

Perancangan Sistem Cerdas Penjadwalan Mata Kuliah Otomatis Berbasis Web Menggunakan Algoritma Genetika

Asrofi^{1,*}, Pol Metra¹, Bastomi Baharsyah¹

¹Sains dan Teknologi, Sistem Informasi, Universitas Islam Negeri Sulthan Thaha Saifuddin, Jambi, Indonesia

Email: ¹asrofibinsarwoto@gmail.com, ²polmetra@uinjambi.ac.id, ³bastomibaharsyah@uinjambi.ac.id

(*Email Corresponding Author: asrofibinsarwoto@gmail.com)

Received: 24 Juni 2025 | Revision: 26 Juni 2025 | Accepted: 26 Juni 2025

Abstrak

Proses penjadwalan mata kuliah di Fakultas Sains dan Teknologi UIN STS Jambi saat ini masih menggunakan metode konvensional melalui *Microsoft Excel*, mulai dari tahap penyusunan oleh program studi, validasi oleh pihak akademik, hingga tahap finalisasi. Proses konvensional ini tidak hanya memerlukan waktu yang relatif lama, tetapi juga rentan menimbulkan bentrokan jadwal antara mata kuliah, dosen, maupun penggunaan ruangan. Untuk mengatasi permasalahan tersebut, dibutuhkan sebuah sistem berbasis web yang bisa diakses secara fleksibel kapan saja oleh admin, dosen, maupun mahasiswa, serta mampu menghasilkan jadwal secara otomatis. Sistem ini dikembangkan menggunakan metode prototyping, dengan penerapan algoritma genetika sebagai solusi untuk menghindari bentrokan jadwal. Website dirancang menggunakan *framework* CodeIgniter 4. Setelah proses pengembangan selesai, dilakukan pengujian menggunakan metode *blackbox testing* dan *user acceptance testing* (UAT) dengan pendekatan skala *Likert*. Hasil dari uji kelayakan menunjukkan bahwa sistem yang dikembangkan memperoleh rata-rata persentase sebesar 94%. Berdasarkan hasil tersebut, dapat disimpulkan bahwa sistem cerdas penjadwalan mata kuliah berbasis web ini berhasil dikembangkan dan mampu membantu admin (staf akademik/program studi) dalam menghasilkan jadwal secara otomatis serta memudahkan mahasiswa dan dosen dalam mengakses informasi jadwal kuliah.

Kata Kunci: Penjadwalan Mata Kuliah, Algoritma Genetika, Prototyping, CodeIgniter 4, Website

Abstract

The course scheduling process at the Faculty of Science and Technology, UIN STS Jambi, currently uses a conventional method through *Microsoft Excel*, from the preparation stage by the study program, validation by the academic staff, to the finalization stage. This conventional process not only takes a relatively long time but is also prone to causing scheduling conflicts between courses, lecturers, and room usage. To address these issues, a web-based system is needed that can be accessed flexibly anytime by the admin, lecturers, and students, and capable of generating schedules automatically. This system is developed using the prototyping method, with the application of a genetic algorithm as a solution to avoid scheduling conflicts. The website is designed using the CodeIgniter 4 framework. After the development process is completed, testing is conducted using the black box testing method and user acceptance testing (UAT) with a *Likert* scale approach. The feasibility test results show that the developed system achieved an average percentage of 94%. Based on these results, it can be concluded that the web-based intelligent course scheduling system has been successfully developed and can assist the admin (academic staff/study program) in generating schedules automatically, as well as facilitate students and lecturers in accessing course schedule information.

Keywords: Course Scheduling, Genetic Algorithm, Prototyping, CodeIgniter 4, Website

1. PENDAHULUAN

Dunia perkuliahan merupakan tempat bertemunya setiap individu dari seluruh penjuru. Universitas Islam Negeri (UIN) Sulthan Thaha Saifuddin Jambi adalah salah satu perguruan tinggi yang berada di provinsi Jambi. Jumlah mahasiswa di UIN Jambi sangatlah banyak, terutama di Fakultas Sains dan Teknologi yang memiliki lebih dari 1.000 mahasiswa. Fakultas ini memiliki fasilitas yang mencakup 35 ruang teori dan 5 ruang laboratorium yang perlu dikelola dengan baik dan dioptimalkan agar mampu menampung seluruh kegiatan belajar mengajar, termasuk kegiatan tatap muka, praktikum, dan penugasan. Setiap tahunnya, jumlah mahasiswa dapat bertambah atau berkurang, sementara fasilitas di kampus perlu dievaluasi apakah masih layak digunakan atau tidak, termasuk ruang kelas untuk kegiatan belajar mengajar [1].

Salah satu kendala saat memberikan pelayanan ialah masalah sulitnya penyusunan jadwal perkuliahan. Jadwal perkuliahan adalah sebuah daftar yang mencakup informasi mengenai nama mata kuliah, waktu, ruang kelas, dan detail lainnya. Merancang jadwal untuk sebuah fakultas yang memiliki banyak mahasiswa bisa menjadi tugas yang rumit, sehingga seringkali muncul masalah seperti bentrokan jam, ruang kelas yang tidak tersedia, dosen yang memiliki jadwal bertabrakan untuk mata kuliah yang berbeda, serta mahasiswa yang mengikuti perkuliahan pada mata kuliah yang berbeda di waktu yang sama, dan lain sebagainya [2].

Saat ini, di Fakultas Sains dan Teknologi UIN STS Jambi pada proses penyusunan jadwal masih dilakukan secara konvensional. Setiap program studi menyusun jadwal mata kuliah masing-masing dengan menggunakan *Microsoft Excel*, yang kemudian dikirimkan ke staf akademik untuk diunggah ke Sistem Informasi Akademik (SIKAD). Meskipun *Microsoft Excel* merupakan alat yang umum digunakan dalam penyusunan jadwal, alat ini memiliki banyak keterbatasan

yang dapat mengganggu efektivitas dalam pengelolaan jadwal perkuliahan. Pengelolaan jadwal perkuliahan secara konvensional sering kali menghabiskan banyak waktu dan merupakan proses yang rumit yang seringkali menghadapi berbagai kendala. Beberapa masalah yang umum terjadi meliputi tumpang tindih antara mata kuliah, sulit menyesuaikan jadwal dosen, serta keterbatasan jumlah ruang kelas yang digunakan oleh berbagai program studi pada jam tertentu [3].

Permasalahan lain juga terdapat pada distribusi informasi jadwal kuliah, yang di mana setelah jadwal diunggah ke SIAKAD oleh staf akademik, informasi tersebut bisa diakses mahasiswa dan dosen, tetapi diharuskan untuk login terlebih dahulu. Metode ini sering kali memerlukan beberapa kali proses masuk berulang. Selain itu, akses ke SIAKAD sendiri seringkali dibatasi dan hanya tersedia pada waktu-waktu tertentu. Mengingat seluruh mahasiswa UIN Jambi juga menggunakan SIAKAD, ketika banyak pengguna mencoba mengaksesnya pada waktu yang sama, sering kali terjadi kendala teknis yang mengakibatkan gangguan akses, sehingga informasi penjadwalan tidak selalu mudah didapatkan oleh pengguna. Melihat berbagai permasalahan-permasalahan di atas, maka dirancanglah sebuah sistem cerdas penjadwalan mata kuliah otomatis berbasis web menggunakan algoritma genetika di Fakultas Sains dan Teknologi. Sistem ini diharapkan dapat membantu proses pengelolaan penjadwalan mata kuliah agar menjadi lebih terorganisir dan meminimalisir konflik jadwal, serta mempermudah dosen dan mahasiswa dalam mengakses informasi mengenai penjadwalan mata kuliah di fakultas.

Algoritma genetika dipilih karena kemampuannya dalam mencari solusi optimal pada permasalahan yang kompleks, seperti penjadwalan mata kuliah. Algoritma ini akan melakukan pencarian terstruktur yang terinspirasi oleh mekanisme seleksi alam dan prinsip kerja genetika dalam kehidupan [4]. Pada algoritma ini juga, solusi-solusi terbaik akan dipilih dan dikombinasikan untuk mendapatkan solusi yang lebih optimal. Dengan pendekatan ini, sistem dapat menghasilkan jadwal yang lebih efisien dibandingkan dengan metode sebelumnya.

