151–161. <https://doi.org/10.25299/itjrd.2022.7896>
- Paranduk, L., Indriani, A., Hafid, M., & Sistem Informasi STMIK PPKIA Tarakanita Rahmawati Tarakan, J. (2018). Sistem Informasi Penjadwalan Mata Kuliah Menggunakan Algoritma Genetika Berbasis Web. In *Seminar Nasional Aplikasi Teknologi Informasi (SNATI)*.
- Ramadoni, Mahmud Zunos Amirudin, Rifki Fahmi, Ema Utami, & Muhammad Syukri Mustafa. (2021). *Evaluasi Penggunaan Prometheus dan Grafana Untuk Monitoring Database MongoDB*.
- Subecz, Z. (2021). Web-development with Laravel framework. *Gradus*, 8(1), 211–218. <https://doi.org/10.47833/2021.1.csc.006>
- Sulehu, M., & Rimalia, W. (2023). Penerapan Algoritma Genetika serta Metode TOPSIS Sebagai Solusi Penjadwalan Mata Kuliah. In *Journal of System and Computer Engineering (JSCE)*, 4(1).
- Supriyanti, W., Putri Utami, R., Tri Romadloni, N., & Sri Rahayu, Y. (2023). Rancang Bangun Sistem Informasi Penerimaan Peserta Didik Baru Di Smk Bhineka Karya Surakarta. *Jurnal Responsif*, 5(2).
- Thamizhmaran, K., & Venkatasubramanian, P. (2022). A Review of Weighted Fair Queuing based Scheduling Algorithm in Wireless Network. In *Journal of Advancement in Electronics Design*, 5(1)
- Trisanti, N., Pramuja, G., Fanani, I., Romadloni, N. T., Efendi, B., & Setiani, H. (2025). Cerdas Digital dengan Artificial Intelligence: Solusi Teknologi untuk Pelayanan dan Keamanan Publik Article history. *Cahaya Pengabdian*, 2(1), 2025. doi: 10.61971/cp.v2i1.207