

# Design and Development of Lecture Scheduling and Learning Equipment Loaning System Using Genetic Algorithm

Putri Nurul Izza<sup>1</sup>, Muhammad Arhami<sup>2\*</sup>, Mahdi<sup>3</sup>

<sup>1,2,3</sup> Jurusan Teknologi Informasi dan Komputer Politeknik Negeri Lhokseumawe Jln. B.Aceh Medan Km.280 Buketrata 24301  
INDONESIA

\*Penulis Korespondensi : muhammad.arhami@pnl.ac.id

## INFORMASI ARTIKEL

### Riwayat artikel:

Diajukan pada 10 Mei 25  
Direvisi pada 01 Juni 25  
Publikasi pada 20 Juni 25

### Kata kunci:

Penjadwalan Kuliah  
Sistem Peminjaman  
Proses Belajar Mengajar  
*Algoritma Genetika*

### Keywords:

Course Scheduling  
Loaning System  
Teaching Learning Process  
Genetic Algorithm

## ABSTRAK

Jadwal perkuliahan dan peminjaman alat pendukung Proses Belajar Mengajar (PBM) merupakan dua hal yang mendukung kelancaran proses belajar mengajar di perguruan tinggi. Proses penyusunan jadwal kuliah seringkali mengalami kesulitan karena terdapat beberapa aturan yang harus ditetapkan sehingga membutuhkan waktu yang lebih lama. Selain itu, prosedur peminjaman alat pendukung PBM juga membutuhkan kemudahan dan kecepatan dalam prosesnya. Penelitian ini bertujuan untuk merancang dan membangun sistem penjadwalan kuliah yang disinkronkan dengan sistem peminjaman alat PBM menggunakan *Algoritma Genetika*. *Algoritma Genetika* digunakan untuk memproses data terkait perkuliahan yang akan menghasilkan penjadwalan kuliah sesuai aturan-aturan yang ditetapkan. Pengujian kinerja *Algoritma Genetika* akan dilakukan dengan melakukan *generate* penjadwalan sebanyak 4 kali dengan memodifikasi nilai-nilai dari operator genetika. Hasil pengujian menunjukkan bahwa *algoritma genetika* menghasilkan solusi yang paling optimal ketika nilai *crossover rate* ditingkatkan dan nilai *mutation rate* diturunkan dengan persentase kesalahan hanya 0.03% dan waktu yang dibutuhkan yaitu 273.6 detik. Selain itu, pengujian fungsionalitas sistem dengan metode *black box* menunjukkan tingkat keberhasilan sebesar 95.65%.

## ABSTRACT

*Class scheduling and borrowing of supporting tools for the teaching and learning process are two elements that enhance the smooth running of the teaching and learning process in universities. The process of preparing class schedules often encounters difficulties due to several rules that must be set, requiring more time. Additionally, the procedure for borrowing supporting tools for the teaching and learning process also requires ease and speed. This study aims to design and build a class scheduling system synchronized with the borrowing system for teaching and learning process tools using a genetic algorithm. The Genetic Algorithm is used to process data related to lectures to produce class schedules that comply with the established rules. The performance of the genetic algorithm will be tested by generating schedules four times while modifying the values of the genetic operators. The test results show that the genetic algorithm produces the most optimal solution when the crossover rate is increased, and the mutation rate is decreased, with an error percentage of only 0.03% and a processing time of 273.6 seconds. Additionally, the system's functionality testing using the black-box method shows a success rate of 95.65%.*

## 1. Pendahuluan

Penjadwalan merupakan proses pengaturan susunan urutan kegiatan berdasarkan waktu yang tersedia sehingga pelaksanaan kegiatan dapat berlangsung secara teratur tanpa adanya benturan antara satu kegiatan dengan kegiatan lainnya [1]. Penjadwalan mata kuliah merupakan salah satu faktor pendukung dalam pelaksanaan proses belajar mengajar di perguruan tinggi. Proses penjadwalan mata kuliah memiliki banyak aturan-aturan (*constraints*) tertentu yang harus dipertimbangkan sehingga tidak terjadi bentrokan antar jadwal [2]. Aturan-aturan tersebut membuat proses pembuatan jadwal menjadi lama dan harus diatur sebaik mungkin. Penelitian-penelitian sebelumnya telah menerapkan algoritma optimasi penjadwalan dalam proses pembuatan jadwal kuliah atau mata pelajaran, salah satunya dengan *Algoritma Genetika*.

*Algoritma genetika* adalah prosedur pencarian global yang mencari dari satu populasi solusi ke populasi solusi lainnya yang berfokus pada area solusi terbaik [3]. *Algoritma Genetika* telah berhasil digunakan oleh banyak peneliti untuk memberikan solusi yang baik dalam berbagai masalah yang kompleks di berbagai bidang. Penelitian yang telah dilakukan oleh Afrizal Nehemia Toscani dan Rusdianto Roestam menghasilkan bahwa implementasi *Algoritma Genetika* sangat mendukung proses penjadwalan dengan menghasilkan jadwal yang lebih presisi sesuai dengan penempatan jam mengajar, ruang kelas dan dosen yang tepat dan juga mempercepat proses penyusunan dan pembuatan jadwal perkuliahan [4].

Politeknik Negeri Lhokseumawe merupakan salah satu perguruan tinggi vokasi di Provinsi Aceh yang memiliki 6 jurusan, salah satunya yaitu Jurusan Teknologi Informasi dan Komputer (TIK). Pelaksanaan kegiatan belajar mengajar di jurusan TIK dilaksanakan sesuai dengan jadwal perkuliahan yang telah disusun. Namun, keberhasilan kegiatan perkuliahan tidak hanya ditentukan oleh jadwal yang baik, melainkan juga oleh ketersediaan alat pendukung Proses Belajar Mengajar (PBM) yang menjadi unsur penting dalam proses pembelajaran.