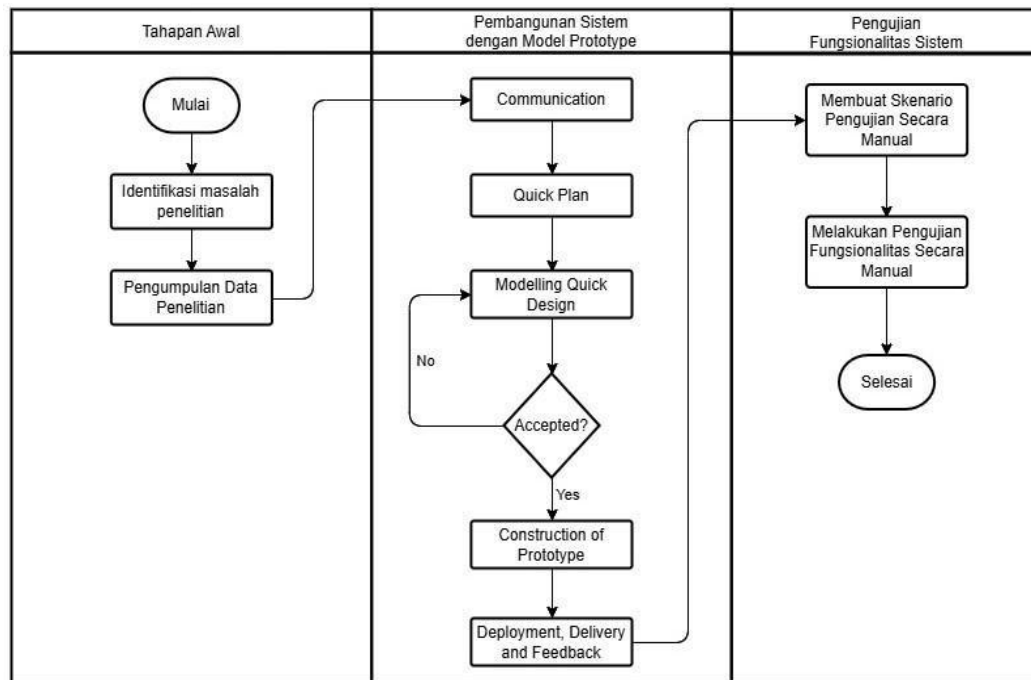
Sistem ini dibangun menggunakan metode *prototyping*. Metode ini merupakan suatu metode pada rekayasa perangkat lunak yang secara langsung menunjukkan cara pengembangan aplikasi dengan lebih sederhana. Salah satu keuntungan dari metode ini adalah kemudahan dalam menentukan kebutuhan pengguna [5]. Untuk memahami alur sistem dan memperlancar proses perancangan, penulis memanfaatkan *Unified Modelling Language* (UML) pada desain sistem, yang mencakup beberapa diagram seperti *Use Case Diagram*, *Activity Diagram*, dan *Class Diagram*. Dalam pengembangan sistem ini, penulis menggunakan bahasa pemrograman PHP serta *database* MySQL.

Sejumlah penelitian sebelumnya telah membahas pengembangan sistem penjadwalan serupa. Penelitian oleh Fitriani et al. menggunakan *algoritma genetika* untuk menyusun jadwal kuliah secara otomatis, namun pengembangan dilakukan dengan metode *waterfall* dan belum melibatkan pengguna akhir dalam proses pengujian [6]. Rachel juga menerapkan *algoritma genetika* dalam sistem penjadwalan, namun sistem yang dikembangkan tidak berbasis *web* dan akses pengguna masih terbatas [7]. Jeriko et al. membangun sistem berbasis *web* menggunakan *framework* CodeIgniter dan menerapkan metode *prototyping*, namun pengujian hanya dilakukan secara internal dan belum melibatkan dosen maupun mahasiswa [8]. Penelitian oleh Andika et al. telah menerapkan teknik *rank-based selection* dalam *algoritma genetika*, tetapi pengujian sistem belum menggunakan pengujian kepuasan pengguna [9]. Sementara itu, Rohim et al. mengembangkan sistem penjadwalan berbasis multivariat dengan pendekatan *genetic algorithm*, namun belum menggunakan pengujian kepuasan pengguna [10].

2. METODOLOGI PENELITIAN

2.1 Tahapan Penelitian

Tahapan penelitian yaitu langkah sistematis yang harus diikuti guna memastikan penelitian berjalan sesuai dengan yang direncanakan. Tahapan penelitian meliputi proses dari awal hingga akhir penelitian. Adapun tahapannya adalah sebagai berikut:



Gambar 1. Tahapan Penelitian

- a. **Identifikasi Masalah Penelitian**
Tahap pertama dilakukan untuk mengidentifikasi masalah dalam proses penjadwalan mata kuliah di fakultas. Proses ini dilakukan melalui observasi dan wawancara dengan staf prodi, staf akademik, dosen, dan mahasiswa guna memahami kendala pada sistem penjadwalan yang ada.
- b. **Pengumpulan Data Penelitian**
Tahap kedua dilakukan dengan mengumpulkan data yang dibutuhkan untuk merancang sistem. Data meliputi daftar dosen dan waktu mengajar, mata kuliah, pembagian kelas, kapasitas ruang, dan jam kuliah yang tersedia. Peneliti juga mewawancarai pihak akademik dan prodi untuk memahami alur kerja jadwal saat ini dan harapan mereka terhadap sistem baru.
- c. **Communication**
Langkah selanjutnya adalah menjalin komunikasi dengan calon pengguna sistem. Dalam tahap ini, peneliti berdiskusi untuk memahami apa saja fitur yang dibutuhkan, bagaimana sistem akan digunakan nantinya, serta masalah-masalah teknis yang mungkin muncul. Komunikasi ini penting sistem yang dibangun mampu memenuhi kebutuhan pengguna secara optimal.
- d. **Quick Plan**
Pada tahap ini, peneliti menyusun perencanaan awal pengembangan sistem. Rencana ini mencakup teknologi yang digunakan (seperti PHP, CodeIgniter 4, dan MySQL), arsitektur sistem, modul-modul penting seperti input data, pemrosesan algoritma genetika, dan output jadwal. Peneliti juga menetapkan batasan sistem serta target waktu untuk menyelesaikan setiap bagian pada metode prototipe.
- e. **Modelling Quick Design**
Kemudian peneliti membuat rancangan awal sistem. Ini mencakup desain tampilan antarmuka, struktur database, serta bagaimana alur sistem penjadwalan yang akan dirancang. Dalam tahap ini, peneliti akan menggunakan *Unified Modelling Language (UML)* yang terdiri dari *use case diagram*, *activity diagram*, dan *class diagram*.
- f. **Construction of Prototype**
Setelah rancangan awal pada modelling quick design disetujui, peneliti mulai membuat prototype yang kemudian akan diimplementasikan ke tahap pengkodean. Peneliti mulai membangun sistem secara utuh. Semua fitur penting mulai dari pengolahan data hingga algoritma penjadwalan diterapkan secara menyeluruh.
- g. **Deployment, Delivery, and Feedback**
Setelah sistem selesai dibangun, sistem diuji coba langsung oleh pengguna untuk melihat apakah sudah berfungsi dengan baik. Peneliti menerima masukan dari pengguna mengenai elemen-elemen yang memerlukan penyempurnaan, baik dari aspek visual, proses kerja sistem, maupun hasil penjadwalannya.
- h. **Membuat Skenario Pengujian Manual**

Peneliti menyusun skenario pengujian berdasarkan fungsionalitas (*black box testing*) sistem yang telah dibangun. Skenario ini mencakup pengujian terhadap input data, proses generate jadwal, dan hasil akhir penjadwalan, apakah sudah memenuhi kriteria tanpa bentrok jadwal dosen, ruang, dan waktu.

i. Melakukan Pengujian Fungsionalitas Secara Manual

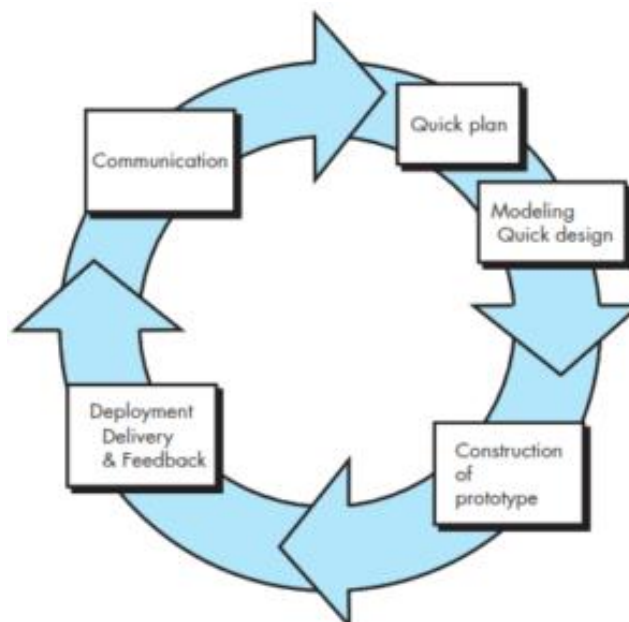
Setiap skenario yang telah disusun akan diuji secara bertahap guna memastikan bahwa sistem berfungsi dengan baik. Pengujian ini juga memiliki tujuan guna menilai kemampuan sistem dalam men-generate jadwal yang tepat dan bebas dari bentrok. Lalu, peneliti akan meminta responden untuk mengisi angket *user acceptance testing* (UAT) dengan skala *Likert*.

