

Optimasi Jadwal Perkuliahan Berbasis Web Menggunakan Algoritma Genetika: Studi Kasus Departemen Teknik Elektronika Universitas Negeri Padang

Nurhawa Nabila ^{1*}, Rizkayeni Marta ², Titi Sriwahyuni ³, Syafrijon ⁴

¹⁻⁴ Program Studi Pendidikan Teknik Informatika, Departemen Teknik Elektronika,
Fakultas Teknik, Universitas Negeri Padang

*Penulis Korespondensi: nurhawanabila019@gmail.com

Abstract. *Manual course scheduling often precipitates timetable conflicts and operational time inefficiencies, a distinct constraint currently faced by the Department of Electronics Engineering at Universitas Negeri Padang. This study aims to develop a web-based classroom scheduling information system to automate the computational formulation of schedule matrices integrated with academic administration. The software development was designed utilizing the linear sequential model (Waterfall method). The core optimization engine is driven by a Genetic Algorithm to reconcile both hard constraints and soft constraints, further complemented by a Partial Re-optimization functionality based on the Minimal Perturbation Problem (MPP) to mitigate sudden scheduling disruptions. The functionality validation via User Acceptance Testing (UAT) yielded an absolute success rate of 100%, demonstrating the algorithm's capability to generate completely zero-conflict schedules. Furthermore, the usability evaluation using the System Usability Scale (SUS) instrument achieved an average score of 80.21, positioning it within the "Excellent" category. Empirically, the deployment of this system successfully compressed the scheduling duration from one to two weeks down to mere minutes, while simultaneously providing an interactive interface that substantially enhances the productivity of academic governance.*

Keywords: *Genetic Algorithm; Optimization; Scheduling; Web Application; Waterfall; System Usability Scale*

Abstrak. Penyusunan jadwal perkuliahan secara manual sering kali memicu bentrok jadwal dan inefisiensi waktu operasional, sebagaimana kendala yang dialami oleh Departemen Teknik Elektronika Universitas Negeri Padang. Penelitian ini bertujuan untuk mengembangkan sistem informasi penjadwalan ruang kelas perkuliahan berbasis web guna mengotomatisasi komputasi penyusunan matriks jadwal yang terintegrasi dengan administrasi akademik. Pengembangan perangkat lunak dirancang menggunakan metode sekuensial linier (*Waterfall*). Mesin optimasi pada sistem ini digerakkan oleh Algoritma Genetika untuk menyelaraskan batasan mutlak (*hard constraints*) dan batasan fleksibel (*soft constraints*), serta didukung oleh fungsionalitas Re-optimasi Parsial berbasis *Minimal Perturbation Problem* (MPP) untuk mitigasi perubahan jadwal yang mendadak. Hasil pengujian fungsionalitas melalui *User Acceptance Testing* (UAT) mencatat tingkat keberhasilan absolut sebesar 100%, membuktikan kemampuan algoritma dalam mencetak jadwal yang sepenuhnya bebas konflik (*zero-conflict*). Selain itu, evaluasi tingkat kebergunaan menggunakan instrumen *System Usability Scale* (SUS) menghasilkan skor 80,21 (kategori *Excellent*). Secara empiris, implementasi sistem ini berhasil mereduksi waktu penyusunan jadwal dari satu hingga dua minggu menjadi hanya dalam hitungan menit, sekaligus menyediakan antarmuka interaktif yang meningkatkan produktivitas tata kelola akademik secara signifikan.

Kata kunci: *Algoritma Genetika; Optimasi; Penjadwalan; Aplikasi Web; Waterfall; System Usability Scale*

1. LATAR BELAKANG

Manajemen penjadwalan merupakan pilar utama operasional perguruan tinggi karena berdampak langsung pada efisiensi penggunaan sumber daya dan kelancaran studi.

Jadwal yang efektif menghindari tumpang tindih dengan menyinkronkan waktu, dosen, mata kuliah, ruang kelas, dan kelompok mahasiswa. Abdullah dkk. (2024) mendefinisikan proses ini sebagai masalah optimasi tingkat tinggi akibat adanya kewajiban untuk menyelaraskan batasan mutlak (*hard constraints*) dengan preferensi pengguna (*soft constraints*). Jadwal juga sangat berpengaruh pada seberapa efisien tata ruang dikelola, bagaimana beban mengajar dosen didistribusikan, dan seberapa efektif kegiatan akademik dapat berjalan (Romaguera dkk., 2024).

Pada kenyataannya, pembuatan jadwal perkuliahan merupakan tugas yang penuh tantangan akibat tingginya keterkaitan antarvariabel. Seiring dengan peningkatan jumlah mata kuliah, staf pengajar, kapasitas ruangan, hingga kelas paralel, alternatif kombinasi jadwal melonjak secara signifikan. Dalam situasi seperti ini, metode manual menjadi sangat tidak efektif karena pihak penyusun harus menghindari tumpang tindih jadwal dosen, bentrok pemakaian ruangan, dan fungsi fasilitas yang tidak sesuai dengan tujuan pembelajaran. Mahlous dan Mahlous (2023) menyatakan bahwa mekanisme manual sudah tidak relevan lagi untuk menangani kompleksitas data akademik saat ini, sehingga transisi menuju pendekatan optimasi berbasis komputasi menjadi sebuah keharusan.