Alat pendukung PBM seperti kunci ruang kuliah, kunci laboratorium, dan proyektor menjadi salah satu penunjang keberhasilan kegiatan perkuliahan. Peminjaman alat ini dilakukan di ruang layanan yang ditangani oleh seorang petugas. Prosedur peminjaman dilakukan dengan pendataan nama mahasiswa yang akan meminjam, kelas, dosen yang mengajar, nomor ruang kuliah atau laboratorium, dan kode proyektor yang akan dipinjam. Data tersebut dicatat dalam sebuah buku peminjaman. Prosedur peminjaman alat pendukung PBM yang saat ini diterapkan di Jurusan TIK menimbulkan beberapa permasalahan. Pertama, sering terjadi antrean, terutama di pagi hari, yang dapat menyebabkan keterlambatan dalam pelaksanaan perkuliahan sesuai jadwal. Kedua, terdapat ketidaksinkronan antara jadwal perkuliahan dengan proses peminjaman alat yang dilakukan oleh mahasiswa, sehingga berpotensi mengganggu kelancaran kegiatan belajar mengajar. Ketiga, kurangnya informasi yang tersedia bagi petugas peminjaman mengenai alat PBM yang belum dikembalikan, serta kesulitan dalam melacak keberadaan alat karena harus memeriksa kembali catatan peminjaman secara manual, yang memerlukan waktu lebih lama dan tidak efisien.

Permasalahan-permasalahan tersebut dapat diselesaikan dengan menerapkan teknologi dalam prosesnya. Sistem peminjaman alat PBM yang disinkronkan dengan penjadwalan mata kuliah yang akan dibuat secara otomatis dengan menerapkan *algoritma genetika* dalam prosesnya. Sistem penjadwalan kuliah

dengan *Algoritma Genetika* ini akan dikembangkan dengan penambahan aturan-aturan (*constraints*) yang disesuaikan dengan kondisi pada Jurusan TIK. Penelitian ini bertujuan untuk merancang dan membangun sistem penjadwalan kuliah dengan menerapkan *Algoritma Genetika* yang kemudian disinkronkan dengan sistem peminjaman alat pendukung proses belajar mengajar. Selain itu, penelitian ini menguji kinerja *Algoritma Genetika* dalam menghasilkan penjadwalan kuliah.

## 2. Metode

### 2.1 *Algoritma Genetika*

John Holland pertama kali memperkenalkan *Algoritma Genetika* menyatakan bahwa *Algoritma Genetika* dapat digunakan untuk menjelaskan setiap masalah adaptasi, baik buatan maupun alami. *Algoritma Genetika* adalah teknik pencarian heuristik yang didasarkan pada gagasan evolusi seleksi alam dan genetik. Algoritma ini memanfaatkan proses seleksi alamiah yang dikenal dengan proses evolusi. Pada proses evolusi, individu secara terus-menerus mengalami perubahan gen untuk menyesuaikan dengan lingkungan hidupnya, hanya individu yang kuat yang mampu bertahan [5].

Struktur suatu *Algoritma Genetika* dapat didefinisikan dengan langkah-langkah sebagai berikut.

1. Membangkitkan populasi awal

Populasi awal ini dibangkitkan secara *random* sehingga didapatkan solusi awal. Populasi itu sendiri terdiri atas sejumlah kromosom yang merepresentasikan solusi yang diinginkan.

2. Membentuk generasi baru

Pembentukan generasi baru menggunakan operator reproduksi/seleksi, *crossover* dan mutasi. Proses ini dilakukan berulang-ulang sehingga didapatkan jumlah kromosom yang cukup untuk membentuk generasi baru yang merupakan representasi dari solusi baru. Generasi baru ini dikenal dengan istilah anak (*offspring*).

3. Evaluasi Solusi

Pada tiap generasi, kromosom akan melalui proses evaluasi dengan menggunakan alat ukur yang dinamakan *fitness*. Nilai *fitness* suatu kromosom menggambarkan kualitas kromosom dalam populasi tersebut. Proses ini akan mengevaluasi setiap populasi dengan menghitung nilai *fitness* setiap kromosom dan mengevaluasinya sampai terpenuhi kriteria berhenti. Jika kriteria berhenti belum terpenuhi maka akan dibentuk lagi generasi baru dengan mengulangi langkah pembentukan generasi baru. Beberapa kriteria berhenti yang sering digunakan antara lain: berhenti pada generasi tertentu, berhenti setelah dalam beberapa generasi berturut-turut didapatkan nilai *fitness* tertinggi tidak berubah, berhenti bila dalam n generasi berikut tidak didapatkan nilai *fitness* yang lebih tinggi [6].

### 2.2 Operator Genetika

#### 2.2.1 Seleksi

Seleksi bertujuan untuk memberikan kesempatan reproduksi yang lebih besar bagi anggota populasi yang paling fit. Langkah pertama dalam seleksi ini adalah pencarian nilai *fitness*. Masing-masing individu dalam suatu wadah seleksi akan menerima probabilitas reproduksi yang tergantung pada nilai objektif dirinya sendiri terhadap nilai objektif dari semua individu dalam wadah seleksi tersebut. Nilai *fitness* inilah yang nantinya akan digunakan pada tahap-tahap seleksi berikutnya [7].

Seleksi yang akan digunakan adalah seleksi roda rolet. Seleksi ini menggunakan *fitness* skala dalam memilih kromosom dari populasi. Prosedur seleksinya sebagai berikut:

- a. Hitung total *fitness*

$$Total\ Fitness = \sum F_k, k = 1, 2, 3, \dots, popsize \quad (1)$$

- b. Hitung *fitness* relatif tiap individu

$$P_k = \frac{F_k}{Total\ Fitness} \quad (2)$$

- c. Hitung *fitness* kumulatif

$$q_i = p_i \quad (3)$$

- d. Pilih induk yang akan menjadi kandidat untuk di-*crossover* dengan cara:

- i. Bangkitkan bilangan random  $r$
- ii. Jika  $q_i \leq r$  maka pilihlah kromosom ke  $(k + 1)$  sebagai kandidat induk.