2.2 Analisis dan Perancangan

Berikut ini akan dibahas analisis serta perancangan sistem yang akan diimplementasikan dalam bentuk aplikasi.

a. Metode Pengembangan Perangkat Lunak

Penelitian ini menggunakan metode *prototyping*, yaitu dengan membuat model awal (*prototype*) yang mewakili fitur utama sistem. *Prototype* ini memungkinkan pengguna dan peneliti melakukan uji coba serta memberikan masukan langsung. Adapun alurnya dapat dilihat pada gambar sebagai berikut.



Gambar 2. Kerangka *Prototype*

1. Communication (Komunikasi)
Proses pertama, pengembang berkomunikasi kepada pemangku kepentingan (*stakeholders*) untuk memahami kebutuhan dan harapan mereka. Diskusi ini melibatkan pengumpulan informasi tentang fungsi yang diinginkan, batasan, dan preferensi pengguna.
2. Quick Plan (Perencanaan Cepat)
Setelah memahami kebutuhan, selanjutnya dilakukan perencanaan secara cepat. Disusun rencana dasar yang mencakup lingkup proyek, waktu yang dibutuhkan, dan sumber daya yang diperlukan untuk pembuatan *prototype*.
3. Modelling Quick Design (Pemodelan Desain Cepat)
Pada tahap ini, desain awal *prototype* dibuat. Ini mencakup model yang menggambarkan fitur dan fungsi utama dari sistem yang akan dikembangkan yang di mana dapat berupa diagram, *wireframe*, atau *mockup*.
4. Construction of Prototype (Konstruksi Prototype)
Setelah desain disetujui, maka selanjutnya *prototype* dibangun. Ini adalah versi awal dari produk yang memungkinkan pengembang dan pemangku kepentingan untuk melihat dan berinteraksi dengan solusi yang diusulkan.
5. Deployment, Delivery & Feedback (Penyebaran, Pengiriman, dan Umpan Balik)

Prototype yang telah dibangun kemudian disampaikan kepada pemangku kepentingan untuk diuji. Pengembang mengumpulkan umpan balik atau *feedback* untuk melakukan perbaikan dan penyesuaian sebelum melanjutkan ke pengembangan produk akhir.

2.3 Algoritma Genetika

Algoritma Genetika (AG) merupakan teknik optimasi yang terinspirasi dari evolusi biologis dan genetika [11]. Algoritma ini digunakan untuk mencari solusi optimal atau mendekati optimal dalam suatu masalah yang kompleks. Penulis memilih menggunakan algoritma genetika sebagai solusi dalam pemetaan jadwal kuliah dan lokal karena tahapan-tahapan dalam metode ini dinilai sangat sesuai dan relevan dalam mengoptimalkan jadwal secara efisien dengan mempertimbangkan berbagai kendala, seperti ketersediaan dosen, ruang kelas, dan konflik waktu. Adapun tahapan-tahapan algoritma genetika menurut Eka Yulia Sari dkk. yaitu inisialisasi, menghitung fitness, seleksi, crossover, dan mutasi [12].

2.3.1 Inisialisasi Populasi

Langkah pertama yang penting dalam membangun sistem penjadwalan mata kuliah adalah merancang struktur kromosom yang akan digunakan dalam populasi. Perancangan ini memiliki peran penting karena menentukan bagaimana data diproses dan seberapa efektif algoritma genetika dalam menghasilkan jadwal yang optimal. Adapun susunan kromosom ini terdiri dari beberapa elemen utama, yaitu:

- a. Mata kuliah (MK)
- b. Dosen (D)
- c. Kelas (K)
- d. Ruang kelas (R)
- e. Hari (H)
- f. Jam kuliah (J)

Sehingga, susunan kromosom dalam sistem ini dapat direpresentasikan sebagai $\langle \text{MK}, \text{D}, \text{K}, \text{R}, \text{H}, \text{J} \rangle$. Panjang suatu kromosom ditentukan berdasarkan jumlah mata kuliah yang harus dijadwalkan dalam satu semester. Setiap gen dalam kromosom akan menyimpan informasi mengenai waktu dan tempat perkuliahan untuk satu mata kuliah tertentu. Sebagai contoh dalam proses inisialisasi populasi adalah sebagai berikut.

Tabel 1. Tabel Individu

Kromosom	Gen 1			Gen 2	Gen 3	
	Matakuliah	Dosen	Kelas	Ruang	Hari	Jam/Sesi
1						
2						
3						

Dalam proses pembentukan populasi awal, data sampel diambil dari tabel distribusi mata kuliah, tabel distribusi waktu perkuliahan (hari dan jam/sesi), dan tabel distribusi ruangan. Kombinasi data ini dipilih secara acak untuk membentuk susunan awal populasi. Proses pemilihan acak ini bertujuan untuk memastikan keberagaman dalam populasi awal, yang merupakan prinsip dasar algoritma genetika agar dapat menemukan solusi optimal melalui seleksi dan evolusi. Misalkan telah ditentukan populasi awal secara acak sebanyak 4 individu sebagai berikut:

Tabel 2. Tabel Individu Ke-1 Pembentukan Populasi Awal

ID Mata Kuliah	ID Dosen	ID Kelas	Ruang	ID Hari	ID Jam	
MK01	D01	KS11A	A	01	03	[1,1,1,3]
MK02	D02	KS11B	B	01	01	[2,2,1,1]
MK03	D02	KS13A	C	01	01	[3,3,1,1]

Keterangan Penalti : Dosen yang sama mengajar di waktu yang sama = Penalti +1

Tabel 3. Tabel Individu Ke-2 Pembentukan Populasi Awal

ID Mata Kuliah	ID Dosen	ID Kelas	Ruang	ID Hari	ID Jam	
MK01	D01	KS11A	A	02	01	[1,1,2,1]
MK02	D02	KS11B	A	02	01	[2,1,2,1]
MK03	D02	KS13A	B	01	02	[3,2,1,2]

Keterangan Penalti : Ruang bentrok di waktu yang sama = Penalti +1

Tabel 4. Tabel Individu Ke-3 Pembentukan Populasi Awal

ID Mata Kuliah	ID Dosen	ID Kelas	Ruang	ID Hari	ID Jam	
MK01	D01	KS11A	B	01	01	[1,2,2,1]
MK02	D02	KS11B	B	01	01	[2,2,2,1]
MK03	D03	KS13A	C	03	02	[3,2,1,2]

Keterangan Penalti : Ruang bentrok di waktu yang sama = Penalti +1

Tabel 5. Tabel Individu Ke-4 Pembentukan Populasi Awal

ID Mata Kuliah	ID Dosen	ID Kelas	Ruang	ID Hari	ID Jam	
MK01	D02	KS11A	A	02	02	[1,1,2,2]
MK02	D02	KS11B	B	01	02	[2,2,1,2]
MK03	D02	KS13A	B	01	02	[3,2,1,2]

Keterangan Penalti : Dosen bentrok mengajar di waktu yang sama dan ruang yang sama = Penalti +2

2.3.2 Evaluasi Fitness

Setelah dibentuk populasi awal, selanjutnya setiap individu akan dievaluasi menggunakan fungsi fitness untuk mengukur kualitas solusi yang dihasilkan. Fungsi *fitness* ini akan menilai seberapa baik setiap solusi menyelesaikan masalah yang diberikan. Adapun rumus untuk menghitung *fitness* adalah sebagai berikut.

$$Fitness = \frac{1}{(1+Penalti)}$$

Penalti diberikan jika terjadi bentrok jadwal, seperti:

- Dosen mengajar di jam yang sama.
- Ruang digunakan di jam yang sama.
- Pada satu kelas memiliki dua mata kuliah pada jam yang sama.