Kondisi serupa saat ini masih dialami oleh Departemen Teknik Elektronika, Universitas Negeri Padang. Temuan observasi di lapangan menunjukkan bahwa pengelola akademik masih mengandalkan Microsoft Excel sebagai instrument utama dalam menyusun jadwal. Seluruh data dosen, mata kuliah, ruang teori, laboratorium, serta kelompok mahasiswa dicocokkan secara manual sehingga membutuhkan waktu sekitar satu hingga dua minggu untuk menghasilkan satu matriks jadwal. Selain menuntut ketelitian tingkat tinggi, prosedur manual ini juga dapat menyebabkan konflik jadwal apabila terjadi pembaruan data, seperti pergantian dosen pengampu, pembukaan kelas baru, atau relokasi ruangan. Hasilnya, penyusun harus memverifikasi ulang sebagian besar draf yang sudah dibuat, yang pada akhirnya mengurangi efisiensi kinerja operasional.

Untuk merespons problematika ini, banyak riset telah mengeksplorasi penerapan metode optimasi, di mana Algoritma Genetika (*Genetic Algorithm*) menjadi salah satu pendekatan yang paling banyak diadopsi karena kemampuannya memecahkan ruang pencarian yang masif (Pamungkas & Neta, 2025). Algoritma ini terbukti andal dalam

menemukan solusi terbaik melalui tahapan evolusi, mulai dari pembentukan populasi, perhitungan nilai kebugaran (*fitness*), proses seleksi, pindah silang (*crossover*), hingga mutasi. Penelitian Romaguera dkk. (2024) membuktikan bahwa penerapan *Enhanced Genetic Algorithm* yang ditingkatkan pada sistem penjadwalan web dapat menghasilkan draf yang memenuhi pada seluruh batasan mutlak (*hard constraints*) dan mengoptimalkan kualitas jadwal berdasarkan *soft constraints*. Abdullah dkk. (2024) menemukan hasil yang serupa, menunjukkan bahwa algoritma genetika yang menggunakan representasi kromosom berbasis matriks mampu mempercepat pencarian solusi komputasi dan menekan angkat konflik penjadwalan secara lebih efektif dibandingkan dengan metode manual.

Oleh karena itu, tinjauan literatur sebelumnya menunjukkan bahwa penelitian seringkali terbatas pada optimasi jadwal murni tanpa mengintegrasikan penelitian dengan sistem administrasi akademik secara menyeluruh. Sebagian besar sistem saat ini hanya menghasilkan draf jadwal otomatis, tetapi tidak memiliki fitur seperti publikasi jadwal yang terpusat, validasi, pengelolaan ruang kelas, manajemen basis data dosen, dan penawaran mata kuliah. Sistem juga memiliki kelemahan ketika perlu merespons perubahan data akademik yang mendadak. Penyusun seringkali harus menyusun ulang semuanya dari awal alih-alih menyesuaikan secara instan. Kondisi tersebut menunjukkan bahwa masih diperlukan pengembangan sistem yang tidak hanya mampu menghasilkan jadwal yang optimal, tetapi juga mendukung pengelolaan informasi akademik secara menyeluruh.

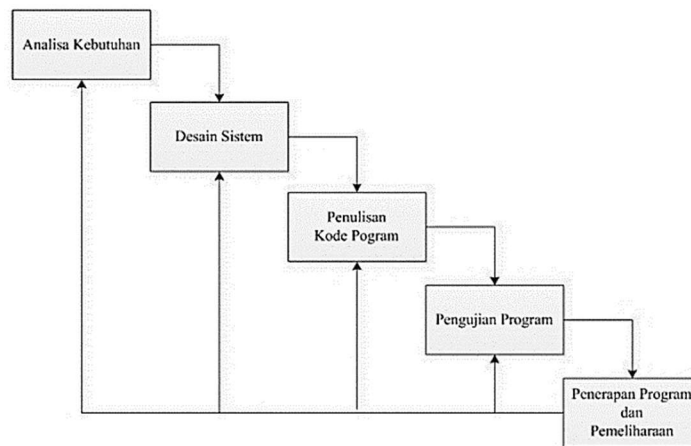
Berdasarkan kondisi tersebut, penelitian ini berkonsentrasi pada pengembangan aplikasi penjadwalan ruang kelas perkuliahan berbasis web yang menggunakan Algoritma Genetika pada Departemen Teknik Elektronika Universitas Negeri Padang. Sistem ini dirancang untuk mengintegrasikan pengelolaan data akademik, proses optimasi jadwal, validasi, dan publikasi jadwal dalam satu platform yang dapat diakses sesuai hak akses pengguna. Untuk pengembangan perangkat lunak, penelitian ini menggunakan metode *Waterfall*, sementara uji validitas kelayakannya diukur melalui *User Acceptance Testing* (UAT) dan instrumen *System Usability Scale* (SUS) untuk mengevaluasi kesesuaian fungsional dan tingkat penerimaan pengguna. Inovasi sistem informasi ini diharapkan dapat mempercepat durasi penjadwalan, mengurangi risiko jadwal bentrok,

dan meningkatkan produktivitas administrasi akademik di Departemen Teknik Elektronika Universitas Negeri Padang.

