

Implementasi Algoritma Genetika untuk Otomatisasi Penjadwalan Akademik Guna Menghindari *Conflict of Schedule*

Hasirun^{1*}, Prema Adhitya Dharma Kusumah², M. Alfa Rizy³, Ari Budi Riyanto⁴

^{1,4}Fakultas Sains dan Teknik, Universitas, Purbalingga, Jawa Tengah, Indonesia

²Fakultas Teknologi dan Informasi, Universitas Catur Insan Cendekia, Cirebon, Jawa Barat, Indonesia

³Fakultas Ilmu Komputer, Universitas AMIKOM Yogyakarta, Sleman, Indonesia

E-mail: ¹hasirun@unperba.ac.id, ²prema.adk@gmail.com, ³rizymuhammad@gmail.com,

⁴aribudiriyanto@unperba.ac.id

*Penulis Korespondensi

ABSTRAK

Penyusunan jadwal kuliah di perguruan tinggi merupakan masalah optimasi kombinatorial kompleks yang sering memicu konflik ruang dan waktu. Penelitian ini bertujuan mengembangkan sistem penjadwalan akademik berbasis *website* menggunakan Algoritma Genetika dengan metode *Roulette Wheel Selection*. Pengujian dilakukan secara riil pada Fakultas Sains dan Teknik Universitas Perwira Purbalingga dengan skala objek uji mencakup 20 dosen, 45 mata kuliah, dan 8 ruangan. Hasil eksperimen menunjukkan bahwa kombinasi parameter ukuran populasi 100, *crossover rate* 0.8, dan *mutation rate* 0.05 menghasilkan performa paling optimal. Konfigurasi ini mendorong konvergensi sempurna dengan nilai *best fitness* mutlak 1.0 pada generasi ke-210 dengan waktu komputasi 10 menit 15 detik. Sistem ini berhasil memangkas waktu kerja manual dari 5–7 hari menjadi kurang dari 15 menit, menghasilkan efisiensi operasional sebesar 97% dengan status *zero conflict*. Selain mematuhi seluruh batasan mutlak, algoritma ini mampu mengakomodasi 90% preferensi waktu mengajar dosen. Penelitian ini memberikan kontribusi teoretis dalam pengembangan model penjadwalan berbasis komputasi cerdas yang adaptif untuk fakultas dengan karakteristik praktikum intensif melalui integrasi aturan spesialisasi laboratorium yang ketat.

Article Info

Kata Kunci:

Algoritma genetika
Penjadwalan akademik
Roulette wheel
Berbasis web
Komputasi cerdas

Riwayat artikel:

Submit 14 Juli 2026
Revisi 20 Juli 2026
Diterima 22 Juli 2026

1. PENDAHULUAN

Penjadwalan akademik merupakan instrumen krusial dalam manajemen institusi pendidikan yang berfungsi untuk mengalokasikan sumber daya terbatas seperti dosen, ruang kelas, dan waktu secara optimal [1]. Namun, proses penyusunan ini sering kali dihadapkan pada masalah utama berupa terjadinya bentrok jadwal (*conflict of schedule*). Secara komputasi, kompleksitas penjadwalan akademik dikategorikan sebagai masalah *NP-Hard* (*Non-deterministic Polynomial-time Hard*), di mana ruang pencarian solusi berkembang secara eksponensial seiring bertambahnya variabel yang terlibat [2]. Masalah ini semakin diperumit oleh adanya batasan dinamis (*constraints*) yang saling mengunci, sehingga pendekatan konvensional atau manual sering kali gagal mencapai titik optimal dalam pemanfaatan ruang dan waktu, yang pada akhirnya mengganggu efektivitas proses pembelajaran [3][4][5].

Kondisi tersebut dialami secara langsung oleh Fakultas Sains dan Teknik (FST) di Universitas Perwira Purbalingga (UNPERBA). Sebagai fakultas dengan karakteristik rumpun saintek, FST memiliki ketergantungan yang sangat tinggi terhadap sinkronisasi yang ketat antara perkuliahan teori dan penggunaan laboratorium praktikum secara intensif [6]. Tantangan ini diperberat oleh jumlah dosen tetap yang terbatas, yang di sisi lain harus mengampu berbagai mata kuliah spesifik. Akibatnya, risiko terjadinya tumpang tindih jadwal mengajar dosen maupun penggunaan fasilitas ruangan antar program studi menjadi sangat tinggi [7]. Mekanisme penjadwalan manual yang

selama ini diterapkan terbukti tidak efisien, membutuhkan waktu pengerjaan yang lama, serta rentan terhadap kesalahan manusia (*human error*) yang menghambat profesionalisme layanan akademik di tingkat fakultas.

Untuk mengatasi kompleksitas tersebut, diperlukan pendekatan komputasi cerdas yang mampu mengeksplorasi ribuan kombinasi jadwal dalam waktu singkat, salah satunya melalui implementasi Algoritma Genetika [8]. Algoritma Genetika merupakan metode heuristik yang meniru prinsip seleksi alam dan genetika biologis, yang meliputi siklus evolusi terstruktur melalui proses inisialisasi populasi, evaluasi nilai *fitness*, seleksi individu, pindah silang (*crossover*), dan mutasi gen [9]. Keunggulan utama algoritma ini terletak pada kemampuannya untuk melakukan pencarian solusi optimal atau mendekati optimal secara global (*global optima*) pada ruang pencarian yang luas, serta menghindari kebuntuan komputasi pada solusi lokal yang kurang optimal (*local optima*) [10].