### 2.2.2 Crossover

Operator *crossover* dalam *Algoritma Genetika* digunakan untuk menambah keanekaragaman *string* dalam populasi melalui penyilangan antar *string* yang diperoleh dari proses seleksi sebelumnya. Langkah-langkah *crossover* tersebut sebagai berikut:

- a. Bangkitkan bilangan acak  $r$  antara  $[0,1]$
- b. Pilih bilangan acak tersebut yang kurang dari  $pc$ . Jika jumlah bilangan acak yang dihasilkan dari pilihan tersebut ganjil maka buang salah satunya.
- c. Pilih bilangan acak antara 1 sampai  $(L-1)$ ,  $L$  adalah panjang kromosom. Bilangan ini akan menentukan posisi *crossover* 1 titik.
- d. Penyilangan ini dilakukan untuk semua pasangan kromosom yang akan disilangkan.

### 3.2.3 Mutasi

Mutasi dilakukan untuk menghindari terjadinya konvergensi prematur. Proses mutasi dilakukan dengan mengubah gen-gen tertentu secara acak sehingga memunculkan beragam calon solusi. Pemilihan gen dilakukan secara acak untuk dimutasi. Mutasi dapat ditentukan dengan langkah-langkah berikut:

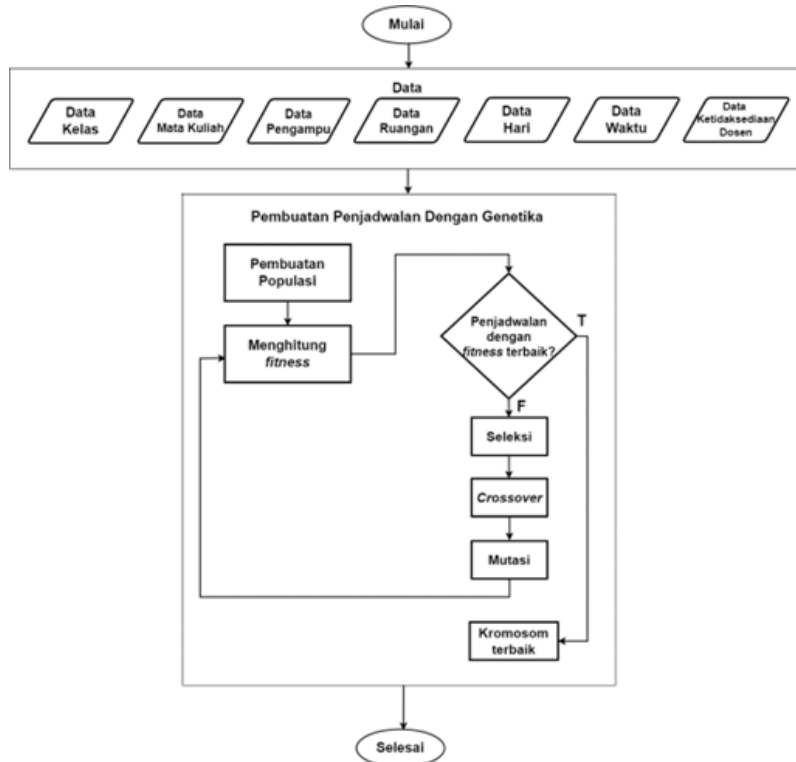
- a. Hitung jumlah bit yang ada pada populasi, yaitu  $JumlahBit = Ukuran\ populasi \times L$
- b. Bangkitkan bilangan acak antara  $[0\ 1]$  sebanyak  $JumlahBit$
- c. Pilih bilangan acak tersebut yang kurang dari  $pm$ . Kemudian, lakukan mutasi [6].

## 2.3 Arsitektur dan Variabel Penelitian

Gambar 1 menggambarkan alur pembuatan penjadwalan kuliah dengan *Algoritma Genetika* serta variabel-variabel yang digunakan. Variabel-variabel yang digunakan dalam penelitian ini yaitu kelas, mata kuliah, pengampu, ruangan, hari, waktu, dan hari ketidaksediaan dosen mengajar. Selain itu, beberapa *constraint* yang ditetapkan yaitu:

- a. Tidak boleh terdapat dua mata kuliah atau lebih pada satu ruangan dan waktu yang sama
- b. Tidak boleh terdapat jadwal pada hari yang sudah dipilih oleh dosen sebagai hari ketidaksediaannya mengajar
- c. Mata kuliah dengan jenis praktikum harus di laboratorium
- d. Tidak boleh ada jadwal untuk hari Jum'at pada jam 12:00-12:50

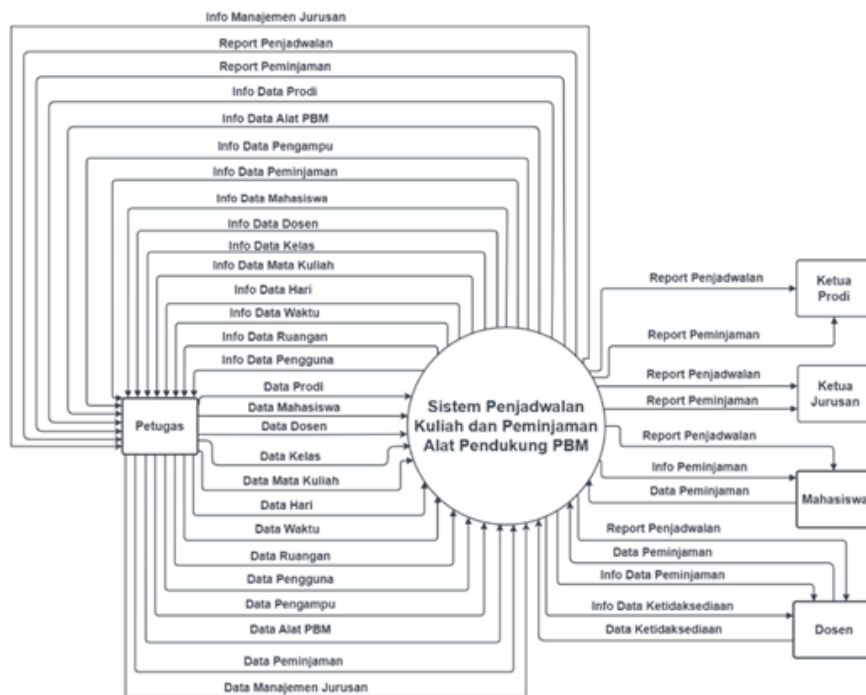
- e. Jika suatu mata kuliah memiliki ruangan yang telah ditetapkan, maka mata kuliah tersebut harus di ruangan yang telah ditetapkan.