2. METODE PENELITIAN

2.1 Kerangka Pengembangan Sistem (Model *Waterfall*)

Dalam merancang sistem informasi penjadwalan ini, peneliti bertumpu pada metode sekuensial linier atau yang lazim dikenal sebagai model *Waterfall*. Pendekatan rekayasa perangkat lunak ini diaplikasikan karena selaras dengan sifat parameter penjadwalan (*constraints*) dan aturan main akademik di Departemen Teknik Elektronika Universitas Negeri Padang yang cenderung statis dan sudah terpetakan sejak awal. Melalui alur kerja yang berjalan runut ke bawah, setiap tahapan diwajibkan tuntas secara komprehensif sebelum berpindah ke tahap selanjutnya. Hal ini dinilai efektif untuk mereduksi potensi anomali logika pada saat fase penulisan kode dilakukan (Pressman & Maxim, 2020; Wahid, 2020). Penggunaan *Waterfall* memastikan bahwa setiap fase, mulai dari analisis hingga pengujian, tervalidasi sebelum sistem diimplementasikan (Mulyanto & Hakim, 2025).



Gambar 1. Tahapan Pengembangan Sistem dengan Model *Waterfall*

Implementasi rekayasa sistem dalam penelitian ini diuraikan ke dalam empat tahapan operasional utama sebagai berikut:

a. Analisis Kebutuhan (*Requirement Analysis*)

Identifikasi hambatan sistem dilakukan melalui teknik observasi lapangan dan wawancara mendalam dengan unit pengelola akademik (Wahid, 2020). Fase ini

menghasilkan dokumen yang menjelaskan semua kebutuhan sistem, baik fungsional maupun non-fungsional.

b. Perancangan Sistem (*System Design*)

Proses mentransformasikan dokumen spesifikasi menjadi cetak biru teknis (*blueprint*). Langkah ini mencakup desain struktur database MySQL, pemodelan visual menggunakan Unified Modeling Language (UML) seperti *use case* serta *activity diagram*, dan perancangan tata letak antarmuka pengguna (*user interface*).

c. Penulisan Kode Program (*Implementation*)

Fase konversi cetak biru menjadi aplikasi nyata dengan menuliskan sintaks pemrograman PHP di atas *framework* Laravel (dengan arsitektur *Model-View-Controller*). Pada tahapan krusial ini, logika kecerdasan dari Algoritma Genetika mulai ditanamkan ke dalam mesin sistem untuk memproses penentuan ruang dan waktu kuliah.

d. Pengujian Program (*Testing*)

Pemeriksaan kualitas aplikasi menggunakan metode *Black-box testing* guna menjamin seluruh fitur beroperasi dengan benar (Rosa & Shalahuddin, 2018). Mengingat fokus utama riset ini bertumpu pada pembuktian algoritma (*proof of concept*), tahap pemeliharaan sistem jangka panjang (*maintenance*) tidak disertakan. Sementara itu, pengukuran tingkat kepuasan dan kemudahan pakai bagi *end-user* diukur menggunakan *User Acceptance Testing* (UAT) serta kuesioner *System Usability Scale* (SUS) yang terbukti valid dalam mengevaluasi interaksi manusia dan komputer (Setiawati dkk., 2023; Suria, 2024).

2.2 Analisis Kebutuhan Operasional dan Solusi Algoritma

Menurut analisis sistem berjalan, manajemen penjadwalan konvensional yang menggunakan Microsoft Excel memerlukan waktu kerja sekitar satu hingga dua minggu. Kesalahan visual, atau kesalahan manusia, yang dapat menyebabkan tumpang tindih jadwal, sangat mungkin terjadi selama proses pemetaan manual. Sangat penting untuk beralih ke sistem optimasi berbasis komputasi karena data yang sangat dinamis dan rumit di institusi pendidikan saat ini (Mahlous & Mahlous, 2023).

Aplikasi yang dikembangkan dirancang untuk menyatukan otorisasi empat aktor utama (*Admin Akademik, Ketua Program Studi, Dosen Pengampu, dan Admin Sistem*) ke dalam satu platform terintegrasi. Tabel 1 menyajikan rincian masalah khusus dan intervensi teknologi yang diterapkan.