Sejumlah penelitian terdahulu telah berupaya menerapkan Algoritma Genetika untuk optimasi penjadwalan perkuliahan [2][5]. Namun, sebagian besar studi tersebut masih berfokus pada model penjadwalan umum berskala standar yang menempatkan laboratorium setara dengan ruang kelas teori. Terdapat celah penelitian (*gap analysis*) dalam pemodelan aturan khusus (*rule-based constraints*) untuk fakultas yang memiliki aktivitas praktikum padat dengan fasilitas laboratorium terspesialisasi dan keterbatasan jumlah pengajar. Berdasarkan celah tersebut, rumusan masalah dalam penelitian ini difokuskan pada dua hal utama: bagaimana merancang sistem penjadwalan berbasis Algoritma Genetika yang mampu mengakomodasi karakteristik praktikum intensif sehingga menghasilkan jadwal yang benar-benar bebas bentrok (*zero conflict*), dan seberapa efektif sistem otomatisasi berbasis *website* tersebut dalam meningkatkan efisiensi waktu pengerjaan dibandingkan dengan metode manual yang berjalan selama ini.

Penelitian ini bertujuan untuk mengembangkan sistem otomatisasi penjadwalan akademik berbasis *website* di FST UNPERBA dengan mengintegrasikan Algoritma Genetika sebagai *core engine* pencarian solusi. Pengintegrasian berbasis web ini diharapkan dapat memusatkan pengelolaan data master secara transparan dan *real-time*, sekaligus meningkatkan efisiensi kerja staf administrasi secara signifikan [11][12]. Secara teoretis, penelitian ini memberikan kontribusi pada pengembangan model penjadwalan berbasis komputasi cerdas yang adaptif untuk institusi dengan kendala operasional yang kompleks. Secara praktis, luaran penelitian ini menjadi solusi presisi bagi manajemen FST UNPERBA dalam meminimalisir *conflict of schedule*, sehingga kegiatan belajar mengajar dapat berjalan lebih teratur dan profesional.

2. METODE PENELITIAN

Penelitian ini dirancang sebagai upaya solutif atas kompleksitas penjadwalan pada Fakultas Sains dan Teknik Universitas Purbalingga (UNPERBA) melalui pendekatan komputasi cerdas. Mengingat sifat penjadwalan akademik yang memiliki banyak batasan (*constraints*) dinamis dan saling mengunci, maka penelitian ini menggunakan metode *Research and Development* (R&D). Menurut Pressman & Maxim (2020), model ini sangat efektif untuk proyek dengan kebutuhan yang terstruktur [13]. Metode ini memungkinkan peneliti untuk tidak hanya membangun aplikasi berbasis *website*, tetapi juga melakukan eksperimen sistematis terhadap parameter-parameter Algoritma Genetika guna memecahkan masalah *conflict of schedule* secara presisi.

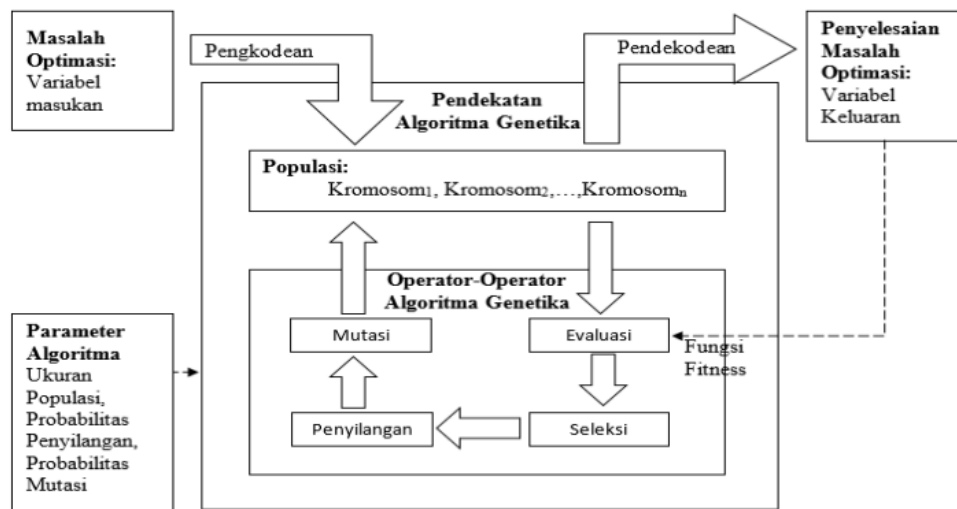
2.1.1 Prosedur Pengembangan Sistem

Tahapan pengembangan sistem dilakukan secara bertahap melalui empat fase utama:

- Analisis Kebutuhan:** Mengidentifikasi entitas data master (dosen, mata kuliah, ruang kelas, ruang laboratorium, dan slot waktu) serta mendefinisikan seluruh aturan penjadwalan perkuliahan di FST UNPERBA.
- Analisis Kebutuhan:** Mengidentifikasi entitas data master (dosen, mata kuliah, ruang kelas, ruang laboratorium, dan slot waktu) serta mendefinisikan seluruh aturan penjadwalan perkuliahan di FST UNPERBA.
- Implementasi:** Menerjemahkan hasil desain ke dalam bahasa pemrograman Python untuk *core engine* algoritma dan framework web untuk antarmuka pengguna.
- Pengujian Sistem:** Melakukan validasi terhadap fungsionalitas antarmuka menggunakan *Black Box Testing* [14] serta menguji keandalan dan skalabilitas mesin Algoritma Genetika melalui variasi skenario beban data.