Gambar 1. Arsitektur Penjadwalan dengan *Algoritma Genetika*

### 2.4 Diagram Konteks

Diagram konteks sistem ini dirancang dengan 5 entitas yaitu petugas, dosen, mahasiswa, ketua prodi, dan ketua jurusan yang digambarkan seperti pada Gambar 2 berikut:



Gambar 2. Diagram Konteks

Diagram konteks pada gambar 2 menunjukkan bahwa pada sistem ini terdapat 5 entitas, yaitu ketua jurusan, ketua prodi, petugas, dosen, dan mahasiswa. Entitas ketua jurusan dan ketua prodi hanya dapat mengakses *report* penjadwalan kuliah dan *report* peminjaman alat pendukung PBM. Entitas mahasiswa dapat mengakses *report* penjadwalan dan membuat peminjaman alat PBM. Entitas dosen dapat mengakses *report* penjadwalan, mengelola hari ketidaksediaan dosen mengajar, dan membuat peminjaman alat PBM. Entitas petugas memiliki akses untuk mengelola semua data-data yang ada, yaitu data mahasiswa, data dosen, data pengampu, data kelas, data mata kuliah, data hari, data waktu, data ruangan, data pengguna, data peminjaman, data manajemen jurusan, data prodi, dan data alat PBM. Selain itu, petugas dapat mengakses *report* penjadwalan dan *report* peminjaman.

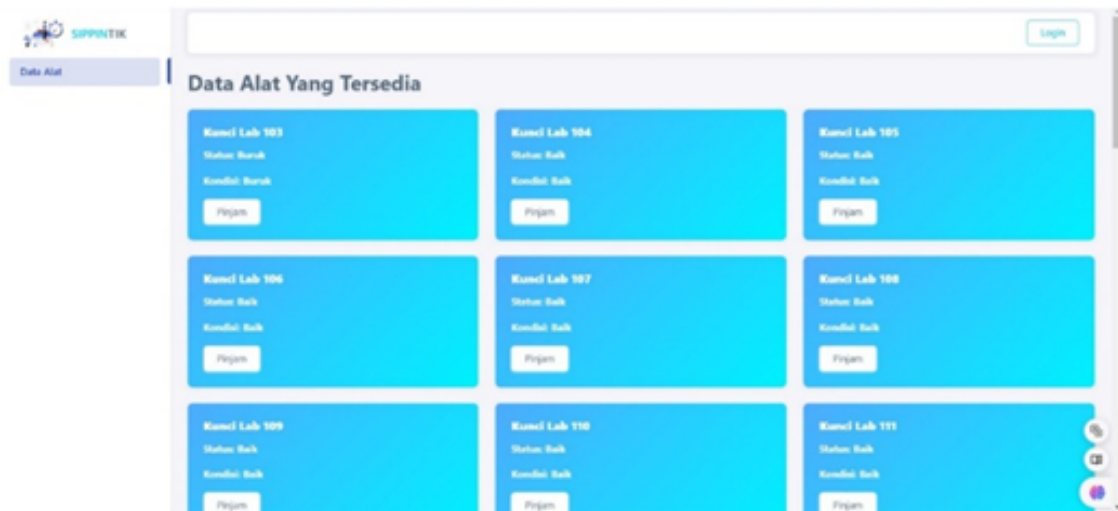
### 3. Hasil Dan Pembahasan

#### 3.1 Antarmuka Pengguna

Hasil implementasi antarmuka pengguna dari sistem penjadwalan kuliah dan peminjaman alat pendukung proses belajar mengajar ini sebagai berikut:

##### 3.1.1 Halaman *Landing*

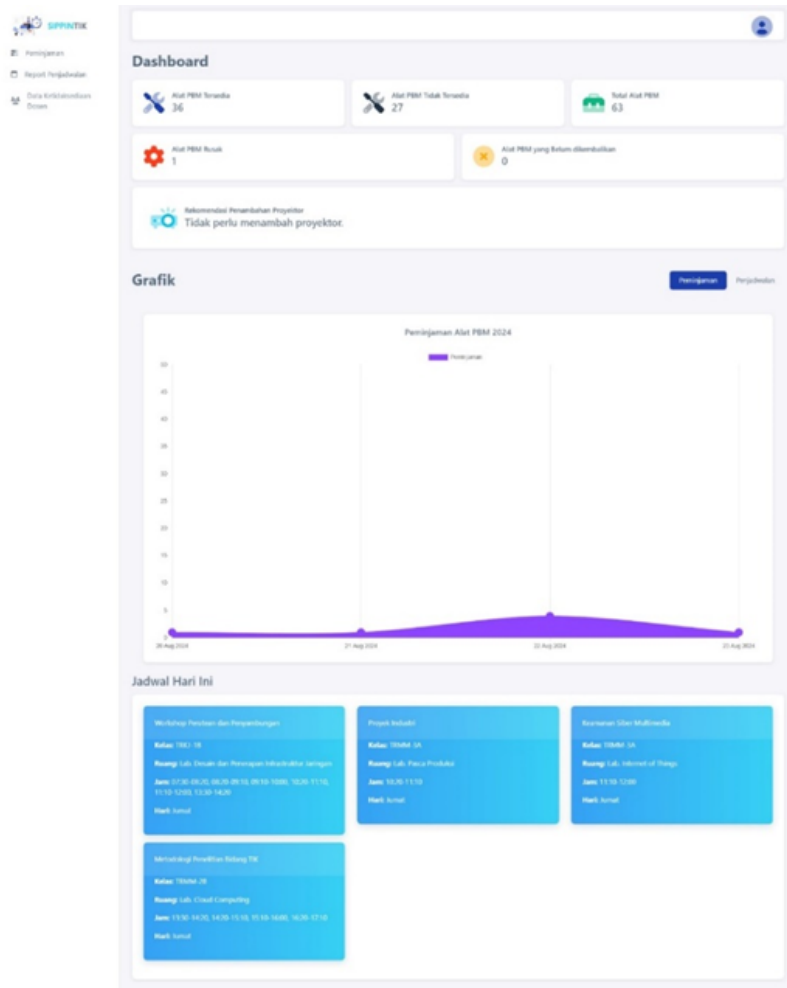
Hasil implementasi halaman *landing* pada sistem ini seperti yang terdapat pada Gambar 3 dimana menggambarkan tampilan halaman *landing* yang informasi mengenai alat pendukung PBM yang tersedia untuk dipinjam.



