

## PENGIMPLEMENTASIAN ALGORITMA GENETIKA DALAM SISTEM PENJADWALAN PRAKTIKUM

Atong Nazarius, Ryan Delon Pratama, Rafael Aryapati Soebagijo,  
Ressa Priskila, Viktor Handrianus Pranatawijaya  
Teknik Informatika, Universitas Palangka Raya  
Jl. Yos Sudarso Palangka Raya, Indonesia  
atongnazarius2022@mhs.eng.upr.ac.id

### ABSTRAK

Penelitian ini melakukan pengimplementasian algoritma genetika pada sistem penjadwalan praktikum untuk mencapai solusi yang akurat dan efisien. Proses inisialisasi populasi dilakukan dengan menghasilkan kromosom yang berisi gen-gen yang merepresentasikan kombinasi data mata kuliah, asisten praktikum, jam, hari, dan ruangan. Evaluasi *fitness* dilakukan dengan menggunakan rumus  $Fitness\ Poin = 1/(1 + Penalty)$  untuk menentukan seberapa baik suatu individu dalam memenuhi kriteria yang telah ditetapkan. Percobaan menunjukkan kromosom terbaik bernilai *fitness* = 1, dengan populasi, generasi, dan tingkat mutasi yang semakin besar akan membutuhkan waktu eksekusi yang lebih lama dan sebaliknya. Juga ukuran populasi, generasi dan tingkat mutasi juga mempengaruhi kemungkinan untuk ditemukan hasil yang optimal. Hasil penelitian ini didapatkan bahwa algoritma genetika efektif dalam proses penyelesaian permasalahan penjadwalan praktikum yang kompleks, memberikan jadwal praktikum yang akurat dan efisien. Implikasi penelitian ini dapat menjadi acuan penting dalam pengembangan algoritma genetika khususnya dalam bidang penjadwalan praktikum.

**Kata kunci :** Penjadwalan, Algoritma Genetika

### 1. PENDAHULUAN

Pada perkembangan zaman saat ini, teknologi berkembang sangat pesat, sehingga pemanfaatan teknologi informasi sangat diperlukan, agar suatu kegiatan yang dilakukan dapat berjalan secara optimal. Dalam suatu kegiatan pasti diperlukan yang namanya sebuah penjadwalan, yang di mana terkadang suatu sistem penjadwalan tersebut mengalami bentrok dengan jadwal lain, seperti yang terjadi pada mahasiswa yang memiliki mata kuliah berpraktikum, di mana terkadang terjadinya bentrok antara jadwal praktikum dengan mata kuliah yang sedang dilaksanakan[1].

Di dalam pelaksanaan penjadwalan terutama penjadwalan praktikum masih sering dilakukan secara manual, salah satunya dengan menggunakan *microsoft excel* yang masih kurang efektif dan sering terjadi kesalahan dalam sistem penjadwalannya[2][3].

Maka daripada itu dikembangkanlah sebuah sistem penjadwalan praktikum ini. Dalam sistem penjadwalan yang dibuat, diterapkan juga sebuah algoritma genetika di dalamnya, agar sistem penjadwalan yang dibuat dapat bekerja secara optimal dan dapat menghasilkan generasi yang terbaik[4][5].

Pengimplementasian algoritma genetika menekankan pada fleksibilitas dalam proses menghasilkan jadwal yang optimal dan efektif dengan melakukan pengidentifikasian terhadap kendala[6].

Tujuan dari penelitian ini adalah mempermudah sistem administrasi dalam penentuan jadwal praktikum mahasiswa sehingga bisa mendapatkan hasil yang optimal, sehingga tidak terjadi bentrok antar jadwal mata kuliah, dan praktikum, serta dapat melakukan penerapan algoritma genetika.

### 2. TINJAUAN PUSTAKA

#### 2.1. Penelitian Terkait

Penelitian tentang “Sistem Informasi Penjadwalan Mata Pelajaran Berbasis Web Menggunakan Algoritma Genetika Pada SMKN 6 Majene” membahas mengenai pengembangan sistem penjadwalan mata pelajaran berbasis web tersebut. Di mana sistem tersebut berguna untuk mengatasi kesulitan dalam penjadwalan secara efisien. Di mana *output* yang dihasilkan melalui analisis algoritma genetika, dapat memberikan jadwal yang dapat diakses oleh staf, guru, dan siswa[7].