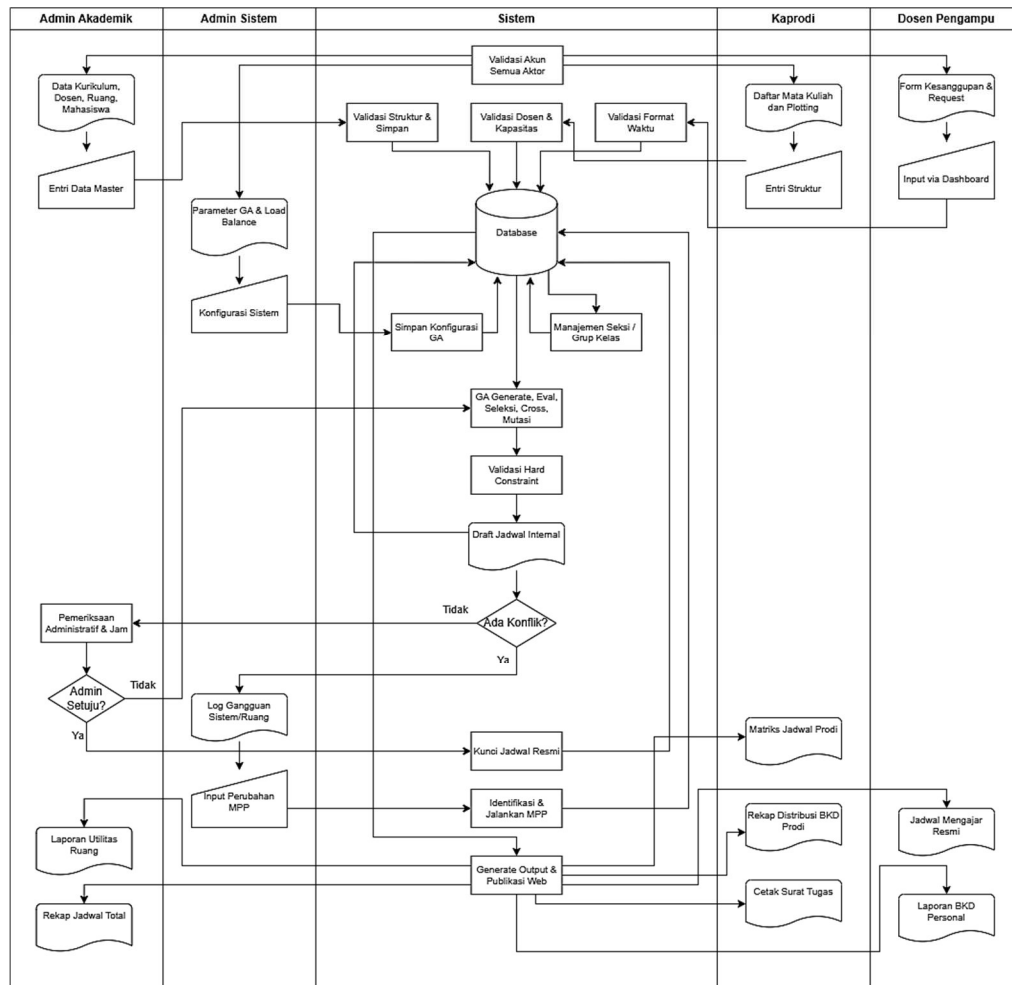
Tabel 1. Matriks Problematika Penjadwalan dan Intervensi Solusi Komputasi

No	Kendala Operasional Saat Ini	Strategi Penyelesaian Berbasis Sistem
1	Pembuatan matriks secara manual menghabiskan waktu hingga 14 hari serta berpeluang besar menciptakan draf yang tumpang tindih.	Menerapkan mesin <i>auto-generate</i> berbasis Algoritma Genetika untuk mencetak susunan jadwal yang bebas konflik (<i>zero-conflict</i>) pada parameter syarat mutlak (<i>hard constraints</i>).
2	Proses pendataan ketersediaan waktu mengajar dosen dilakukan secara tidak terstruktur, sehingga sulit untuk dikompilasi ulang.	Mengembangkan antarmuka <i>Lecturer Request Dashboard</i> yang memungkinkan dosen memasukkan langsung preferensi jam mengajar sebagai syarat kelonggaran (<i>soft constraints</i>).
3	Perubahan data secara mendadak memaksa staf pengelola untuk merombak ulang matriks jadwal secara keseluruhan.	Memasukkan fungsionalitas Re-Optimasi Parsial berlandaskan teori <i>Minimal Perturbation Problem</i> (MPP) guna menyesuaikan slot yang bermasalah saja tanpa merusak komposisi jadwal yang sudah stabil.
4	Penghitungan jumlah beban Satuan Kredit Semester (SKS) dari masing-masing dosen lintas prodi sulit dipantau secara kilat.	Merancang Dasbor Rekapitulasi yang bertugas mengakumulasi dan menampilkan grafik jam terbang dosen secara seketika (<i>real-time</i>).

2.3 Pemodelan Aliran Sistem dan Arsitektur Algoritma

Aliran data dan interaksi hak akses dimodelkan menggunakan diagram terstandarisasi untuk menjamin validitas konversi data ke dalam kode program.