**Gambar 3.** Halaman *Landing*

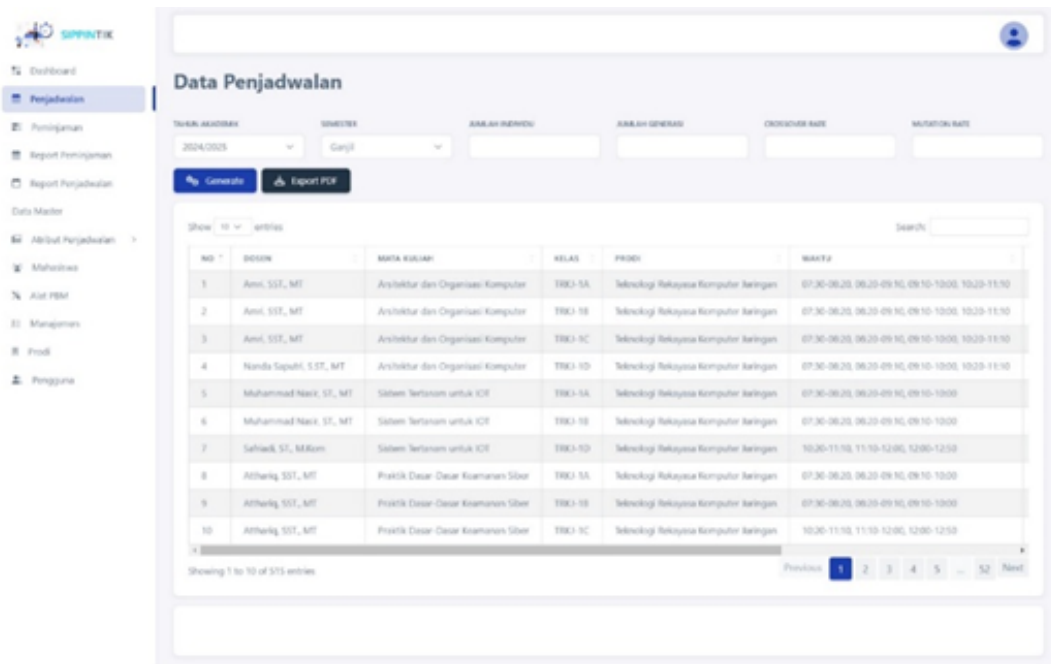
##### 3.1.2 Halaman *Dashboard*

Gambar 4 menggambarkan tampilan halaman *dashboard* yang memuat informasi terkait alat PBM yang tersedia, alat PBM yang tidak tersedia atau sedang dipinjam, total alat PBM yang ada, alat PBM yang kondisinya rusak, alat PBM yang belum dikembalikan, serta rekomendasi penambahan proyektor. Selain itu, terdapat grafik peminjaman dan grafik yang menunjukkan jumlah penjadwalan tiap kelas. Selain itu, terdapat informasi jadwal perkuliahan hari ini untuk dosen dan mahasiswa.



Gambar 4. Halaman Dashboard

### 3.1.3 Halaman Penjadwalan

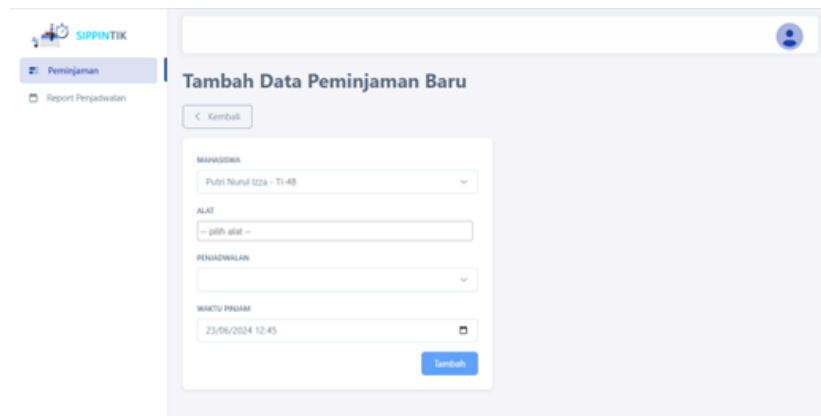


Gambar 5. Halaman Penjadwalan

Gambar 5 menggambarkan tampilan halaman penjadwalan. Sebelum memulai proses generate penjadwalan, petugas memasukkan terlebih dahulu tahun akademik, semester, jumlah individu, jumlah generasi, *crossover rate*, dan *mutation rate* kemudian menekan tombol *generate*. Hasil dari pembuatan jadwal ini akan ditampilkan pada tabel dan dapat diekspor ke dalam bentuk pdf dan excel.

#### 3.1.4 Halaman Peminjaman

Gambar 6 menggambarkan tampilan halaman tambah peminjaman berdasarkan jadwal yang dapat dilakukan oleh mahasiswa. Pada halaman ini, mahasiswa hanya dapat membuat peminjaman apabila memiliki jadwal perkuliahan untuk kelasnya pada jam yang sesuai. Proyektor dapat dipilih oleh mahasiswa sedangkan kunci ruangan akan ditetapkan langsung oleh sistem sesuai jadwal perkuliahannya.