Penelitian tentang “Optimasi Penjadwalan Proyek Dengan Metode Algoritma Genetika” menggunakan pendekatan System Development Life Cycle dengan metode *waterfall* yang dengan hal tersebut dapat mengetahui alur perkembangan dari proyek yang dibuat, di mana jika dilihat pada penelitian yang dilakukan, bahwa algoritma genetika yang diterapkan pada sistemnya juga menghasilkan hasil yang optimal dalam optimasi penjadwalan proyek tersebut[1].

Dua tahun sebelum penelitian di atas, pernah dilakukan juga penelitian yang membahas tentang “Penerapan Algoritma Genetika Untuk Optimasi Penjadwalan pada MTS Negeri 1 Pangkalpinang” di mana penelitian tersebut bertujuan untuk melakukan pemanfaatan terhadap algoritma genetika dalam proses mencari jadwal yang efisien, di mana pendekatan yang dilakukan berdasar pada pengujian mata Pelajaran yang berbeda, yaitu 15, 20, 25, dan 30 mata Pelajaran[8].

Penelitian ini didasarkan pada penggunaan algoritma evolusioner, khususnya algoritma genetika

sebagai upaya dalam proses menyelesaikan masalah penjadwalan jadwal kursus akademik[6].

## 2.2. Penjadwalan

Penjadwalan merupakan salah satu proses alokasi sumber daya, di dalam proses menyelesaikan serangkaian pekerjaan dengan mempertimbangkan waktu penyelesaian, biaya dan ketersediaan terhadap sumber daya[9].

Penjadwalan merupakan masalah *optimisasi* kombinatorial yang di mana sulit diselesaikan jika menggunakan algoritma klasik, maka daripada itu penelitian ini memanfaatkan pendekatan dengan kecerdasan komputasional salah satunya algoritma genetika, untuk proses optimalisasi penjadwalan dalam mengajar[10].

Penjadwalan merupakan sebuah metode dalam proses penentuan waktu kerja yang dibutuhkan dalam melakukan pekerjaan dengan berbagai macam serangkaian kerja yang dilakukan[3].

## 2.3. Algoritma Genetika

Algoritma genetika merupakan pendekatan pencarian *heuristic* berbasis mekanisme alam dan genetika yang dapat diterapkan pada beragam masalah optimasi kompleks yang sulit diselesaikan oleh metode konvensional[5]. Algoritma Genetika sebagai pendekatan dalam upaya menciptakan jadwal yang optimal dan efisien dengan mempertimbangkan batasan-batasan yang ada[6].

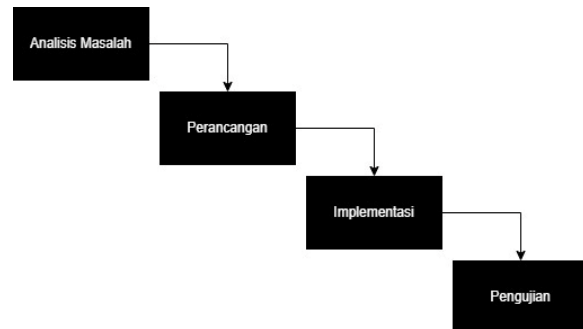
Algoritma genetika merupakan algoritma yang secara otomatis dapat melakukan penentuan sumber daya yang digunakan untuk dapat meminimalkan waktu yang diperlukan[11].

Algoritma Genetika merupakan algoritma pencarian yang berdasar pada seleksi secara alami dan genetika alamiah[12].

Mekanisme dasar yang menginspirasi algoritma genetika terinspirasi dari evolusi biologis, di mana populasi berevolusi melalui seleksi dan variasi. Algoritma ini menggunakan mekanisme seperti seleksi, rekombinasi, dan mutasi untuk menghasilkan solusi baru terhadap masalah dari sebuah populasi solusi yang ada[6][12].

## 3. METODE PENELITIAN

Dalam penelitian ini, metode yang digunakan adalah metode *waterfall*, dimana metode *waterfall* merupakan salah satu *Software Development Life Cycle* (SDLC) yang menggunakan bentuk pola aliran, seperti air terjun. Di mana pada metode ini setiap tahapan harus diselesaikan terlebih dahulu sebelum melanjutkan ke tahapan selanjutnya[13]. Berikut merupakan gambaran alur penelitian dengan menggunakan metode *waterfall*.