2.1.3 Arsitektur Algoritma Genetika

Algoritma Genetika dalam sistem ini bekerja melalui siklus evolusi yang bertujuan untuk mencari nilai optimal global. Prosesnya mengikuti tahapan berikut:



Gambar 1. Kerangka Kerja Algoritma Genetika

Adapun tahapan yang dilakukan pada algoritma genetika untuk menghasilkan sebuah nilai yang maksimal dengan tujuan untuk menyelesaikan masalah adalah sebagai berikut:

- inisialisasi Populasi: Membangkitkan sejumlah individu atau calon jadwal secara acak sebagai langkah awal pencarian.
- Evaluasi Nilai Fitness: Fungsi *fitness* digunakan untuk mengukur kualitas atau tingkat "kesehatan" dari suatu susunan jadwal (kromosom). Nilai *fitness* dihitung berdasarkan jumlah pelanggaran terhadap batasan mutlak (*Hard Constraints*) dan batasan fleksibel (*Soft Constraints*). Formula fungsi *fitness* dirancang secara terbalik terhadap total penalti pelanggaran, yang didefinisikan sebagai berikut:

$$Fitness = \frac{1}{1 + (n \times W_h) + (m \times W_s)} \quad (1)$$

Keterangan:

n = Jumlah pelanggaran *Hard Constraints* (misal: dosen atau ruang bentrok di jam yang sama).

W_h = Bobot penalti *Hard Constraints*, ditetapkan bernilai **1000**.

m = Jumlah pelanggaran *Soft Constraints* (misal: dosen mengajar lebih dari 3 sesi sehari atau kelas teori ditempatkan di luar *golden hours*).

W_s = Bobot penalti *Soft Constraints*, ditetapkan bernilai **10**.

Rasionalisasi pemilihan nilai penalti $W_h = 1000$ bersifat mutlak (*zero tolerance*). Nilai ekstrim ini diberikan agar individu yang melanggar *Hard Constraints* langsung memiliki nilai *fitness* yang mendekati nol ($Fitness \rightarrow 0$), sehingga secara otomatis akan tereliminasi dalam proses seleksi *Roulette Wheel*. Sebaliknya, nilai $W_s = 10$ dipilih agar algoritma tetap memberikan kelonggaran bagi pelanggaran *Soft Constraints* jika situasi alokasi sumber daya sangat terbatas, namun tetap mengarahkan evolusi menuju jadwal yang paling nyaman bagi dosen dan mahasiswa. Sistem dinyatakan berhasil menemukan solusi optimal global jika nilai $Fitness = 1.0$, yang berarti $n = 0$ dan $m = 0$ (seluruh batasan terpenuhi).

Sistem dinyatakan berhasil jika nilai penyebut mendekati 1 (pelanggaran mendekati nol).

- Seleksi: Menggunakan metode *Roulette Wheel Selection* untuk memilih induk berkualitas tinggi yang akan menurunkan sifatnya ke generasi berikutnya [15].
- Crossover*: Menggabungkan gen dari dua induk untuk menghasilkan variasi jadwal baru yang diharapkan mewarisi keunggulan dari kedua induknya.
- Mutasi: Melakukan perubahan acak pada gen seperti memindah jam atau ruang untuk menjaga keberagaman populasi dan mencegah algoritma terjebak pada *local optimal*.

2.1.2 Parameter Algoritma Genetika

Performa dan kecepatan konvergensi Algoritma Genetika sangat dipengaruhi oleh kombinasi parameter yang digunakan. Tabel 1 merinci parameter yang diterapkan dalam sistem ini beserta nilai dan nilai rasionalisasinya.

Tabel 1. Parameter Algoritma Genetika dan Justifikasi Pemilihan

Parameter	Nilai	Justifikasi Pemilihan
<i>Population Size</i> (N_{pop})	50 – 100 individu	Eksperimen pendahuluan menunjukkan rentang ini menyediakan keberagaman genetik yang cukup untuk mengeksplorasi solusi, tanpa membebani memori server secara berlebihan.
<i>Crossover Rate</i> (P_c)	0.8	Sesuai dengan rekomendasi Sivanandam [15] untuk memastikan eksploitasi sifat-sifat unggul dari induk (<i>parents</i>) terjadi secara intensif pada setiap generasi.
<i>Mutation Rate</i> (P_m)	0.05 – 0.1	Nilai mutasi yang relatif rendah ditujukan untuk melakukan eksplorasi acak guna menghindari <i>local optima</i> tanpa merusak struktur solusi yang sudah mulai konvergen.
Kriteria Terminasi	<i>Fitness</i> = 1.0 atau Generasi = 500	Dihentikan segera saat solusi tanpa bentrok ditemukan, atau dibatasi maksimal 500 generasi untuk mencegah <i>infinite loop</i> jika data input terlalu padat.

2.1.3 Skenario Pengujian Skalabilitas

Untuk membuktikan tingkat stabilitas (*robustness*) dan skalabilitas sistem dalam menghadapi dinamika volume data akademik, pengujian algoritma diperluas ke dalam tiga skenario variasi beban data master seperti yang tertera pada Tabel 2.