**Gambar 6.** Halaman Peminjaman

### 3.2 Analisis Hasil Penjadwalan dengan *Algoritma Genetika*

Pembuatan jadwal dengan *Algoritma Genetika* ini dimulai dari proses pembuatan populasi awal, evaluasi *fitness*, seleksi, *crossover*, dan mutasi.

#### 3.2.1 Pembuatan Populasi Awal

Proses pembuatan populasi awal ini akan menghasilkan individu-individu yang berisi kromosom yang merepresentasikan penjadwalan kuliah. Jadwal yang dihasilkan dari proses ini memungkinkan masih adanya jadwal yang tumpang tindih. Berikut adalah *array* yang berisi populasi awal dengan 4 individu yang telah dibentuk.

a. Individu 1:

[{"Amri, SST., MT", "Arsitektur dan Organisasi Komputer", "TRKJ-1A", "07:30-08:20, 08:20-09:10, 09:10-10:00, 10:20-11:10", "Jumat", "Ruang Kuliah 302"}, {"Amri, SST., MT", "Arsitektur dan Organisasi Komputer", "TRKJ-1B", "07:30-08:20, 08:20-09:10, 09:10-10:00, 10:20-11:10", "Jumat", "Ruang Kuliah 311"}, {"Amri, SST., MT", "Arsitektur dan Organisasi Komputer", "TRKJ-1C", "07:30-08:20, 08:20-09:10, 09:10-10:00, 10:20-11:10", "Senin", "Ruang Kuliah 310"}, ...]

b. Individu 2:

[{"Amri, SST., MT", "Arsitektur dan Organisasi Komputer", "TRKJ-1A", "07:30-08:20, 08:20-09:10, 09:10-10:00, 10:20-11:10", "Senin", "Ruang Kuliah 302"}, {"Amri, SST., MT", "Arsitektur dan Organisasi Komputer", "TRKJ-1B", "07:30-08:20, 08:20-09:10, 09:10-10:00, 10:20-11:10", "Kamis", "Ruang Kuliah 311"}, {"Amri, SST., MT", "Arsitektur dan Organisasi

Komputer”, “TRKJ-1C”, “07:30-08:20, 08:20-09:10, 09:10-10:00, 10:20-11:10”, “Kamis”, “Ruang Kuliah 310”}, ...]

c. Individu 3:

[{"Amri, SST., MT", "Arsitektur dan Organisasi Komputer", "TRKJ-1A", "08:20-09:10, 09:10-10:00, 10:20-11:10, 11:10-12:00", "Senin", "Ruang Kuliah 302"}, {"Amri, SST., MT", "Arsitektur dan Organisasi Komputer", "TRKJ-1B", "07:30-08:20, 08:20-09:10, 09:10-10:00, 10:20-11:10", "Kamis", "Ruang Kuliah 311"}, {"Amri, SST., MT", "Arsitektur dan Organisasi Komputer", "TRKJ-1C", "07:30-08:20, 08:20-09:10, 09:10-10:00, 10:20-11:10", "Kamis", "Ruang Kuliah 310"}], ...]

d. Individu 4:

[{"Amri, SST., MT", "Arsitektur dan Organisasi Komputer", "TRKJ-1A", "08:20-09:10, 09:10-10:00, 10:20-11:10, 11:10-12:00", "Senin", "Ruang Kuliah 301"}, {"Amri, SST., MT", "Arsitektur dan Organisasi Komputer", "TRKJ-1B", "07:30-08:20, 08:20-09:10, 09:10-10:00, 10:20-11:10", "Jumat", "Ruang Kuliah 310"}, {"Amri, SST., MT", "Arsitektur dan Organisasi Komputer", "TRKJ-1C", "07:30-08:20, 08:20-09:10, 09:10-10:00, 10:20-11:10", "Kamis", "Ruang Kuliah 310"}], ...]

### 3.2.2 Evaluasi *Fitness*

Proses perhitungan nilai *fitness* ini bertujuan untuk mengukur kualitas tiap individu dalam populasi berdasarkan jumlah konflik yang terjadi. Terdapat dua jenis konflik yang diperhitungkan dalam sistem ini karena aturan berikut dijadikan *hard constraints*, yaitu:

- Jika terdapat dua mata kuliah atau lebih pada satu ruangan dan waktu yang sama
- Jika terdapat hari dosen mengajar yang seharusnya dosen sudah memilih hari tersebut sebagai hari ketidaksediaannya untuk mengajar.

Gambar 7 menunjukkan hasil perhitungan *conflict* dan *fitness* pada tiap individu. Dari keempat individu yang dibuat masih terdapat banyak aturan yang belum terpenuhi. *Conflict* terendah dimiliki oleh individu kedua dan memiliki nilai *fitness* yang paling tinggi.

```
Fitness-1: 0.45711, Conflicts: 98
Fitness-2: 0.66663, Conflicts: 82
Fitness-3: 0.4, Conflicts: 96
Fitness-4: 0.29481, Conflicts: 128
```

**Gambar 7.** *Conflict* dan *Fitness* Tiap Individu

### 3.2.3 Seleksi

Proses seleksi dilakukan dengan teknik *Roulette Wheel* untuk pemilihan kromosom yang mempunyai nilai *fitness* lebih tinggi memiliki kemungkinan terpilih lebih besar. Proses ini dilakukan dengan menghitung probabilitas kumulatif tiap individu. Kemudian dilakukan *generate* nilai *random* dengan batasan 0 sampai 1 untuk masing-masing individu. Nilai *random* dari tiap-tiap individu ini akan untuk menentukan area milik individu tersebut. Misalnya, jika nilai *random* dari individu pertama terletak pada area milik individu kedua maka kromosom individu pertama akan berisi kromosom dari individu kedua. Proses ini akan menghasilkan populasi baru yang cenderung memiliki lebih sedikit konflik.