Gambar 1. Alur penelitian (metode *waterfall*)

### 3.1. Analisis

Dengan mengacu pada metode *waterfall* hal utama yang perlu dilakukan adalah analisis, dimana pada tahap ini perolehan informasi yang dibutuhkan dalam pembuatan sistem[3][14]. analisis ini dibagi menjadi dua bagian yaitu analisis masalah dan analisis sistem[15].

#### 3.1.1. Analisis Masalah

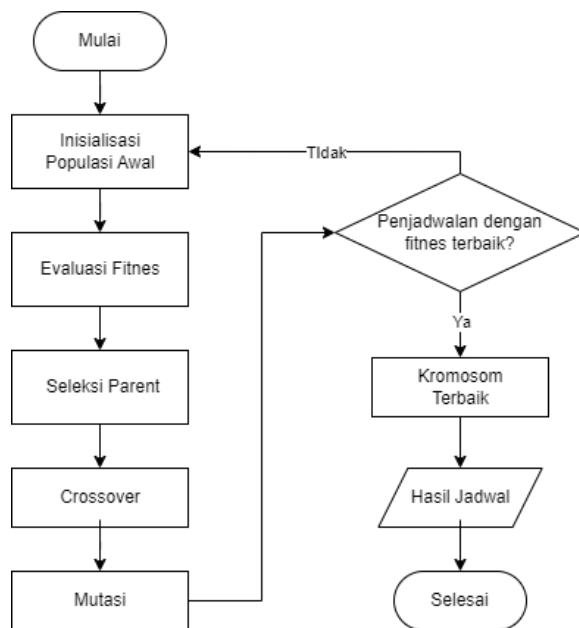
Pada tahap ini merupakan bagian yang bertujuan untuk mengetahui permasalahan yang terjadi pada penjadwalan praktikum, di mana didapatkan dari data yang diterima bahwa pembuatan jadwal praktikum dilakukan oleh admin program studi yang dilakukan secara manual, hal ini berakibat pada lamanya hasil jadwal praktikum yang dibuat dan rawan sekali terjadi bentrok jadwal jika kurang teliti. Hal ini disebabkan oleh banyaknya mahasiswa dan juga keterbatasan waktu yang tersedia untuk praktikum, di mana ada 5 hari untuk praktikum di mana praktikum tidak boleh bentrok dengan mata kuliah, dan tidak jarang mahasiswa mendapatkan jadwal yang bentrok dengan jadwal perkuliahannya.

#### 3.1.2. Analisis Sistem

Hasil dari dilakukannya analisis permasalahan di atas akan digunakan sebagai data utama untuk mendapatkan solusi yang terbaik di mana dapat diselesaikan dengan membangun sebuah aplikasi berbasis *website* yang dapat digunakan untuk menghasilkan jadwal yang lebih efektif dan efisien, analisis sistem dilakukan dengan cara mengumpulkan semua data yang dapat di *input* pada sebuah penjadwalan di mana akan digunakan data asisten praktikum, data mata kuliah, data ruangan, dan data waktu.

### 3.2. Perancangan

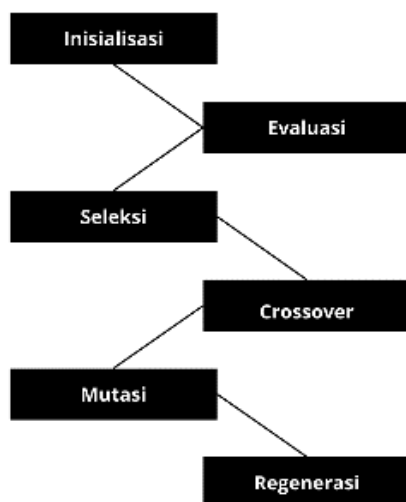
Pada tahap ini dilakukan agar memperoleh gambaran awal terhadap langkah selanjutnya yang akan dilakukan. Pada tahap ini akan dirancang sistem dengan menggunakan *flowchart* sebagai berikut