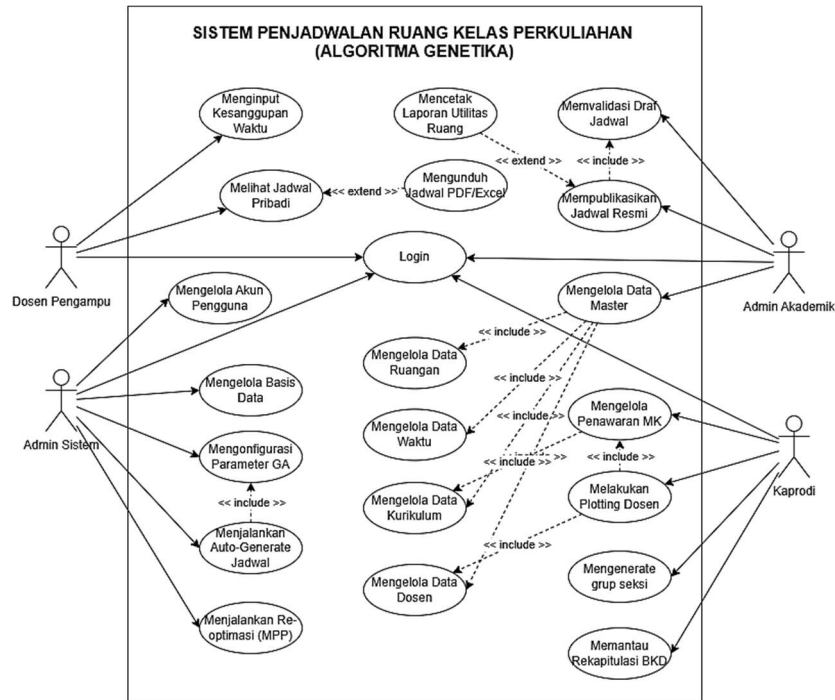
a. Flowmap



Gambar 2. Flowmap Sistem Penjadwalan

Gambar 2 menunjukkan flowmap yang divisualisasikan alur pergerakan data akademik. Siklus ini dimulai dengan pendataan komponen kurikulum oleh Kaprodi. Kemudian, Dosen Pengampu mengisi preferensi ketersediaan waktu mengajar. Mesin Algoritma Genetika kemudian mengolah data. Sebelum draf jadwal selesai, Administrasi Akademik melakukan validasi. Evolusi algoritma melibatkan pembentukan populasi awal (inisialisasi kromosom berbasis matriks), evaluasi nilai *fitness* (fungsi *fitness*) berdasarkan pemenuhan *constraints*, seleksi individu, pindah silang (crossover), dan mutasi gen untuk meminimalkan nilai penalti konflik (Abdullah dkk., 2024; Romaguera dkk., 2024).

b. Use Case Diagram



Gambar 3. Use Case Diagram

Pembatasan hak operasional masing-masing aktor ditunjukkan dalam Use Case Diagram di Gambar 3. Diagram ini menegaskan bahwa ada pemisahan wewenang yang jelas antara admin akademik dan kaprodi (yang menangani manajemen data administratif), dosen pengampu (yang menangani pengajuan preferensi tentang ketersediaan waktu mengajar), dan admin sistem (yang menangani kendali teknis operasional algoritma).

3. HASIL DAN PEMBAHASAN

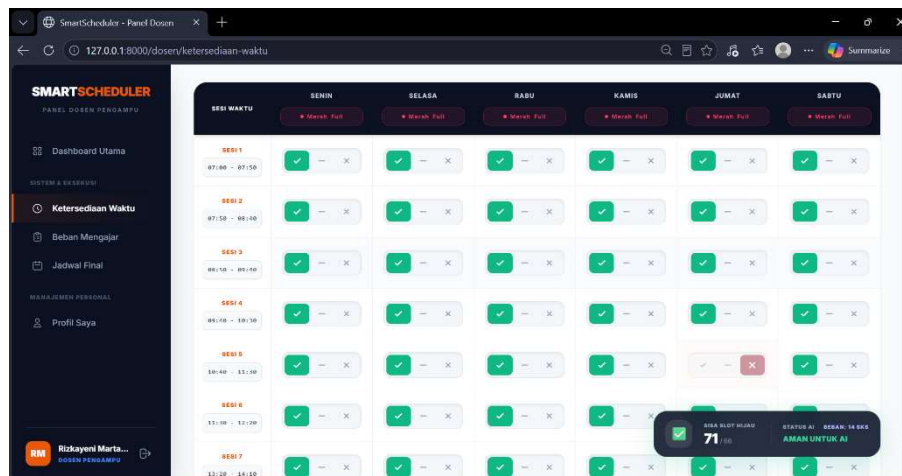
3.1 Implementasi Arsitektur Sistem Penjadwalan

Pengembangan sistem informasi penjadwalan (*SmartScheduler*) ini menghasilkan sebuah *platform* web terintegrasi yang mampu mengotomatisasi penyusunan matriks jadwal perkuliahan secara cerdas. Arsitektur sistem sengaja dirancang untuk mendobrak kekakuan metode konvensional di mana staf akademik harus menyusun jadwal secara manual menggunakan lembar kerja, yang memakan waktu lama dan rentan bentrok. Aplikasi ini dibangun dengan mengimplementasikan arsitektur *Model-View-Controller* (MVC) menggunakan *framework* Laravel (PHP) dan sistem manajemen basis data

MySQL. Platform ini bertindak sebagai wadah kolaborasi digital yang mempertemukan kebutuhan ruang (Admin Akademik), plotting mata kuliah (Kaprodi), ketersediaan waktu dosen (Dosen), dan mesin kecerdasan buatan (Admin Sistem) ke dalam satu pangkalan data yang terpusat.

Sebuah platform web yang dikembangkan sebagai bagian dari proyek SmartScheduler berfungsi untuk menggabungkan mesin kecerdasan buatan dengan manajemen data akademik. Dengan mendigitalisasi seluruh variabel penjadwalan ke dalam pangkalan data terpusat, sistem ini mengurangi kebutuhan akan intervensi manual. Platform ini memiliki pola arsitektur Model-View-Controller (MVC) yang memisahkan logika komputasi dari antarmuka pengguna. Ini memungkinkan sistem untuk menangani beban data komputasi yang besar tanpa mengurangi responsivitas halaman.