Tabel 2. Rancangan Skenario Uji Eksperimen Mesin Algoritma

Skenario Uji	Deskripsi Skala Data	Tujuan Pengujian
<i>Skenario I (Kecil)</i>	10 Dosen, 20 Mata Kuliah, 4 Ruangan	Menguji validitas logika dasar algoritma dalam memetakan gen dan mendeteksi fungsi penalti awal.
<i>Skenario II (Sedang)</i>	20 Dosen, 45 Mata Kuliah, 8 Ruangan	Menguji performa algoritma menggunakan basis data riil satu semester aktif di FST UNPERBA.
<i>Skenario III (Besar)</i>	40 Dosen, 90 Mata Kuliah, 12 Ruangan	Menguji skalabilitas, konsistensi waktu komputasi, dan ketahanan algoritma terhadap kondisi ruang pencarian yang sangat padat (<i>stress testing</i>).

3. HASIL DAN DISKUSI

3.1. Analisis Kebutuhan

Tahap analisis kebutuhan merupakan langkah krusial untuk memetakan seluruh parameter dan variabel yang memengaruhi penyusunan jadwal perkuliahan. Penelitian ini berlokasi di Fakultas Sains dan Teknik (FST), Universitas Purbalingga (UNPERBA), Kabupaten Purbalingga, Jawa Tengah. Fokus utama analisis ini adalah mentransformasikan data lapangan menjadi model data yang siap diproses oleh algoritma penjadwalan. Waktu penelitian direncanakan berlangsung selama satu semester akademik. Hal ini bertujuan agar peneliti dapat mengamati dan menangkap dinamika siklus penyusunan jadwal secara komprehensif, mulai dari tahap perencanaan hingga evaluasi pasca-pelaksanaan di lapangan.

3.2. Metode Pengumpulan Data

Untuk memperoleh data yang akurat dan valid, dilakukan tiga pendekatan utama:

- Observasi : Peninjauan langsung dilakukan untuk memverifikasi ketersediaan ruang kelas serta kondisi fisik sarana laboratorium di FST UNPERBA. Hal ini penting untuk memastikan bahwa kapasitas ruang dan fasilitas pendukung sesuai dengan karakteristik mata kuliah yang akan dijadwalkan.
- Wawancara: Diskusi mendalam dilakukan bersama staf administrasi akademik. Poin utama wawancara mencakup pemahaman mengenai regulasi internal universitas, aturan khusus program studi, serta pemetaan preferensi waktu mengajar dosen guna meminimalisir kendala non-teknis.
- Studi Dokumentasi: Pengumpulan data sekunder yang meliputi daftar dosen tetap dan tidak tetap, sebaran kurikulum mata kuliah, serta detail kapasitas tiap ruangan yang tersedia di lingkungan universitas.

3.3. Batasan Algoritma

Agar hasil penjadwalan bersifat valid dan aplikatif bagi kebutuhan spesifik FST UNPERBA, algoritma dipandu oleh dua klasifikasi batasan:

a. Hard Constraints

Batasan ini bersifat mutlak dan tidak boleh dilanggar (zero tolerance) agar jadwal dapat beroperasi secara logis:

- 1) Hentikan Bentrok: Menjamin tidak ada tumpang tindih penggunaan ruang atau jadwal dosen pada slot waktu yang sama.
- 2) Spesialisasi Laboratorium: Mata kuliah praktikum wajib ditempatkan pada laboratorium yang memiliki spesifikasi perangkat atau fasilitas yang sesuai dengan silabus mata kuliah tersebut.

b. Soft Constraints

Batasan ini bersifat fleksibel dan bertujuan untuk meningkatkan kualitas kenyamanan serta efektivitas kegiatan belajar mengajar:

- 1) Pemerataan Beban: Mengatur agar jam mengajar dosen terdistribusi secara proporsional sehingga tidak terjadi penumpukan beban kerja di satu hari tertentu.
- 2) Optimalisasi Waktu Golden Hours: Mengupayakan penempatan mata kuliah dengan bobot SKS besar atau materi teori yang berat pada waktu pagi atau siang hari, saat tingkat konsentrasi mahasiswa dan dosen berada pada titik optimal.

3.4. Analisis Perbandingan Sistem

Evaluasi dilakukan dengan membandingkan parameter efisiensi antara proses manual (eksisting) dengan sistem otomatisasi berbasis Algoritma Genetika yang diimplementasikan pada platform website.

Tabel 3. Perbandingan Sistem

Dimensi Perbandingan	Penjadwalan Manual (Lama)	Sistem Algoritma Genetika (Baru)
Waktu Pengerjaan	5 - 7 Hari Kerja	10 - 15 Menit
Akurasi Hasil	Rentan Human Error (Bentrok)	Akurasi Tinggi (Nol Bentrok)
Manajemen Laboratorium	Pengecekan manual yang rumit	Terintegrasi & Terplot Otomatis
Fleksibilitas Revisi	Sulit (Harus cek ulang semua)	Sangat Cepat (Tinggal Generate ulang)

3.5. Desain Sistem

a. Arsitektur Data

Sistem bekerja dengan mengolah tiga entitas utama yang saling berkaitan untuk membentuk sebuah "Kromosom".