Gambar 2. Flowchart sistem algoritma genetika

Pada gambar 3. di atas merupakan *flowchart* dari sistem penjadwalan menggunakan algoritma genetika di mana proses awal dilakukan inisialisasi populasi dan kromosom, kemudian populasi tersebut di evaluasi sesuai nilai *fitness* yang ditentukan, setelah itu dilakukan seleksi parent yang memiliki nilai *fitness* tertinggi, setelah didapatkan dua parent dengan *fitness* tertinggi akan dilakukan crossover dan mutasi, kemudian akan dimunculkan hasilnya lalu di cek dengan kriteria kondisi nilai *fitness* yang diinginkan, dimana jika jadwal tidak memiliki nilai *fitness* terbaik maka akan dilakukan inisialisasi populasi awal kembali, sedangkan jika memenuhi nilai *fitness* terbaik maka jadwal diproses dan ditampilkan, setelah keluaran *output* jadwal kemudian selesai[16].

Selanjutnya akan dilakukan pembuatan struktur sistem penjadwalan praktikum, dengan alur kerja sistem sebagai berikut.



Gambar 3. Alur kerja sistem

Berdasarkan gambar 2. Maka akan dijelaskan tahapan-tahapannya sebagai berikut.

### 3.2.1. Inisialisasi

Inisialisasi populasi adalah tahap untuk membuat populasi awal dari individu secara acak. Setiap individu akan direpresentasikan dalam bentuk kromosom atau genom. Pada sistem ini, akan menggunakan data mata kuliah, data sistem praktikum, dan data ruangan.

#### Evaluasi

Evaluasi *fitness* Poin adalah tahap penilaian dan evaluasi individu berdasarkan suatu kriteria. Jika individu tidak memenuhi suatu kriteria, maka akan diberikan sebuah *penalty*. Pada proses ini, akan menggunakan rumus:  $Fitness\ Poin = 1/(1 + Penalty)$ .

### 3.2.2. Seleksi

Seleksi adalah tahap pemilihan individu-individu yang memiliki nilai *fitness* tertinggi. Pada proses ini, akan menentukan individu *parent* yang akan digunakan pada tahap berikutnya.

### 3.2.3. Crossover

Crossover adalah tahap menggabungkan informasi genetik dari dua atau lebih individu *parent* untuk menghasilkan keturunan baru, yaitu individu *child*.

### 3.2.4. Mutasi

Mutasi adalah tahap mengubah secara acak satu atau lebih gen dari individu *child* untuk menambah variasi genetik pada individu.

### 3.2.5. Regenerasi

Regenerasi adalah tahap pengulangan tahap-tahap sebelumnya. Pada proses ini, individu *child* akan dimasukkan kembali ke populasi dengan cara eliminasi individu yang memiliki *fitness* poin terendah. Setelah itu, populasi baru ini akan digunakan pada pengulangan sebagai inisialisasi populasi. Proses ini akan berlanjut hingga salah satu kriteria berhenti terpenuhi, yaitu maksimum generasi, atau solusi yang tepat telah ditemukan.

## 3.3. Implementasi

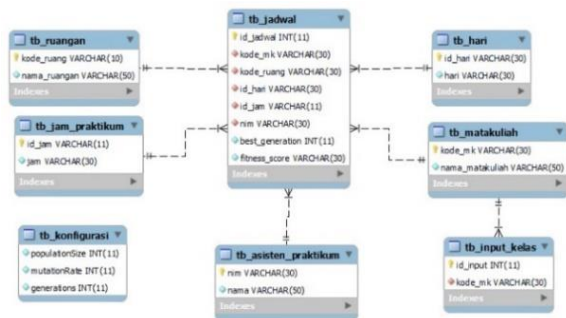
Pada tahap ini penulis melakukan pengkodean untuk mengimplementasikan algoritma genetika menggunakan bahasa pemrograman Web yaitu HTML, CSS dan PHP yang dikoneksikan secara langsung menggunakan *database* yang dijalankan menggunakan XAMPP 7.2. Untuk proses pengkodean digunakan teks editor Visual Studio Code. Pada proses pengkodean dirancang agar dapat menampilkan jadwal praktikum yang disusun dengan algoritma genetika, juga memberikan menu untuk *user* memodifikasi populasi dan *mutation rate*, sehingga aplikasi dapat digunakan sesuai kebutuhan.