Berdasarkan populasi awal tadi, maka perhitungan *fitness*-nya adalah sebagai berikut.

- Individu ke-1: Penalti = 1 (dosen sama mengajar di jam yang sama).
Fitness = $1/(1+1) = 0,5$
- Individu ke-2: Penalti = 1 (ruang bentrok di waktu yang sama).
Fitness = $1/(1+1) = 0,5$
- Individu ke-3: Penalti = 1 (ruang bentrok di waktu yang sama).
Fitness = $1/(1+1) = 0,5$
- Individu ke-4: Penalti = 2 (dosen sama mengajar di waktu dan ruang yang sama).
Fitness = $1/(1+2) = 0,33$

2.3.3 Seleksi

Seleksi dilakukan dengan menggunakan pendekatan *rank selection*, di mana individu dengan *fitness* yang lebih tinggi memiliki peringkat lebih tinggi dan peluang lebih besar untuk dipilih untuk menjadi induk pada proses selanjutnya.

Berdasarkan perhitungan *fitness*, maka individu ke-1 adalah peringkat 4, individu ke-2 adalah peringkat ke-3, individu ke-3 adalah peringkat 2, dan individu ke-4 adalah peringkat 1.

- Pengurutan berdasarkan *fitness*
 - Individu 1, 2, 3: Fitness 0,5 (penalti 1) merupakan peringkat 4, 3, dan 2.
 - Individu ke-4: Fitness 0,33 (penalti 2) merupakan peringkat 1.
 - Total peringkat: $4+3+2+1 = 10$.
- Membuat rentang kumulatif
 - Pengurutan individu dari peringkat terendah:
 - Individu ke-4: Peringkat 1, Kumulatif 1, Rentang 0-1.
 - Individu ke-3: Peringkat 2, Kumulatif $1+2 = 3$, Rentang 2-3.
 - Individu ke-2: Peringkat 3, Kumulatif $3+3 = 6$, Rentang 4-6.
 - Individu ke-1: Peringkat 4, Kumulatif $6+4 = 10$, Rentang 7-10.
- Melakukan *randomize* untuk menghasilkan nilai acak (0 sampai 9, Sesuai total peringkat = 10)

- Nilai: [2, 6, 8, 3].
- Pemetaan:
 - Nilai 2 masuk ke rentang 2-3 = Individu ke-3
 - Nilai 6 masuk ke rentang 4-6 = Individu ke-2
 - Nilai 8 masuk ke rentang 7-10 = Individu ke-1
 - Nilai 3 masuk ke rentang 2-3 = Individu ke-3
- d) Berdasarkan hasil pemetaan tadi, maka induk terpilih yaitu individu ke-3, individu ke-2, individu ke-1, dan individu ke-3.
- e) Memasangkan induk terpilih untuk proses *crossover*
 - Pasangan 1: Individu ke-3 x Individu ke-2
 - Pasangan 2 : Individu ke-1 x Individu ke-3

2.3.4 Crossover

Crossover adalah proses menggabungkan dua individu dari induk atau *parent* untuk menghasilkan individu baru (anak). Tujuannya adalah untuk menciptakan solusi yang lebih baik dari yang sebelumnya. Pada tahap ini penulis menggunakan *two-point crossover*, di mana dua titik dipilih secara acak dan gen antara titik tersebut dipertukarkan. Adapun proses *crossover* adalah sebagai berikut:

- a) Memilih dua titik potong antar *array*, kemudian menukar *array* di antara titik yang telah ditentukan.
- b) Menentukan nilai probabilitas *crossover*, misalnya probabilitas = 0,85 (85%). Selanjutnya *digenerate* 4 nilai secara acak yang nilainya diantara 0% sampai 100% perpasangan. Jika semua < 85%, *crossover* terjadi; jika ada $\geq 85\%$, maka induk disalin.
- c) Melakukan persilangan pada pasangan 1: Individu ke-3 x Individu ke-2:
 - Kromosom:
 - o Individu ke 3: [1, 2, 2, 1], [2, 2, 2, 1], [3, 2, 1, 2].
 - o Individu ke 1: [1, 1, 2, 1], [2, 1, 2, 1], [3, 2, 1, 2].
 - *Roulette*: [43%, 53%, 69%, 80%] (semua < 85%, *crossover*).
 - *Cut Points*: Indeks 1 dan 2 (antara *array* 0-2).
 - Menukar *array* (MK02):
 - o *Offspring* 1: [1, 2, 2, 1] (MK01 dari Individu ke-2), [2, 1, 2, 1] (MK02 dari Individu ke-1), [3, 2, 1, 2] (MK03 dari Individu ke-2).
 - o *Offspring* 2: [1, 1, 2, 1] (MK01 dari Individu ke-1), [2, 2, 2, 1] (MK02 dari Individu ke-2), [3, 2, 1, 2] (MK03 dari Individu ke-1).
- d) Melakukan persilangan pada pasangan 2: Individu ke-1 x Individu ke-3
 - Kromosom:
 - o Individu ke-1: [1, 1, 1, 3], [2, 2, 1, 1], [3, 3, 1, 1].
 - o Individu ke-3: [1, 2, 2, 1], [2, 2, 2, 1], [3, 2, 1, 2].
 - *Roulette*: [73%, 33%, 19%, 89%] (ada 89% $\geq 85\%$, tidak *crossover*).
 - Hasil:
 - o *Offspring* 3: Salinan Individu ke-3
 - o *Offspring* 4: Salinan Individu ke-1

Berdasarkan proses *crossover* yang telah dilakukan, maka didapatkan populasi baru berdasarkan *offspring* yang dihasilkan.

Tabel 6. Tabel Hasil Crossover

Kromosom	MK01	MK02	MK03
0 (<i>Offspring</i> 1)	[MK01, D01, KSI1A, B, 2, 1]	[MK02, D02, KSI1B, A, 2, 1]	[MK03, D02, KSI1A, A, 1, 2]
1 (<i>Offspring</i> 2)	[MK01, D01, KSI1A, A, 2, 1]	[MK02, D02, KSI1B, B, 2, 1]	[MK03, D02, KSI1A, A, 1, 2]
2 (<i>Offspring</i> 3)	[MK01, D01, KSI1A, B, 2, 1]	[MK02, D02, KSI1B, A, 2, 1]	[MK03, D03, KSI1A, C, 1, 2]
3 (<i>Offspring</i> 4)	[MK01, D01, KSI1A, A, 1, 1]	[MK02, D02, KSI1B, B, 1, 1]	[MK03, D02, KSI1A, C, 1, 1]

2.3.5 Mutasi

Mutasi dilakukan dengan mengganti nilai gen dalam kromosom menggunakan nilai baru untuk menghasilkan individu baru. Meskipun hampir serupa dengan *crossover* yang melibatkan pertukaran data antar individu dalam populasi, pada mutasi pergantian data dilakukan secara acak, bukan berasal dari individu lain [13].

Mutasi bertujuan untuk menjaga keberagaman genetik dalam populasi dan mencegah algoritma genetika terjebak pada solusi lokal (*local optimum*). Dengan adanya mutasi, kemungkinan munculnya solusi yang lebih baik tetap terbuka,

karena individu baru yang dihasilkan memiliki karakteristik yang tidak dimiliki oleh individu sebelumnya. Proses mutasi biasanya dilakukan dengan probabilitas tertentu (*mutation rate*), sehingga tidak semua gen mengalami perubahan dalam setiap generasi.