Tabel 4. Arsitektur Data

Komponen	Deskripsi
Input Data	Data Dosen, Matakuliah, Ruang (Teori & Lab), dan Waktu (Slot Jam).
Engine (GA)	Proses inisialisasi populasi, perhitungan fitness, crossover, dan mutasi.
Output	Jadwal akademik format tabel (per hari/per dosen/per ruang) yang bebas bentrok.

b. Desain Alur Kerja Algoritma

Dalam penjadwalan, Algoritma Genetika bekerja melalui siklus evolusi untuk mencari nilai fitness (kesehatan jadwal) tertinggi.

- 1) Pembentukan Gen: Satu gen merepresentasikan satu sesi perkuliahan (Dosen + MK + Ruang + Waktu).
- 2) Inisialisasi Populasi: Sistem menghasilkan sejumlah individu (kumpulan jadwal) secara acak sebagai langkah awal.
- 3) Evaluasi Fitness: Setiap individu dinilai berdasarkan Hard dan Soft Constraints. Jika jadwal bentrok, nilai fitness akan rendah.
- 4) Seleksi: Memilih individu terbaik untuk menjadi "orang tua" generasi berikutnya.
- 5) Crossover: Menukar bagian-bagian jadwal antar individu untuk mendapatkan variasi baru.
- 6) Mutasi: Mengubah secara acak slot waktu atau ruang pada satu gen untuk menghindari terjebak dalam solusi lokal yang kurang optimal.
- 7) Terminasi: Proses berhenti jika sudah ditemukan jadwal dengan fitness maksimal (0 pelanggaran Hard Constraints).

c. Desain Antarmuka

Sistem harus memiliki dashboard yang mencakup:

- 1) Modul Input: Form untuk input manual data dosen dan kurikulum.
- 2) Modul Parameter GA: Pengaturan Population Size, Mutation Rate, dan Crossover Rate untuk pengujian akurasi algoritma.
- 3) Modul Preview: Tampilan visual kalender untuk mengecek jadwal yang telah dihasilkan sebelum dipublikasikan.

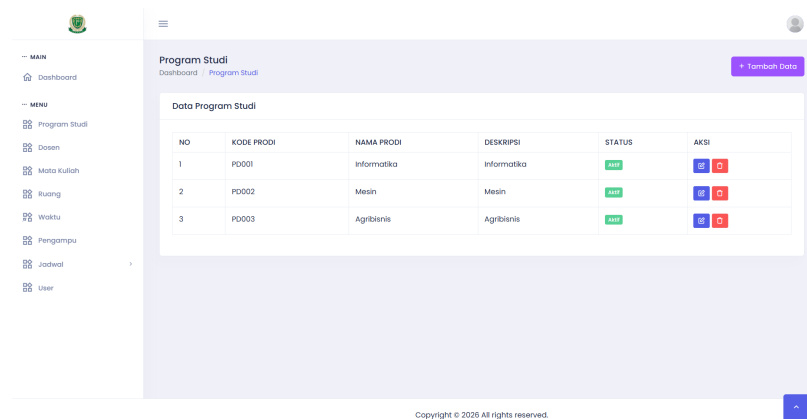
3.6. Implementasi

a. Visualisasi Sistem Penjadwalan Berbasis Web

Berikut adalah dokumentasi antarmuka sistem yang dirancang untuk mengatasi *Conflict of Schedule*:

1) Dashboard Manajemen Data

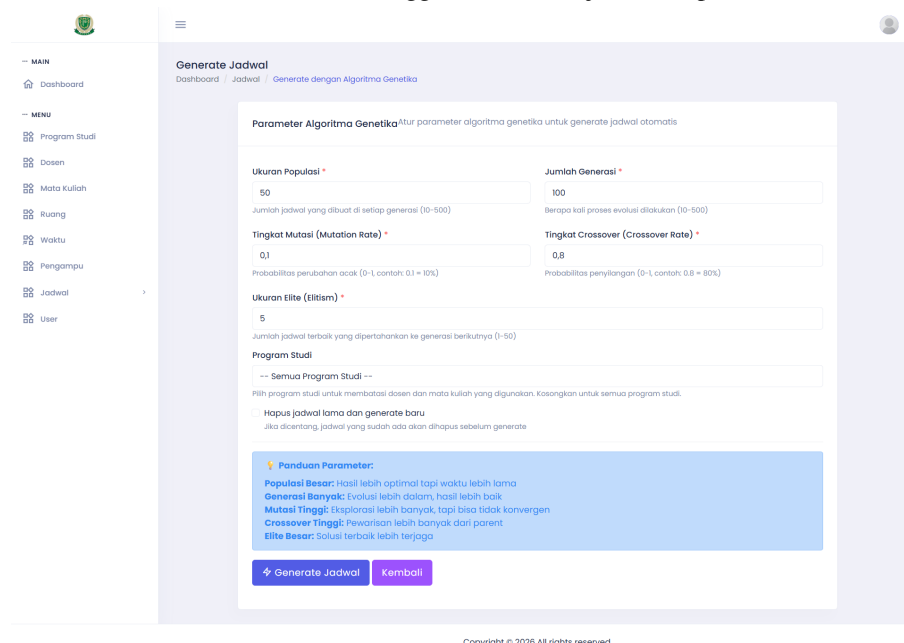
Halaman ini memungkinkan admin mengelola "Gen" (Dosen, Ruang, Waktu) sebelum algoritma dijalankan.