### 3.2.4 *Crossover*

Proses *crossover* dilakukan dengan menyilangkan kromosom satu individu dengan individu lain yang disilangkan. Penyilangan ini dilakukan berdasarkan nilai *random*. Proses ini akan menghasilkan populasi baru dari tiap individu yang disilangkan. Berikut ini salah satu hasil dari proses *crossover* yang dilakukan.

#### Individu 1 x Individu 4:

- a. Individu 1 (sebelum *crossover*):  
[{"Amri, SST., MT", "Arsitektur dan Organisasi Komputer", "TRKJ-1A", "07:30-08:20, 08:20-09:10, 09:10-10:00, 10:20-11:10", "Jumat", "Ruang Kuliah 302"}, ...]
- b. Individu 4 (sebelum *crossover*):  
[{"Amri, SST., MT", "Arsitektur dan Organisasi Komputer", "TRKJ-1C", "07:30-08:20, 08:20-09:10, 09:10-10:00, 10:20-11:10", "Kamis", "Ruang Kuliah 310"}, ...]
- c. Hasil *crossover*:  
Anak 1: [{"Amri, SST., MT", "Arsitektur dan Organisasi Komputer", "TRKJ-1A", "07:30-08:20, 08:20-09:10, 09:10-10:00", "Jumat", "Ruang Kuliah 302"}, {"Amri, SST., MT", "Arsitektur dan Organisasi Komputer", "TRKJ-1C", "10:20-11:10", "Kamis", "Ruang Kuliah 310"}, ...]  
Anak 2: [{"Amri, SST., MT", "Arsitektur dan Organisasi Komputer", "TRKJ-1C", "07:30-08:20, 08:20-09:10, 09:10-10:00", "Kamis", "Ruang Kuliah 310"}, {"Amri, SST., MT", "Arsitektur dan Organisasi Komputer", "TRKJ-1A", "10:20-11:10", "Jumat", "Ruang Kuliah 302"}, ...]

#### 3.2.4 Mutasi

Proses mutasi ini dilakukan jika nilai *random* yang dihasilkan lebih kecil dari nilai *mutation rate* (MR) sehingga tidak semua kromosom akan mengalami mutasi untuk meminimalkan risiko kerusakan solusi yang sudah baik. Mutasi dilakukan dengan mengganti nilai-nilai hari, waktu, dan ruangan secara acak, tetapi menjaga atribut *teach*, *subject*, dan kelas tetap sama.

Penjadwalan yang dihasilkan dalam satu kali proses *algoritma genetika* dengan nilai parameter jumlah individu = 4, jumlah generasi = 50, *crossover rate* = 0.9, dan *mutation rate* = 0.3 menghasilkan bahwa masih memiliki 3 jadwal yang tumpang tindih yang dikarenakan kesalahan dalam pemilihan waktu. Oleh karena itu, dilakukan pengujian kinerja *Algoritma Genetika* dalam menghasilkan penjadwalan kuliah dengan memasukkan kombinasi nilai parameter jumlah generasi, *crossover rate*, dan *mutation rate* yang berbeda sedangkan parameter jumlah individunya tetap. Pengujian kinerja ini dilakukan dengan 4 percobaan sehingga hubungan antara perubahan jumlah generasi, *crossover rate*, dan *mutation rate* dengan waktu yang dibutuhkan dalam satu kali proses dan banyaknya kesalahan jadwal yang dihasilkan dapat dilihat pada Tabel 1 berikut.

**Tabel 1.** Hasil Percobaan Generate Jadwal

Percobaan	Jumlah Individu	Jumlah Generasi	CR	MR	Waktu (detik)	Total Kesalahan/Persentase
1	4	50	0.9	0.3	273.6	7 / 0.03%
2	4	50	0.9	0.8	190.11	44 / 0.17%
3	4	50	0.4	0.3	198.87	22 / 0.09%

4	4	85	0.9	0.3	337.73	16 / 0.06%
---	---	----	-----	-----	--------	------------

Percobaan kedua menunjukkan persentase kesalahan meningkat menjadi 0.17% daripada percobaan pertama. Percobaan kedua ini dilakukan dengan menaikkan nilai *mutation rate* sehingga menghasilkan lebih banyak kesalahan dari percobaan pertama. Peningkatan nilai *mutation rate* menyebabkan lebih banyak individu yang mengalami mutasi pada setiap generasi sehingga meningkatkan kemungkinan merusak kembali solusi yang sudah baik. Hal ini menunjukkan bahwa nilai *mutation rate* yang terlalu tinggi dapat mengganggu kestabilan solusi yang optimal dalam *Algoritma Genetika*.

Percobaan ketiga menunjukkan persentase kesalahan yang juga meningkat menjadi 0.09% daripada percobaan pertama tetapi persentasenya menurun daripada percobaan kedua. Percobaan ketiga ini dilakukan dengan menurunkan nilai *crossover rate* yang menyebabkan semakin sedikit individu yang dipindah silangkan.

Percobaan keempat menunjukkan persentase kesalahan dua kali lipat daripada percobaan pertama yaitu sebesar 0.06 persen. Percobaan keempat ini dilakukan dengan menaikkan jumlah generasi menjadi 85. Hasil ini lebih baik daripada percobaan kedua dan ketiga akan tetapi waktu yang dibutuhkan untuk percobaan keempat ini yang paling lama yaitu 337.73 detik. Hal ini menunjukkan bahwa waktu yang dibutuhkan akan semakin lama apabila jumlah generasi semakin banyak.

Berdasarkan empat percobaan tersebut dapat disimpulkan bahwa nilai *crossover rate* yang semakin tinggi dan nilai *mutation rate* yang semakin rendah menghasilkan solusi penjadwalan yang lebih baik dengan waktu yang tidak terlalu lama apabila jumlah generasi tidak terlalu banyak. Jumlah generasi yang semakin ditingkatkan akan membuat waktu yang dibutuhkan semakin lama.

Total kesalahan jadwal pada Tabel 2 dihitung berdasarkan kategori kesalahan berikut ini.