Implementasi hak akses dikonfigurasi melalui *Role-Based Access Control* (RBAC) yang membedakan wewenang antara Admin Akademik, Kaprodi, Dosen, dan Admin Sistem. Pada tahap pra-komputasi, salah satu fitur penting adalah mengumpulkan preferensi pengajar melalui modul *Lecturer Request Dashboard*.



Gambar 4. Antarmuka *Lecturer Request Dashboard*

Gambar 4 menunjukkan bahwa dosen dapat menandai blok waktu di mana mereka tidak dapat mengajar. Input ini secara otomatis merekam sebagai batasan lunak (*soft constraints*) oleh sistem. Metode ini secara signifikan memperbaiki mekanisme lama di Departemen Teknik Elektronika Universitas Negeri Padang, yang sebelumnya hanya bergantung pada komunikasi informal yang sulit direkapitulasi.

3.2 Komputasi Algoritma Genetika dan Re-optimasi Parsial (MPP)

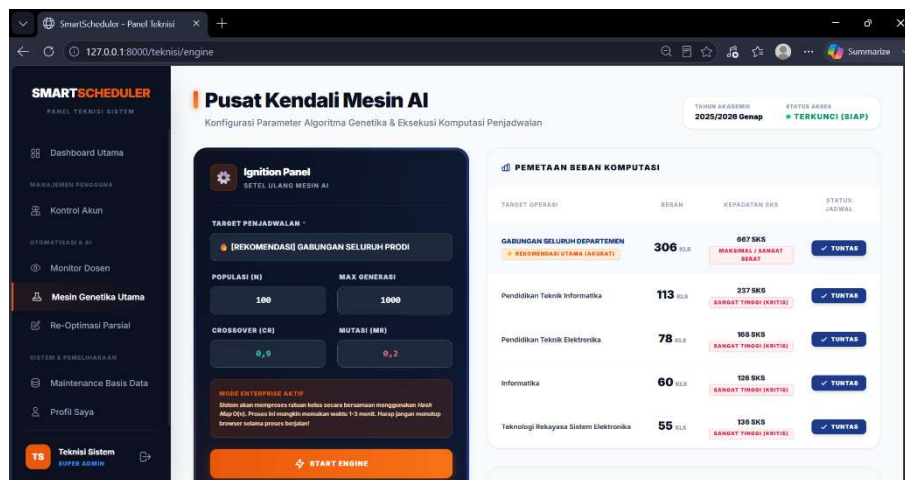
Inti komputasi dari perangkat lunak ini bertumpu pada modul Algoritma Genetika yang dieksekusi oleh Admin Sistem. Untuk transformasi data ke dalam kromosom, setiap gen diwakili dengan entitas mata kuliah, dosen, ruang, dan slot waktu. Proses pencarian jadwal optimal dinilai menggunakan fungsi kebugaran (*fitness function*) yang berorientasi pada minimasi nilai denda (*penalty*).

Model matematis yang digunakan untuk menghitung kualitas setiap individu (jadwal) diformulasikan sebagai berikut:

$$\text{Fitness} = \frac{1}{1 + \sum (Wh \cdot Hv) + \sum (Ws \cdot Sv)}$$

Keterangan:

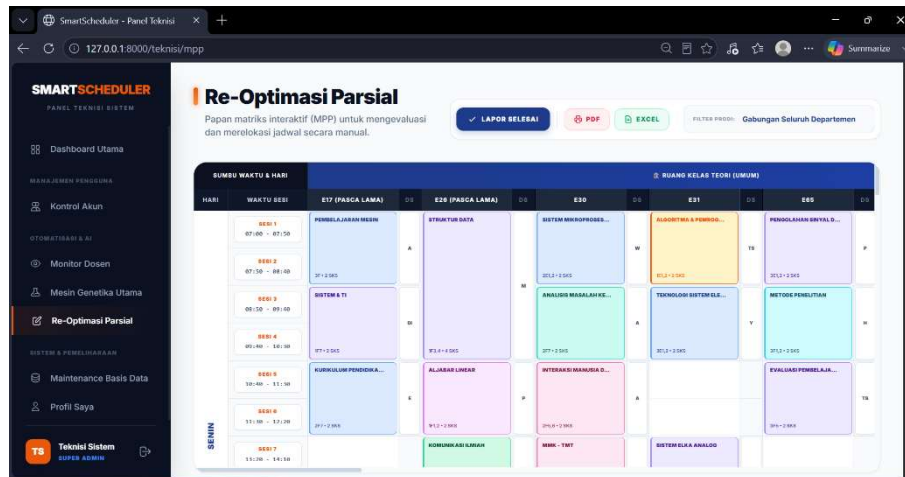
- Wh = Bobot penalti untuk pelanggaran batasan mutlak (*hard constraints*).
- Hv = Jumlah pelanggaran *hard constraints*
- Ws = Bobot penalti untuk pelanggaran batasan lunak (*soft constraints*).
- Sv = Jumlah pelanggaran *soft constraints*



Gambar 5. Antarmuka Optimasi Algoritma Genetika

Gambar 5 menunjukkan proses mutasi dan crossover dalam proses pencarian solusi. Mesin komputasi mengevaluasi ribuan kemungkinan hingga nilai penalti nol. Ini adalah kondisi zero-conflict, di mana tidak ada benturan waktu atau ruang.