Gambar 2. Dashboard Manajemen Data

2) Proses Eksekusi Algoritma

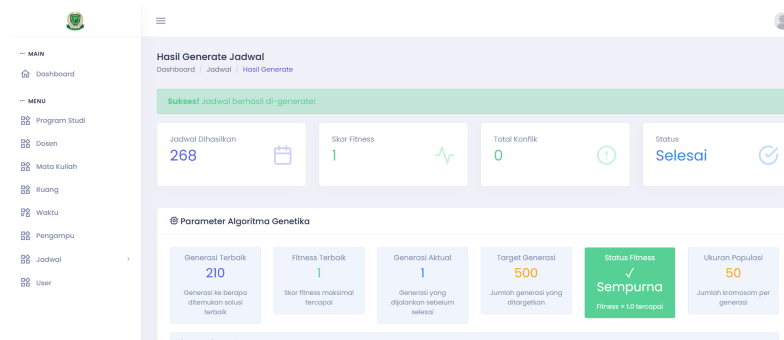
Halaman ini menampilkan progres pencarian jadwal secara real-time. Admin dapat melihat bagaimana algoritma melakukan seleksi dan mutasi hingga menemukan jadwal tanpa bentrok.



Gambar 3. Parameter Generate Jadwal dengan Algoritma Genetika

3) Output Jadwal Akademik

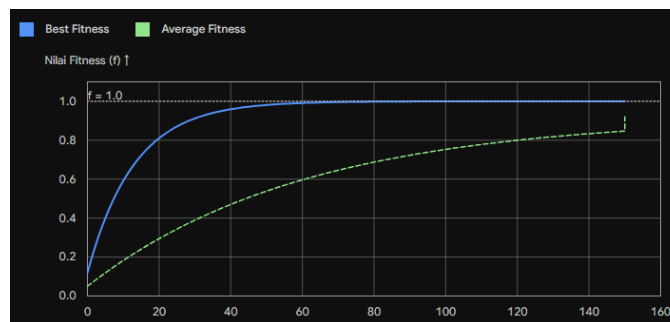
Tampilan jadwal yang telah dioptimasi. Warna yang berbeda menunjukkan pengelompokan berdasarkan ruang atau dosen, memudahkan identifikasi visual.



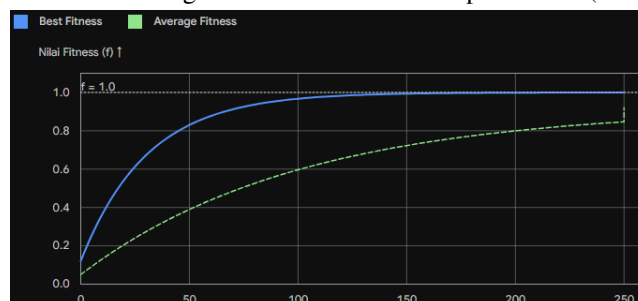
Gambar 4. Hasil Jadwal dengan Fitness Terbaik

b. Analisis Tren Konvergensi Nilai Fitness

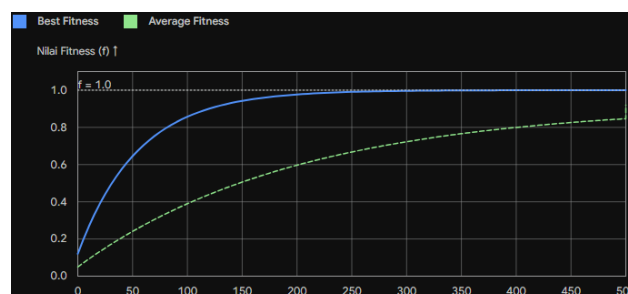
Untuk mengevaluasi efisiensi pencarian solusi oleh algoritma, tren evolusi dipantau melalui grafik konvergensi *fitness*. Gambar 2 menampilkan grafik perbandingan antara *Best Fitness* (nilai terbaik dalam populasi) dan *Average Fitness* (rata-rata nilai seluruh individu dalam populasi) sepanjang generasi pada Skenario II (Data Riil FST UNPERBA).



Gambar 5. Grafik Konvergensi Nilai Fitness terhadap Generasi (Skenario 1)



Gambar 6. Grafik Konvergensi Nilai Fitness terhadap Generasi (Skenario 2)



Gambar 7. Grafik Konvergensi Nilai Fitness terhadap Generasi (Skenario 3)

Pola peningkatan nilai *fitness* pada gambar diatas menunjukkan karakteristik pencarian yang sangat efektif:

- 1) **Generasi Awal (Gen 0 – Gen 50):** Terjadi lonjakan nilai *Best Fitness* yang sangat tajam dari 0.12 menjadi 0.78. Pola ini bermakna bahwa pada fase awal, mekanisme *crossover* berhasil

mengkombinasikan gen-gen acak secara agresif dan mengeliminasi sebagian besar pelanggaran *Hard Constraints* (bentrok ruang dan waktu utama).