- Kesalahan apabila terdapat pemilihan waktu yang tidak berurutan
- Kesalahan dalam pemilihan hari apabila terdapat hari yang sudah dipilih oleh dosen sebagai hari ketidaksediannya mengajar
- Kesalahan apabila terdapat mata kuliah praktikum yang tidak di ruang laboratorium
- Kesalahan apabila terdapat jadwal untuk hari Jum'at pada jam 12:00-12:50
- Kesalahan apabila terdapat mata kuliah yang memiliki ruangan yang ditetapkan yang ditempatkan pada ruangan yang tidak sesuai

#### 4. Kesimpulan

*Algoritma Genetika* dapat diterapkan pada sistem penjadwalan kuliah untuk menghasilkan jadwal yang telah disinkronkan dengan proses peminjaman alat pendukung PBM, sehingga mempermudah pekerjaan berbagai pihak yang terlibat. Hasil penelitian menunjukkan bahwa *Algoritma Genetika* memberikan solusi paling optimal ketika nilai *crossover rate* ditingkatkan dan *mutation rate* diturunkan, dengan tingkat kesalahan hanya sebesar 0,03% dan waktu pemrosesan yang relatif singkat, yaitu 273,6 detik. Selain itu, hasil pengujian fungsionalitas sistem menggunakan metode *Black Box* menunjukkan tingkat keberhasilan sebesar 95,65%, yang menandakan bahwa sistem ini telah memenuhi persyaratan fungsional dan layak untuk digunakan.

## Referensi

- [1] M. Lukman Hakim and M. Hasibuan, "Penerapan Metode Simulated Annealing Untuk Penjadwalan Perkuliahan," *Comput. Technol. Inf. Syst.*, vol. 5, no. 2, pp. 25–38, Jun. 2021, Accessed: Nov. 16, 2023. [Online]. Available: <https://jurnal.stikommedan.ac.id/index.php/ctis/article/view/37>
- [2] T. Handayani, D. H. Fudholi, and S. Rani, "Kajian Algoritma Optimasi Penjadwalan Mata Kuliah," *PETIR*, vol. 13, no. 2, 2020, doi: 10.33322/petir.v13i2.1027.
- [3] S. Katoch, S. S. Chauhan, and V. Kumar, "A review on genetic algorithm: past, present, and future," *Multimed. Tools Appl.*, vol. 80, no. 5, 2021, doi: 10.1007/s11042-020-10139-6.
- [4] A. N. Toscani and R. Roestam, "Pengembangan Sistem Penjadwalan Kuliah Menggunakan Algoritma Genetik (Studi Kasus : Pascasarjana Universitas Jambi)," *J. Manaj. Sist. Inf.*, vol. 2, no. 2, 2017. M. Wegmuller, J. P. von der Weid, P. Oberson, and N. Gisin, "High resolution fiber distributed measurements with coherent OFDR," in *Proc. ECOC'00*, 2000, paper 11.3.4, p. 109.
- [5] D. Setiawan et al., "Implementasi *Algoritma Genetika* Untuk Prediksi Penyakit Autoimun," *Rabit J. Teknol. dan Sist. Inf. Univrab*, vol. 4, no. 1, pp. 8–19, Jan. 2019, doi: 10.36341/RABIT.V4I1.595.
- [6] H. Fang, "Genetic Algorithms for Scheduling," 1992, Accessed: Nov. 26, 2023. [Online]. Available: <https://www.dai.ed.ac.uk/papers/documents/mt92115.html>
- [7] A. Desiani and M. Arhami, "Konsep Kecerdasan Buatan," 1st ed., D. Hardjono, Ed., Andi Offset, 2006, pp. 187–225.
- [8] A. Boudjeline, I. A. Chaudhry, A. F. Rafique, I. A. Q. Elbadawi, M. Aichouni, and M. Boujelbene, "Multi-Objective Flexible Job Shop Scheduling Using Genetic Algorithms," *Teh. Vjesn.*, vol. 29, no. 5, 2022, doi: 10.17559/TV-20211022164333.
- [9] A. R. Juwita, A. R. Pratama, and T. Triono, "Implementasi Algoritma Particle Swarm Optimization untuk Penjadwalan Perkuliahan di Fakultas Teknik Dan Ilmu Komputer Universitas Buana Perjuangan Karawang," *J. SISFOTEK Glob.*, vol. 10, no. 1, 2020, doi: 10.38101/sisfotek.v10i1.270.
- [10] M. NUGRAHA and J. YASKURNIAAM, "Sistem Informasi Peminjaman Barang Berbasis Web dengan Metode Waterfall," *MIND J.*, vol. 5, no. 1, 2020, doi: 10.26760/mindjournal.v5i1.14-23.
- [11] S. Thakare, T. Nikam, and P. M. Patil, "Automated Timetable Generation using Genetic Algorithm," *Int. J. Eng. Res. Technol.*, vol. 9, no. 07, 2020, Accessed: Nov. 26, 2023. [Online]. Available: <http://secretgeek.net/content/bambrilg.pdf>
- [12] B. Samuel and J. Mathew, "Resource Allocation in a Repetitive Project Scheduling Using Genetic Algorithm," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 330, 2018, doi: 10.1088/1757-899X/330/1/012098.
- [13] I. A. Ashari, "Perbandingan Performansi *Algoritma Genetika* dan Algoritma Ant Colony Optimization dalam Optimasi Penjadwalan Mata Kuliah," *Repos. Univ. Negeri Semarang*, 2016.
- [14] N. Rochmawati and E. R. Saputra, "Perancangan Sistem Informasi Inventaris Untuk Peminjaman Dan Pengembalian Barang Di Laboratorium Jurusan Teknik Informatika Ft Unesa," *Surabaya Univ. Negeri Surabaya*, vol. 2, no. 1, 2016.
- [15] M. Faizal and D. A. Januar, "Aplikasi Sistem Perbantuan Komputer untuk Peminjaman Barang," *J. Teknol. Inf. dan Komun.*, 2013.