Selain itu, sistem dilengkapi dengan kapabilitas resolusi dinamis melalui fitur Re-optimasi Parsial. Saat terjadi anomali mendadak di tengah semester seperti perubahan ketersediaan ruang algoritma tidak perlu mengalkulasi ulang seluruh matriks dari awal. Sistem mengadopsi penyelesaian berbasis *Minimal Perturbation Problem* (MPP), di mana komputasi hanya memodifikasi slot gen yang terdampak, sehingga menjaga stabilitas susunan jadwal kelas paralel lainnya (Phillips dkk., 2017).



Gambar 6. Antarmuka Re-optimasi Parsial

3.3 Hasil Validasi dan Evaluasi Sistem

Validasi tingkat fungsionalitas dan keberterimaan platform diukur menggunakan teknik *Black-box testing* dan *System Usability Scale* (SUS) yang melibatkan 12 responden perwakilan pengguna dari berbagai entitas akademik.

Tabel 2. Rekapitulasi Pengujian Fungsional Sistem (*User Acceptance Testing*)

Modul Pengujian	Skenario Validasi	Kriteria Keberhasilan	Tingkat Keberhasilan
Integrasi Master Data	Sinkronisasi relasi data dosen, mata kuliah, dan kapasitas ruang.	Basis data menyimpan relasi secara persisten tanpa anomali.	100%
Koleksi <i>Soft Constraint</i>	Transformasi input halangan waktu dosen ke dalam parameter	Sistem membaca <i>input</i> sebagai penalti pada <i>fitness function</i> .	100%

	algoritma.		
Optimasi Genetika	Eksekusi parameter evolusi (Populasi, <i>Crossover</i> , Mutasi).	Menghasilkan matriks jadwal mutlak bernilai <i>zero-conflict</i> .	100%
Eksekusi Re-optimasi (MPP)	Pergeseran paksa pada satu kelas akibat perubahan ketersediaan ruang.	Resolusi bentrok tercapai tanpa mengubah struktur kelas lainnya.	100%

Berdasarkan Tabel 2, pengujian UAT menunjukkan rasio kelulusan absolut (100%), yang membuktikan bahwa seluruh alur logika telah berjalan presisi. Selanjutnya, pengukuran evaluasi *usability* menggunakan instrumen SUS mencetak skor rata-rata sebesar 80,21. Merujuk pada pedoman interpretasi teoretis standar SUS, angka tersebut berada di atas ambang batas minimal kelayakan dan menempatkan sistem pada tingkat akseptabilitas *Acceptable* dengan kategori *Grade A* atau *Excellent* (Lewis, 2018; Riyadi dkk., 2020). Skor ini juga sejalan dengan parameter evaluasi sistem informasi modern yang mensyaratkan kemudahan antarmuka bagi pengguna akhir (Anggraini dkk., 2025; Huda dkk., 2023).

3.4 Pembahasan

Hasil penelitian ini secara empiris membuktikan bahwa integrasi Algoritma Genetika mampu mengatasi tantangan optimasi kombinatorial (kategori persoalan *NP-Hard*) dalam penyusunan jadwal perguruan tinggi. Ada dua temuan penting dari penerapan sistem ini yang dapat dievaluasi.

Pertama, efisiensi operasional meningkat secara signifikan. Proses penyusunan draf jadwal, yang sebelumnya membutuhkan waktu satu hingga dua minggu untuk menyelesaikan, kini direduksi menjadi hitungan menit melalui komputasi otomatis, menurut perbandingan dengan mekanisme konvensional. Kesuksesan algoritma mencapai nilai kebugaran maksimal (*zero-conflict*) sekaligus menghilangkan risiko human-error, yang sering menyebabkan tumpang tindih dalam alokasi ruang dan waktu mengajar.

Selanjutnya, skor SUS yang tinggi (80,21) memberikan implikasi positif terhadap desain interaksi sistem. Antarmuka pengguna berhasil dikemas secara ringkas dan mudah dipahami di balik arsitektur matematis Algoritma Genetika dan penyelesaian perturbasi (MPP) di bagian *back-end*. Tingkat *learnability* yang tinggi memastikan para tenaga kependidikan dapat mengoperasikan sistem otomatisasi ini tanpa memerlukan pelatihan adaptasi teknologi yang panjang. Temuan ini menegaskan bahwa otomatisasi penjadwalan tidak hanya sekadar merealisasikan akurasi komputasi yang bebas bentrok, melainkan juga harus menghadirkan pendekatan *user-centered design* untuk menjamin keberlanjutan pemakaiannya di lingkungan institusi akademik (Abdullah dkk., 2024).