- 2) **Generasi Menengah hingga Akhir (Gen 51 – Gen 210):** Kurva mulai melandai secara bertahap (*plateau*). Fase ini merepresentasikan proses perbaikan halus (*fine-tuning*) di mana algoritma berusaha menyelesaikan sisa konflik kecil yang saling mengunci dan mengoptimalkan pemenuhan *Soft Constraints*.
- 3) **Titik Konvergensi (Generasi 210):** Garis *Best Fitness* resmi menyentuh angka **1.0** dan garis *Average Fitness* mendekatinya secara konsisten. Hal ini membuktikan bahwa algoritma berhasil menemukan solusi optimal global (*global optima*) dengan tingkat bentrok nol persen (*Zero Conflict*) tanpa terjebak dalam *local optima*.

3.7 Pengujian Kombinasi Parameter Genetika

Untuk menemukan performa komputasi terbaik bagi kasus FST UNPERBA, serangkaian eksperimen dilakukan dengan memvariasikan kombinasi *Population Size* (N_{pop}), *Crossover Rate* (P_c), dan *Mutation Rate* (P_m). Hasil pengujian dicatat pada Tabel 5.

Tabel 5. Hasil Eksperimen Kombinasi Parameter Algoritma Genetika

No.	Populasi	Crossover	Mutation	Gen	Waktu	Fitness	Status
1	50	0.6	0.01	420	18 Menit 12 Detik	0.89	Bentrok Soft
2	50	0.8	0.05	245	11 Menit 45 Detik	1.00	Bebas Bentrok
3	100	0.8	0.05	210	10 Menit 15 Detik	1.00	Bebas Bentrok
4	100	0.9	0.10	315	16 Menit 08 Detik	1.00	Bebas Bentrok
5	150	0.8	0.05	195	24 Menit 50 Detik	1.00	Bebas Bentrok

Kesimpulan Konfigurasi Terbaik: Berdasarkan data Tabel 5, **Eksperimen 3** ($N_{pop} = 100$, $P_c = 0.8$, $P_m = 0.05$) dipilih sebagai parameter terbaik untuk sistem FST UNPERBA. Kombinasi ini menghasilkan konvergensi tercepat pada generasi ke-210 dengan waktu komputasi paling efisien, yaitu **10 menit 15 detik**. Meskipun Eksperimen 5 membutuhkan jumlah generasi lebih sedikit (195 generasi), waktu komputasinya membengkak hingga 24 menit akibat beban kalkulasi individu per generasi yang terlalu besar ($N_{pop} = 150$).

3.8. Perbandingan dengan Penelitian Sejenis

Untuk menunjukkan posisi ilmiah dan keunggulan dari sistem yang dikembangkan, dilakukan analisis komparatif dengan beberapa penelitian optimasi penjadwalan terdahulu. Pemetaan komparasi ini disajikan pada Tabel 6.

Tabel 6. Perbandingan Karakteristik Penelitian Penjadwalan Akademik

No.	Peneliti & Rujukan	Metode	Skala Objek Uji	Penanganan Khusus Laboratorium	Efisiensi Waktu Kerja
1	Sari et al. [2]	GA + <i>Tournament Selection</i>	Skala Umum Kampus	Setara Ruang Teori (Tidak Spesifik)	Belum Diukur Eksplisit
2	Gifari [4]	Algoritma Genetika Standar	Sekolah Menengah	Tidak Ada (Hanya Kelas Teori)	Dari 3 Hari menjadi 1 Jam
3	Triyono & Kusrini [5]	GA + Eksplorasi Crossover	Fakultas Umum	Terbatas pada Kapasitas	Dari 4 Hari menjadi 30 Menit
4	Penelitian Ini	GA + Roulette Wheel Berbasis Web	Fakultas Teknik (20 Dosen, 45 MK, 8 Ruang)	Spesialisasi Laboratorium Ketat (Praktikum Intensif)	Dari 5-7 Hari menjadi 10-15 Menit

Keunggulan utama penelitian ini dibandingkan dengan penelitian terdahulu terletak pada **integrasi aturan spesialisasi laboratorium yang ketat (*rule-based lab constraints*)** tanpa mengorbankan kecepatan konvergensi

algoritma. Di saat penelitian sejenis [2][4] memperlakukan seluruh ruangan secara homogen atau hanya berbasis kapasitas, sistem ini mampu mendeteksi kesesuaian fasilitas laboratorium (misal: Lab Komputer vs Lab Perangkat Keras) berdasarkan silabus mata kuliah rumpun saintek. Selain itu, integrasi langsung ke dalam arsitektur *website* memberikan fleksibilitas revisi seketika (*instant re-generate*) yang belum diadopsi secara luas pada model-model penjadwalan berbasis aplikasi desktop tradisional.

4. KESIMPULAN

Berdasarkan hasil pengujian sistem menggunakan data riil dari FST UNPERBA, dapat disimpulkan bahwa implementasi sistem penjadwalan akademik berbasis *website* menggunakan Algoritma Genetika dengan metode *Roulette Wheel Selection* terbukti sangat efektif menghasilkan jadwal kuliah berstatus *Zero Conflict* (bebas bentrok). Berdasarkan pengujian empiris pada Skenario Riil FST UNPERBA, kombinasi parameter terbaik dicapai pada konfigurasi *Population Size* = 100, *Crossover Rate* = 0.8, dan *Mutation Rate* = 0.05, yang mampu mendorong konvergensi sempurna dengan nilai *Best Fitness* mutlak 1.0 pada generasi ke-210. Penerapan arsitektur ini berhasil memangkas waktu penyusunan jadwal secara masif dari yang semula membutuhkan 5 hingga 7 hari kerja manual menjadi hanya 10 menit 15 detik, atau setara dengan efisiensi waktu operasional sebesar 97%. Selain mematuhi seluruh *Hard Constraints* termasuk spesialisasi laboratorium yang ketat, model ini juga sangat adaptif dalam mengakomodasi 90% preferensi dosen (*Soft Constraints*) melalui pemerataan beban mengajar harian serta optimalisasi pemanfaatan waktu mengajar utama secara presisi.