Adapun proses cara kerja mutasi ini adalah sebagai berikut:

- a) Untuk setiap kromosom (4 kromosom), digenerate 1 nilai acak dalam persentase (0%-100%) sebagai *roulette*: [83%, 36%, 64%, 76%].
- b) Ditentukan probabilitas mutasinya, misalnya probabilitas mutasi = 0,4 (40%). Jika *roulette* < 40%, mutasi terjadi pada kromosom tersebut.
- c) Contohnya:
 - Kromosom 1: 36% < 40%, mutasi terjadi (*array* acak, MK03, diganti ke [3, 1, 2, 1]).
 - Kromosom 0, 2, 3: $\geq 40\%$, tidak dimutasi.
- d) Probabilitas
 - Probabilitas mutasi adalah 0,4 per kromosom. Setiap kromosom memiliki kemungkinan 40% untuk mengalami mutasi, dan peluang ini tidak dipengaruhi oleh kromosom lain.
 - *Roulette* langsung dibandingkan dengan 40% untuk menentukan mutasi.

Setelah individu dimutasi, selanjutnya dilakukan iterasi atau perulangan dari langkah 2 sampai 5 hingga ditemukan solusi yang optimal dengan nilai *fitness* = 1 atau mendekati, yang artinya tidak ada penalti atau bentrok.

3. HASIL DAN PEMBAHASAN

3.1 Analisis Sistem Berjalan

Analisis sistem yang sedang berjalan bertujuan untuk menggambarkan alur proses serta aktivitas-aktivitas yang berlangsung dalam sistem penjadwalan mata kuliah di Fakultas Sains Teknologi. Berikut merupakan gambaran alur proses sistem yang sedang berjalan.

- a. Staf prodi mengumpulkan dan mencatat data dosen, kelas, matakuliah, hari perkuliahan, dan jam perkuliahan ke dalam file *excel*.
- b. Staf akademik melakukan pembagian lokal/ruang yang dapat digunakan pada setiap program studi.
- c. Staf prodi membuat jadwalnya masing-masing pada setiap prodinya.
- d. Setelah staf prodi membuat jadwal masing-masing prodi, selanjutnya diserahkan kepada staf akademik untuk divalidasi, kemudian publikasi jadwal kuliah di SIAKAD.
- e. Setelah jadwal kuliah dipublikasi, mahasiswa, dan dosen dapat melihat jadwal kuliah yang telah dipublikasi oleh staf akademik.

3.2 Analisis Permasalahan

Penulis mengidentifikasi berbagai permasalahan yang ada di sistem penjadwalan mata kuliah yang sekarang masih digunakan agar dapat memecahkan solusi berdasarkan permasalahan yang ada. Permasalahan-permasalahan tersebut yaitu sebagai berikut.

Tabel 7. Tabel Identifikasi Masalah

Kondisi	Masalah	Penyebab
Penyusunan jadwal mata kuliah masih dibuat menggunakan metode konvensional dengan menggunakan Microsoft Excel.	Penyusunan jadwal rentan terjadi kesalahan data dan adanya bentrokan jadwal.	Belum adanya sistem otomatis yang dapat menghindari adanya bentrok jadwal.
Penyusunan jadwal memakan waktu yang lama.	Penyusunan jadwal memakan waktu lama karena menyesuaikan dengan ketersediaan ruang kelas yang terbatas.	Belum adanya sistem yang dapat mengoptimalkan penggunaan ruang kelas secara efisien.
Informasi jadwal hanya dapat diakses melalui SIAKAD setelah diunggah oleh staf akademik.	Ketersediaan akses jadwal kurang fleksibel karena harus <i>login</i> ke dalam SIAKAD untuk melihat jadwal.	Proses login ke SIAKAD sering mengalami kendala, dan sistem belum menyediakan akses jadwal yang lebih fleksibel bagi pengguna.

3.3 Uraian Pemecahan Masalah

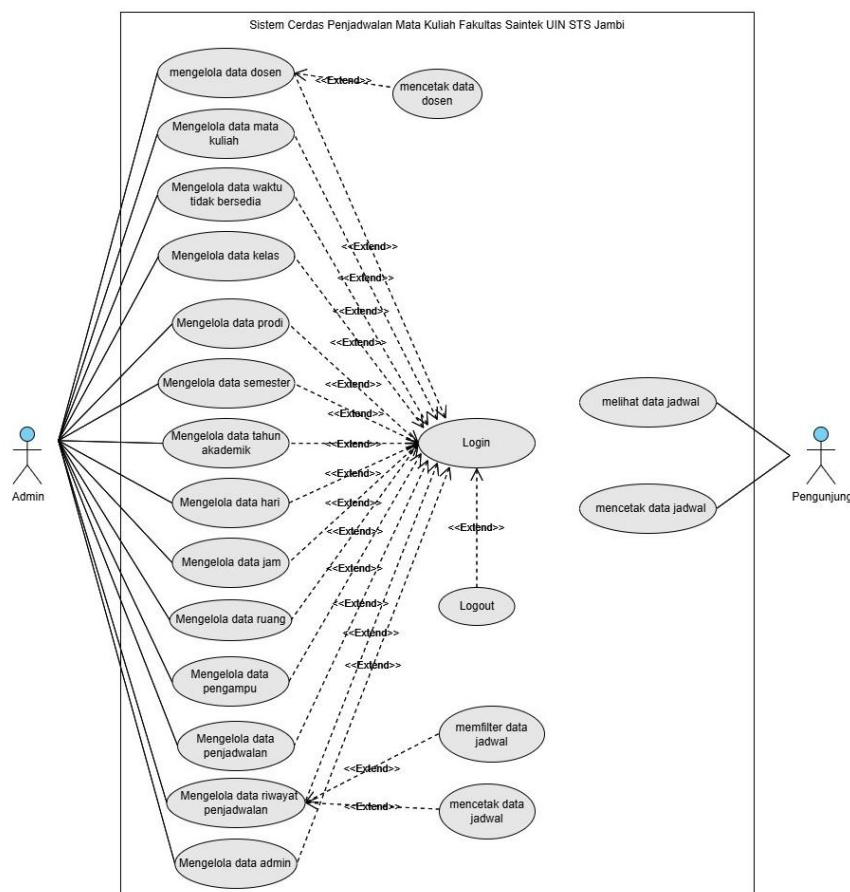
Berdasarkan identifikasi permasalahan yang ada pada sistem yang lama, penulis menguraikan pemecahan masalah dengan merancang sistem penjadwalan menggunakan Algoritma Genetika. Sistem ini akan secara otomatis menghasilkan jadwal yang optimal dengan mempertimbangkan beberapa faktor, seperti ketersediaan dosen, kapasitas ruang kelas, dan waktu perkuliahan. Dengan penggunaan algoritma ini, bentrokan jadwal dapat dihindari dan staf prodi tidak perlu menyusun jadwal secara konvensional yang rentan terhadap kesalahan. Selain itu, untuk meningkatkan kemudahan akses informasi, sistem ini juga akan menyediakan jadwal yang dapat diakses secara langsung melalui web tanpa perlu login ke SIAKAD. Fitur ini bertujuan agar mahasiswa dan dosen dapat melihat jadwal kapan saja tanpa harus menghadapi kendala login atau keterbatasan waktu akses.

3.4 Perancangan Sistem

Metode perancangan sistem yang digunakan untuk perancangan sistem yaitu *Unified Modelling Language (UML)*. Fungsi UML yaitu untuk memvisualisasikan, merancang, serta mendokumentasikan sistem perangkat lunak secara grafis. Dalam pemodelan UML ini, peneliti menggunakan 3 diagram yang meliputi *Use Case Diagram*, *Activity Diagram*, dan *Class Diagram*. Selain itu, peneliliti juga menggunakan *wireframe* sebagai rancangan awal tampilan dari sistem.