4. KESIMPULAN DAN SARAN

Penelitian ini menghasilkan sebuah terobosan teknis dalam tata kelola akademik melalui digitalisasi proses penyusunan jadwal ke dalam satu ekosistem web yang terpadu. Kehadiran platform ini sukses menyinergikan seluruh alur administratif mulai dari pemetaan mata kuliah oleh Kaprodi dan perekaman preferensi waktu mengajar dosen, hingga tahap eksekusi otomatis oleh Admin Sistem yang digerakkan oleh Algoritma Genetika. Terbukti bahwa kinerja komputasi sistem ini dapat mengakomodasi seluruh parameter batasan untuk membangun matriks jadwal yang sepenuhnya bebas konflik (*zero-conflict*). Transisi teknologi ini secara langsung mengeliminasi inefisiensi durasi kerja dan risiko kelalaian manusia (*human error*) yang melekat pada penggunaan lembar kerja konvensional. Selain itu, kehadiran modul Re-optimasi Parsial berbasis *Minimal Perturbation Problem* (MPP) memberikan keluwesan bagi institusi dalam menyelesaikan perubahan jadwal dadakan secara cerdas tanpa memengaruhi stabilitas susunan kelas lainnya.

Tingkat keandalan arsitektur perangkat lunak ini telah tervalidasi secara komprehensif, dibuktikan dengan keberhasilan fungsional absolut (100%) pada *User Acceptance Testing* (UAT). Keunggulan matematis di balik layar tersebut turut ditunjang oleh desain antarmuka yang sangat ramah pengguna, tecermin dari perolehan skor *System Usability Scale* (SUS) di angka 80,21 (kategori *Excellent*). Secara keseluruhan, hasil pengujian empiris ini menegaskan bahwa sistem yang dikembangkan tidak hanya sekadar berhasil memecahkan rumitnya problem optimasi alokasi ruang dan waktu, tetapi juga

terbukti sangat praktis dan obyektif untuk diimplementasikan secara langsung oleh para staf pengguna akhir

DAFTAR REFERENSI

- Abdullah, S., dkk. (2024). Matrix-based chromosome representation in genetic algorithms for university course timetabling. *Journal of Heuristics & Scheduling*.
- Anggraini, H., Alam, R. G., & Hidayah, A. K. (2025). Evaluation of an existing system using the System Usability Scale (SUS) as a guideline for system improvement. *Jurnal Teknik Informatika*, 18(1), 122–132. <https://doi.org/10.15408/jti.v18i1.40766>
- Huda, M., dkk. (2023). System usability scale implementation in e-learning evaluation. *Journal of Educational Technology*, 15(2), 45-56.
- Lewis, J. R. (2018). The system usability scale: Past, present, and future. *International Journal of Human-Computer Interaction*, 34(7), 577–590. <https://doi.org/10.1080/10447318.2018.1455307>
- Mahlous, A. R., & Mahlous, H. (2023). Student timetabling genetic algorithm accounting for student preferences. *PeerJ Computer Science*, 9, e1200. <https://doi.org/10.7717/peerj-cs.1200>
- Mulyanto, A., & Hakim, N. F. (2025). Penerapan metode waterfall pada rancang bangun aplikasi penjualan dan manajemen stock. *ALMUISY: Journal of Al Muslim Information System*, 4(1), 10-22.
- Pamungkas, H., & Neta, F. (2025). Self-adaptive genetic algorithm (SAGA) untuk optimasi penjadwalan mata kuliah: Integrasi fuzzy logic dan cooperative coevolution. *Journal of Innovation Research and Knowledge*, 5(2), 191-200.
- Phillips, A. E., Walker, C. G., Ehr Gott, M., & Ryan, D. M. (2017). Integer programming for minimal perturbation problems in university course timetabling. *Annals of Operations Research*, 252(2), 283–304. <https://doi.org/10.1007/s10479-015-2094-z>
- Pressman, R. S., & Maxim, B. R. (2020). *Software engineering: A practitioner's approach* (9th ed.). McGraw-Hill Education.
- Riyadi, W., Irawan, I., & Istoningtyas, M. (2020). Evaluasi kegunaan aplikasi sistem informasi (SIKOJA) dengan metode system usability scale (SUS). *Processor*, 15(2), 135–144. <https://doi.org/10.33998/processor.2020.15.2.877>
- Romaguera, D., Plender-Nabas, J., Matias, J., & Austero, L. (2024). Development of a web-based course timetabling system based on an enhanced genetic algorithm. *Procedia Computer Science*, 234, 1714-1721. <https://doi.org/10.1016/j.procs.2024.03.177>
- Rosa, A. S., & Shalahuddin, M. (2018). *Rekayasa perangkat lunak terstruktur dan berorientasi objek* (Edisi Revisi). Informatika Bandung.
- Setiawati, N. L. P. L. S., Dewi, D. A. S., & Utami, N. M. C. (2023). Evaluasi usability menggunakan system usability scale (SUS). *Jurnal Teknologi Informasi*, 10(2), 157–163.
- Suria, O. (2024). Assessing financial information system usability using system usability scale (SUS) and usability metric for user experience-lite (UMUX-Lite). *Jurnal Teknologi Sistem Informasi*, 7(2), 538–547. <https://doi.org/10.32493/jtsi.v7i2.38723>
- Wahid, A. A. (2020). Analisis metode waterfall untuk pengembangan sistem informasi. *Jurnal Ilmu-ilmu Informatika dan Manajemen STMIK*, 3(1), 1-5.