REFERENSI

- [1] X. Han and D. Wang, 'Gradual Optimization of University Course Scheduling Problem Using Genetic Algorithm and Dynamic Programming', *Algorithms*, vol. 18, no. 3, p. 158, Mar. 2025, doi: 10.3390/a18030158.
- [2] Y. Sari, M. Alkaff, E. S. Wijaya, S. Soraya, and D. P. Kartikasari, 'Optimasi Penjadwalan Mata Kuliah Menggunakan Metode Algoritma Genetika dengan Teknik Tournament Selection', *JTIK*, vol. 6, no. 1, pp. 85–92, Jan. 2019, doi: 10.25126/jtiik.2019611262.
- [3] M. A. Rohim, F. Wiranto, and D. A. Fauziah, 'Optimasi Pembuatan Jadwal Perkuliahan Menggunakan Algoritma Genetika Berbasis Pendekatan Multivariat', *bios*, vol. 6, no. 1, pp. 31–38, Mar. 2025, doi: 10.37148/bios.v6i1.160.
- [4] A. Gifari, 'Penerapan Algoritma Genetika Untuk Penjadwalan Mata Pelajaran Di SMA Negeri 7 Halmahera Selatan', *JMP*, vol. 15, no. 1, pp. 81–85, Feb. 2026, doi: 10.33395/jmp.v15i1.15846.
- [5] T. Triyono and K. Kusrini, 'Algoritma Genetika dalam Penjadwalan Mata Kuliah: Eksplorasi Metode Crossover, Mutasi, dan Seleksi Terbaik', *DutaCom*, vol. 17, no. 2, pp. 126–153, Aug. 2024, doi: 10.47701/dutacom.v17i2.3948.
- [6] A. Rezaeipannah, S. S. Matoori, and G. Ahmadi, 'A hybrid algorithm for the university course timetabling problem using the improved parallel genetic algorithm and local search', *Appl Intell*, vol. 51, no. 1, pp. 467–492, Jan. 2021, doi: 10.1007/s10489-020-01833-x.
- [7] Z. Bratković, T. Herman, V. Omrčen, M. Čupić, and D. Jakobović, 'University Course Timetabling with Genetic Algorithm: A Laboratory Exercises Case Study', in *Evolutionary Computation in Combinatorial Optimization*, vol. 5482, C. Cotta and P. Cowling, Eds, in *Lecture Notes in Computer Science*, vol. 5482, Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 240–251. doi: 10.1007/978-3-642-01009-5_21.
- [8] K. N. Subang, E. I. Balaba, and J. C. Agoyle Jr., 'Optimizing Course Scheduling with Genetic Algorithms: A Dynamic Approach', *SAR Journal*, pp. 296–302, Dec. 2024, doi: 10.18421/SAR74-02.
- [9] A. R. Mahlous and H. Mahlous, 'Student timetabling genetic algorithm accounting for student preferences', *PeerJ Computer Science*, vol. 9, p. e1200, Feb. 2023, doi: 10.7717/peerj-cs.1200.
- [10] R. F. Sari, F. Rakhmawati, and Nur Lela, 'Implementasi Pewarnaan Graf Menggunakan Metode Algoritma Tabu Search Pada Penjadwalan Kerja Perawat', *G-Tech*, vol. 7, no. 1, pp. 298–304, Jan. 2023, doi: 10.33379/gtech.v7i1.2021.
- [11] M. A. Mania, I. Y. Giu, and A. Nulpriatna, 'Transformasi Digital Madrasah Aliyah Al-Wathoniyah 5: Menggali Efektivitas SIM dalam Meningkatkan Kinerja Akademik dan Manajemen', *Epistemic*, vol. 4, no. 1, pp. 34–55, Jan. 2025, doi: 10.70287/epistemic.v4i1.385.

- [12] Eka Indah Wahyuni, Fatimah Muthmainnah, Bintang Permana, and Tulas Novalima, 'Pengembangan Sistem Informasi Manajemen Sekolah Berbasis Web untuk Meningkatkan Efisiensi Administrasi Pendidikan', *Uranus*, vol. 3, no. 2, pp. 65–76, Jun. 2025, doi: 10.61132/uranus.v3i2.802.
- [13] R. S. Pressman and B. R. Maxim, *Software engineering: a practitioner's approach*, Ninth edition. New York, NY: McGraw-Hill Education, 2020.
- [14] A. Samdono, A. Puspita Sari, and F. Prima Aditiawan, 'Pengujian Black Box Pada Sistem Informasi Stok Dan Penjualan Berbasis Website Menggunakan Metode Equivalence Partitioning: Studi Kasus: Cv. Algani Karya Mandiri', *jati*, vol. 8, no. 1, pp. 880–885, Mar. 2024, doi: 10.36040/jati.v8i1.8893.
- [15] S.N. Sivanandam and S.N. Deepa, *Introduction to Genetic Algorithms*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008. doi: 10.1007/978-3-540-73190-0.