a. Use Case Diagram

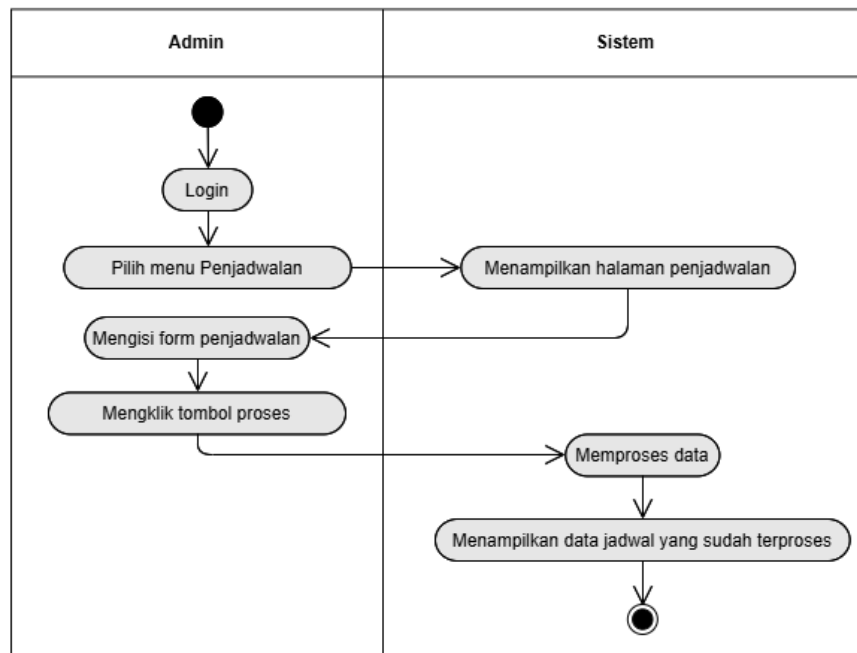
Use Case Diagram menunjukkan berbagai interaksi antara aktor dan sistem yang akan dirancang. Pada penelitian ini, terdapat 2 aktor utama yang memiliki akses ke sistem yang akan dirancang, yaitu admin yang mengelola jadwal, dan user meliputi mahasiswa dan dosen yang dapat mengakses jadwal mata kuliah yang telah disusun.



Gambar 3. Use Case Diagram

b. Activity Diagram

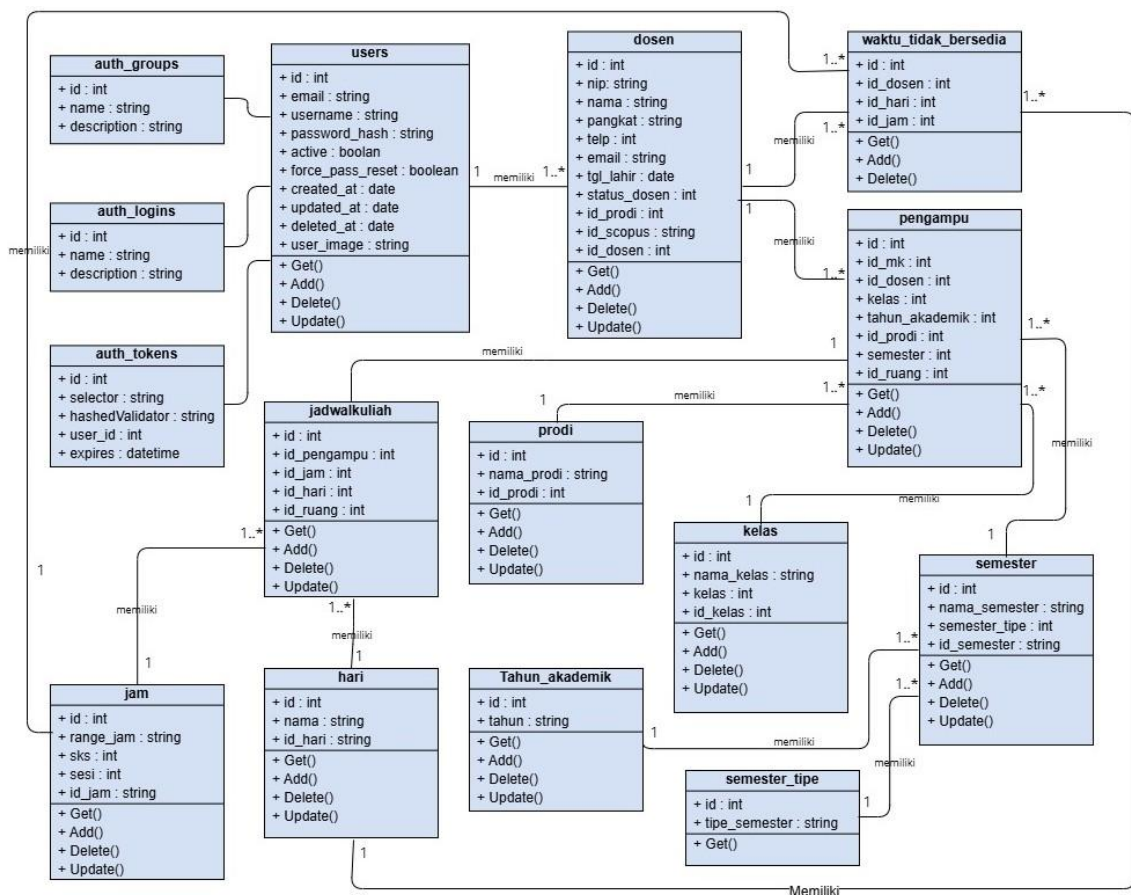
Diagram aktivitas ini berperan penting dalam menggambarkan aktivitas secara sistematis atau alur proses yang terjadi pada sistem penjadwalan. Diagram ini akan memvisualisasikan setiap langkah yang harus dilakukan oleh pengguna maupun sistem dalam mengelola jadwal perkuliahan. Berikut merupakan *activity diagram*-nya.



Gambar 4. Activity Diagram Memproses Data Penjadwalan

c. Class Diagram

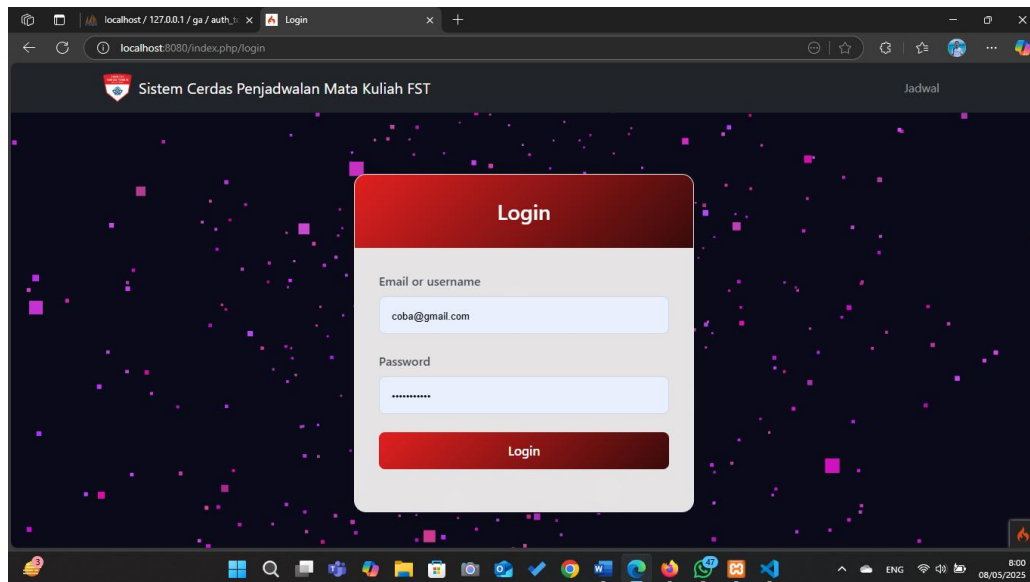
Class diagram ini dirancang dengan tujuan membantu peneliti dalam memahami struktur dan hubungan tabel dalam sistem supaya mempermudah proses pengembangan dan pemeliharaan sistem. Diagram ini akan memvisualisasikan kelas-kelas yang ada pada sistem beserta atribut, metode, dan hubungan antar kelas.