### 3.4. Pengujian

Pengujian diperlukan untuk menemukan masalah atau kekurangan yang ada pada sistem, sehingga sistem dapat berjalan tanpa adanya cacat dan sesuai dengan kebutuhan dari pengguna. Pada tahap ini pengujian menggunakan metode *blackbox texting* di mana pengujian ini hanya bertujuan untuk melihat program tersebut apakah sesuai dengan fungsi yang diinginkan program tersebut tanpa mengetahui kode program yang dipakai[17].

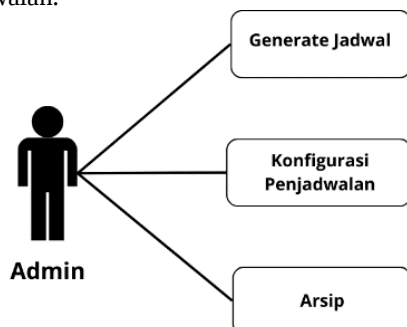
## 4. HASIL DAN PEMBAHASAN

Algoritma genetika memiliki dasar logika untuk proses seleksi secara alami atau genetik. Di bawah ini merupakan gambar ERD dari penjadwalan praktikum.



Gambar 4. Entity Relationship Diagram (ERD)

Algoritma genetika sendiri dianalogikan sebagai suatu seleksi alam dari banyaknya individu, yang merepresentasikan sebuah solusi terbaik yang muncul dari masalah yang dihadapi[18]. Di bawah ini merupakan gambar untuk *use case diagram* penjadwalan.



Gambar 5. Use case diagram penjadwalan

Memuat hasil, Pengujian dan pembahasan tentang skripsi yang telah dilakukan

Tabel 1. Keterangan *use case diagram* sistem penjadwalan

| No. | Use case                | Keterangan                                                   |
|-----|-------------------------|--------------------------------------------------------------|
| 1   | Generate Jadwal         | Tampilan hasil jadwal yang sudah melalui proses perhitungan. |
| 2   | Konfigurasi Penjadwalan | Tampilan untuk mengatur Populasi, mutasi, dan generasi.      |
| 3   | Arsip                   | Tampilan hasil dari <i>generate</i> jadwal.                  |

Tabel 2. Data mata kuliah praktikum

| No. | Nama mata kuliah             | Kode mata kuliah |
|-----|------------------------------|------------------|
| 1   | Basis Data II                | 1DCP584032       |
| 2   | Jaringan Komputer            | 1DCP314032       |
| 3   | Pemrograman Web dan Mobile   | 1DCP644032       |
| 4   | Algoritma dan Pemrograman II | 1DCP092032       |
| 5   | Struktur Data                | 1DCP182032       |

Tabel 3. Data asisten praktikum

| No. | NIM          | Nama                              |
|-----|--------------|-----------------------------------|
| 1   | 213010503003 | Kevin Wilbert Sachio              |
| 2   | 213020503061 | Gabriel Dwi Putra Krisma Rusan    |
| 3   | 213010503014 | Evita Cahyani                     |
| 4   | 213020503058 | Jordan Setiawan Nanyan            |
| 5   | 213020503018 | Aldoni Fahreza                    |
| 6   | 213020503020 | Bayu Pratama                      |
| 7   | 213020503027 | Recal Prasetyo                    |
| 8   | 213030503131 | Samuel Raka Yustianto             |
| 9   | 213030503123 | Elieser Dawson Fritz Simangunsong |
| 10  | 213010503008 | Via Windiana                      |
| 11  | 213030503120 | Hardi                             |
| 12  | 203020503044 | Joshua Evan Savero                |
| 13  | 203010503009 | Resha Ananda Rahman               |
| 14  | 213030503098 | Samuel Fernando                   |
| 15  | 203010503019 | Rholand Deo Eka Putra             |
| 16  | 213030503108 | Ricard Jonathan                   |
| 17  | 213030503133 | Risky Prasetyo                    |
| 18  | 213030503116 | okta panji winata                 |
| 19  | 213030503160 | Thomas Richard Wini Ngele         |
| 20  | 213030503111 | Michael Henokh                    |
| 21  | 213030503156 | Dedan Adam                        |
| 22  | 213030503104 | Euriqo Diaz                       |
| 23  | 213020503083 | Mohammad Alif Tuharea             |
| 24  | DBC117087    | Yoga Pratama Salim                |
| 25  | DBC117085    | Gabriel Padma                     |
| 26  | DBC117040    | Harits Wahid Vijayanto            |