Gambar 5. Class Diagram

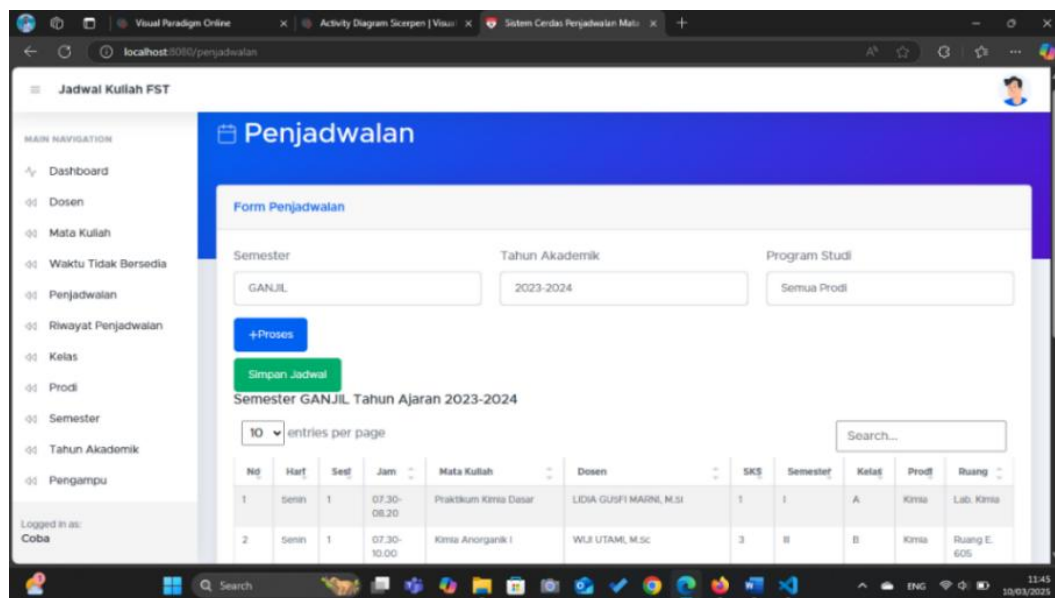
3.5 Implementasi Sistem

Setelah desain antarmuka selesai, desain tersebut kemudian diimplementasikan ke dalam pengkodean sistem. Pada proses pengkodean, digunakan bahasa pemrograman PHP dengan *framework* CodeIgniter 4, *database* MySQL, dan juga *bootstrap* sebagai *framework* CSS. Adapun hasil implementasinya ditampilkan sebagai berikut.



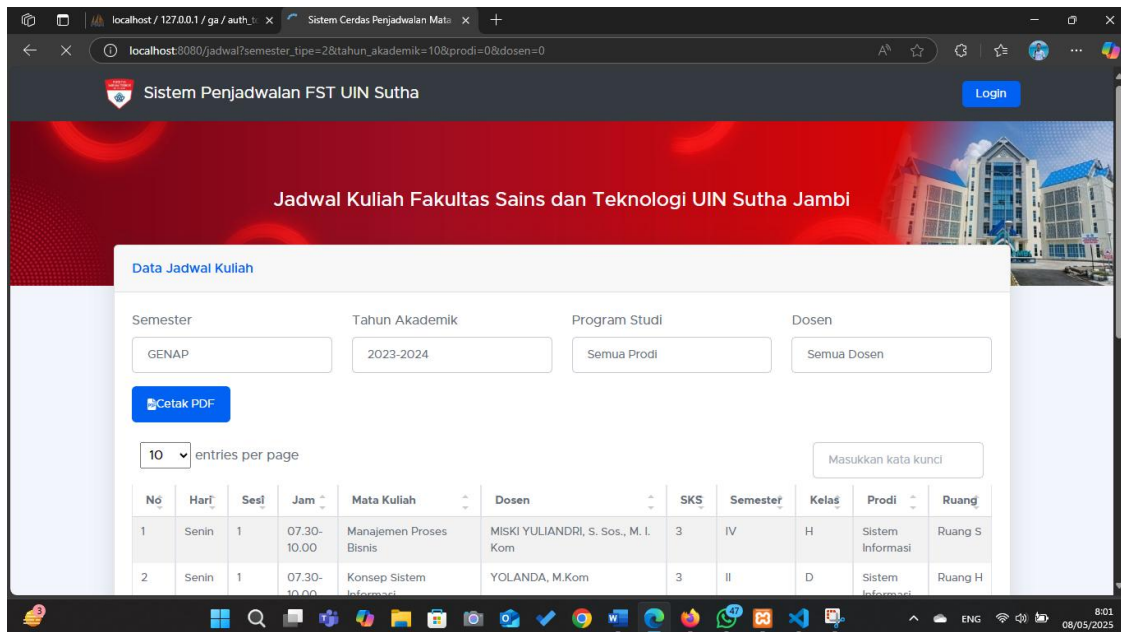
Gambar 6. Halaman *Login* Admin

Gambar di atas merupakan halaman yang digunakan untuk autentikasi pengguna sebelum mengakses sistem. Admin perlu menginputkan *username* dan *password* untuk mendapatkan akses, dan setelah login berhasil, sistem akan menampilkan halaman *dashboard*.



Gambar 7. Halaman Penjadwalan (Admin)

Gambar di atas adalah halaman penjadwalan yang merupakan fitur utama sistem yang digunakan untuk menghasilkan dan mengelola jadwal mata kuliah otomatis menggunakan Algoritma Genetika.



Gambar 8. Halaman Jadwal (*Public*)

Gambar di atas merupakan desain antarmuka halaman yang bisa diakses oleh mahasiswa dan dosen ketika ingin melihat, memfilter, dan mencetak data jadwal mata kuliah yang telah diproses oleh admin tanpa harus *login* ke dalam sistem.

3.6 Pengujian Sistem

Pengujian sistem merupakan langkah yang krusial guna menjamin bahwa sistem yang dikembangkan berfungsi dengan baik serta memenuhi kebutuhan serta ekspektasi pengguna akhir yang diharapkan. Pada pengujian sistem ini dilakukan dengan menggunakan metode *Black-Box Testing* untuk menguji fungsionalitas dari sistem yang dibuat dan juga UAT (*User Acceptance Testing*) guna menguji apakah sistem yang dirancang sudah bisa diterima oleh pengguna atau belum.

a. Pengujian *Black-Box Testing*

Pengujian *black box testing* dilakukan untuk menguji fungsi-fungsi sistem tanpa harus membuka kode sumber. Pada metode ini, digunakan pendekatan *equivalence partitioning* yang di mana pengujian akan dilakukan dengan cara mempartisi data-data masukan menjadi nilai valid (kondisi berhasil) dan invalid (kondisi gagal). Hasil pengujian dari fungsionalitas sistem ini menunjukkan bahwa semua fungsi berjalan sesuai dengan yang diharapkan.

b. Pengujian *User Acceptance Testing (UAT)*

Pengujian UAT dilakukan pada tahap akhir pengujian ketika sistem telah siap digunakan. Tujuan utama dari pengujian ini adalah memastikan bahwa perangkat lunak yang dikembangkan benar-benar dapat memenuhi kebutuhan pengguna [14]. Fokus utama dari pengujian ini ialah untuk memverifikasi bahwa sistem yang dibuat dapat berfungsi sesuai dengan keinginan dan harapan pengguna. Apabila sistem yang dirancang sudah dapat diterima oleh pengguna akhir, selanjutnya sistem dapat diimplementasikan di lingkungan. Pengujian UAT ini dilakukan dengan melibatkan pengguna akhir seperti mahasiswa, dosen, staf akademik, dan staf prodi di Fakultas Sains dan Teknologi yang nantinya akan diberikan beberapa pertanyaan dengan menggunakan 5 skala pada Skala *likert*. Menurut [15] skala *likert* adalah teknik pengukuran yang digunakan dalam menilai sikap, pandangan, atau persepsi individu ataupun kelompok pada suatu peristiwa, yang didasarkan pada definisi operasional yang sudah ditentukan oleh peneliti. Secara umum, terdapat lima pilihan beserta skor yang bisa dilihat pada tabel berikut.

Tabel 8. Tabel Kategori dan Skor

No	Kategori	Skor
1	Sangat Setuju (SS)	5
2	Setuju (S)	4
3	Cukup Setuju (CS)	3
4	Tidak Setuju (TS)	2
5	Sangat Tidak Setuju (STS)	1

Tabel 9. Tabel Kategori dan Persentase

No	Kategori	Persentase
1	Sangat Setuju (SS)	81% - 100%
2	Setuju (S)	61% - 80%
3	Cukup Setuju (CS)	41% - 60%
4	Tidak Setuju (TS)	21% - 40%
5	Sangat Tidak Setuju (STS)	<20%

Jumlah skor dari hasil pengisian kuesioner dianalisis menggunakan rumus di bawah ini:

$$\text{Skor} = T \times P_n$$

$$\text{Indeks Kelayakan} = \frac{\text{total skor}}{y} \times 100$$

Keterangan : T = Jumlah responden yang memilih opsi

P_n = Pilihan skor *likert*

Y = Skor tertinggi x jumlah responden

Adapun hasil pengujian UAT dari sistem ini dapat dilihat pada tabel sebagai berikut:

Tabel 10. Hasil Pengujian

No	Pernyataan	Jawaban					Presentase	Kategori
		5	4	3	2	1		
1	Tampilan <i>user interface</i> sistem ini menarik bagi pengguna	10	9				90 %	Sangat Layak
2	Tampilan sistem mudah dipahami dan digunakan	14	5				95 %	Sangat Layak
3	Sistem dapat menghasilkan jadwal tanpa bentrok	17	2				98 %	Sangat Layak
4	Proses penyusunan jadwal lebih cepat dibandingkan sistem sebelumnya	17	2				98 %	Sangat Layak
5	Informasi jadwal bisa diakses dengan mudah oleh dosen dan mahasiswa	15	4				96 %	Sangat Layak
6	Fitur dalam sistem bekerja sesuai harapan	9	10				89 %	Sangat Layak

$$\begin{aligned} \text{Persentase rata-rata} &= \frac{\text{Jumlah persentase}}{\text{jumlah soal kuesioner}} \\ &= \frac{90\%+95\%+98\%+98\%+96\%+89\%}{6} \\ &= \frac{566}{6} = 94\% \end{aligned}$$

Berdasarkan hasil pengujian sistem menggunakan skala *likert* dengan 6 pertanyaan dan 19 responden, maka ditotalkan skornya adalah 94%.

4. KESIMPULAN

Berdasarkan hasil penelitian dan pengembangan sistem yang telah dilakukan, dapat disimpulkan bahwa perancangan sistem cerdas penjadwalan mata kuliah otomatis berbasis web dengan penerapan algoritma genetika telah berhasil diwujudkan dan mampu menjawab permasalahan yang ada dalam proses penyusunan jadwal kuliah di Fakultas Sains dan Teknologi UIN Sulthan Thaha Saifuddin Jambi. Sistem ini hadir sebagai solusi atas berbagai kendala yang selama ini muncul dalam proses penjadwalan manual, seperti bentrokan jadwal antar mata kuliah, keterbatasan ruang kelas, dan kesulitan dalam menyesuaikan waktu ketersediaan dosen. Dengan pendekatan pengembangan prototyping, sistem dapat dikembangkan secara iteratif berdasarkan kebutuhan pengguna, yang melibatkan staf akademik, program studi, serta masukan dari mahasiswa. Implementasi algoritma genetika memungkinkan sistem untuk menghasilkan jadwal kuliah yang optimal secara otomatis, mengurangi beban kerja staf, serta meningkatkan efisiensi dan efektivitas

penyusunan jadwal. Dari hasil pengujian yang dilakukan melalui metode blackbox testing dan user acceptance testing (UAT), sistem menunjukkan kinerja yang sangat baik dengan tingkat kepuasan pengguna mencapai 94%. Hal ini menunjukkan bahwa sistem yang dibangun tidak hanya layak digunakan, tetapi juga memberikan kemudahan bagi pengguna dalam mengakses informasi penjadwalan kapan saja dan di mana saja melalui platform berbasis web. Dengan demikian, sistem ini dapat menjadi alat bantu strategis bagi fakultas dalam meningkatkan kualitas layanan akademik, khususnya dalam pengelolaan jadwal perkuliahan secara cerdas, cepat, dan minim kesalahan.

REFERENCES

- [1] A. M. Korompis, A. C. Djamen, And J. R. Batmetan, “Pengembangan Sistem Informasi Pengelolaan Ruang Kuliah Berbasis Web Di Jurusan Pendidikan Teknologi Informasi Dan Komunikasi Fakultas Teknik Universitas Negeri Manado,” *Journal Of Education Method And Technology*, Vol. 2(2), 2024.
- [2] H. Marcellino And I. Gede Pasek Suta Wijaya, “Sistem Informasi Akademik Penjadwalan Mata Kuliah Berbasis Website Fkip Unram (Web-Based Course Scheduling Information System Of Fkip Unram),” 2022. [Online]. Available: <http://begawe.unram.ac.id/index.php/jbti/>
- [3] W. Darlin, A. Dwi Putra, N. Hendrastuty, N. Penulis, K. : Wayan, And D. Submitted, “Sistem Informasi Manajemen Kost Putra Trisula Berbasis Web (Studi Kasus : Asrama Putra Trisula),” Vol. 4, No. 3, Pp. 240–249, 2023, doi: 10.33365/jtsi.
- [4] T. G. Bhagya, “Algoritma Genetik Pada Penjadwalan Transportasi Kapal Laut (Studi Kasus Pt. Peln),” *Journal Of Industrial And Manufacture Engineering*, Vol. 5, No. 2, Nov. 2021, Doi: 10.31289/Jime.V5i2.5952.
- [5] N. Hidayat, P. Wijayakusuma, Y. Sainika, And I. Susanto, “Perancangan Website E-Commerce Produk Kopi Menggunakan Metode Prototyping (Studi Kasus: Kedai Kopi Kontekstual),” *Journal Of Information Systems And Informatics*, Vol. 3, No. 3, 2021, [Online]. available: <http://journal-isi.org/index.php/ispublishedbydrpm-ubd>
- [6] F. D. Ramadhani, K. K. A. Rahman, M. Y. Rafi, U. Salamah, And P. Rosyani, “Perancangan Sistem Informasi Penjadwalan Mata Kuliah Menggunakan Algoritma Genetika Berbasis Web,” *Jurnal Kreativitas Mahasiswa Informatika*, Vol. 1(3), 2020.
- [7] R. Kurniawati, “Penjadwalan Perkuliahan Dengan Metode Algoritma Genetika,” 2021.
- [8] T. Kristian Jeriko, D. Faizal Racma, C. Ety Widjayanti, And A. Ary Setyawan, “Penerapan Algoritma Genetika Dalam Sistem Informasi Penjadwalan Mata Kuliah Berbasis Website Pada Stikom Yos Sudarso Purwokerto,” *Jurnal Muara Sains, Teknologi, Kedokteran, Dan Ilmu Kesehatan*, Vol. 6, No. 1, Pp. 101–118, 2022, Doi: 10.24912/Jmstkik.V6i1.17262.
- [9] Andika, M. Salsabil, H. T., And N. Salman, “Implementasi Algoritma Genetika Untuk Penjadwalan Mata Kuliah Berbasis Web,” *Jurnal Dipanegara Komputer Teknik Informatika*, Vol. 15(2), Dec. 2022.
- [10] M. A. Rohim, F. Wiranto, And D. A. Fauziah, “Optimasi Pembuatan Jadwal Perkuliahan Menggunakan Algoritma Genetika Berbasis Pendekatan Multivariat,” *Bios : Jurnal Teknologi Informasi Dan Rekayasa Komputer*, Vol. 6, No. 1, Pp. 31–38, Mar. 2025, Doi: 10.37148/Bios.V6i1.160.
- [11] R. Hartono And A. Zein, “Penerapan Algoritma Genetika Dan Jaringan Syaraf Tiruan Dalam Penjadwalan Mata Kuliah Studi Kasus : Prodi Sistem Informasi Universitas Pamulang,” 6 | *Jurnal Ilmu Komputer Jik*, Vol. Vi, No. 03, 2023.
- [12] E. Yulia Sari, D. Yulina Heriyani, And Dan Titik Rahmawati, “Pemodelan Sistem Optimasi Penjadwalan Matakuliah Dengan Algoritma Genetika (Course Scheduling Optimization System Modeling With Genetic Algorithms),” Jun. 2023.
- [13] Zulmahendra And S. Derta, “Penjadwalan Kuliah Otomatis Menggunakan Algoritma Genetika,” *Jsi : Jurnal Sistem Informasi (E-Journal)*, Vol. 12, No. 2, 2020, [Online]. Available: <http://ejournal.unsri.ac.id/index.php/jsi/index>

- [14] E. L. Hady, K. Haryono, And N. W. Rahayu, “User Acceptance Testing (Uat) Pada Purwarupa Sistem Tabungan Santri (Studi Kasus: Pondok Pesantren Al-Mawaddah) User Acceptance Testing (Uat) Of The Prototype Of Students’ Savings Information System (Case Study: Al-Mawaddah Islamic Boarding School),” 2020.
- [15] A. Ulfah, D. Hermina, And N. Huda, “Desain Instrumen Evaluasi Yang Valid Dan Reliabel Dalam Pendidikan Islam Menggunakan Skala Likert,” 2024.