Tabel 4. Data ruangan

| No. | Kode ruang | Keterangan        |
|-----|------------|-------------------|
| 1.  | R01        | Lab Data Science  |
| 2.  | R02        | Lab Pemrograman 1 |
| 3.  | R03        | Lab Pemrograman 2 |
| 4.  | R04        | Lab Jaringan      |

Tabel 5. Jam praktikum

| No | Id jam | Jam           |
|----|--------|---------------|
| 1. | T01    | 07:00 – 09:00 |
| 2. | T02    | 09:00 – 12:00 |
| 3. | T03    | 12:00 – 14:00 |
| 4. | T04    | 15:00 – 17:00 |

Tabel 6. Hari praktikum

| No. | Kode hari | Hari   |
|-----|-----------|--------|
| 1.  | H1        | Senin  |
| 2.  | H2        | Selasa |
| 3.  | H3        | Rabu   |
| 4.  | H4        | Kamis  |
| 5.  | H5        | Jumat  |

#### 4.1. Pembentukan Kromosom dan Populasi

Pembentukan kromosom dilakukan dengan dilakukan inisialisasi data secara acak yang di simpan dalam *array* untuk menghasilkan kromosom, di mana kromosom tersebut berisikan kumpulan gen yang berisi beberapa kombinasi data berupa mata kuliah, asisten praktikum, jam, hari dan ruangan. Setelah itu akan dibentuk populasi awal, diasumsikan terdapat 2 kromosom yang di dalamnya terdiri dari 4 gen yang merupakan kombinasi dari mata kuliah, asisten praktikum, jam, hari, dan ruangan.

Tabel 7. Populasi

| Gen | Kromosom 1 | Kromosom 2 | Kromosom 3 |
|-----|------------|------------|------------|
| 1   | 4,18,4,5,3 | 5,12,4,1,3 | 2,13,2,5,3 |
| 2   | 2,10,1,1,4 | 4,14,1,3,3 | 1,20,2,4,3 |
| 3   | 5,7,1,4,1  | 3,18,3,4,1 | 3,24,4,5,4 |
| 4   | 1,7,4,5,3  | 1,23,1,3,3 | 5,10,4,5,1 |
| 5   | 3,8,1,3,4  | 2,4,1,3,3  | 4,8,4,4,2  |

#### 4.2. Evaluasi Nilai Fitness

Proses evaluasi *fitness* pada penelitian ini bertujuan untuk menentukan seberapa baik suatu individu dalam hal memenuhi sejumlah kriteria yang telah ditetapkan. Pada proses ini akan menggunakan rumus  $Fitness\ Poin = 1/(1 + Penalty)$  untuk menentukan nilai *fitness* dari individunya. *Penalty* untuk rumus *fitness* poinnya ditentukan berdasarkan kriteria-kriteria berikut.

Tabel 8. Kondisi *fitness*

| No | Kondisi                                                                                 | Penalty          |
|----|-----------------------------------------------------------------------------------------|------------------|
| 1. | Jika dua mata kuliah praktikum mempunyai ruangan dan jam yang sama pada hari yang sama. | <i>Penalty++</i> |
| 2. | Jika ada mata kuliah praktikum pada hari Jumat pukul 12.00.                             | <i>Penalty++</i> |
| 3. | Jika ada asisten praktikum pada ruangan, slot waktu, dan hari yang sama.                | <i>Penalty++</i> |
| 4. | Jika seorang asisten praktikum mengajar lebih dari dua kali.                            | <i>Penalty++</i> |
| 5. | Jika mata kuliah praktikum yang sama muncul lebih dari 4 kali.                          | <i>Penalty++</i> |

#### 4.3. Seleksi Parent

Proses seleksi pada penelitian ini menggunakan metode seleksi *roulette wheel*. Metode ini mensimulasikan *roulette wheel* dimana setiap individu dalam populasi diberi bagian roda yang sebanding dengan nilai *fitness*nya[12]. Tujuan dari metode ini adalah untuk memilih *parent* berdasarkan nilai *fitness* nya. Dalam proses seleksi, individu yang memiliki nilai *fitness* lebih tinggi berpeluang lebih besar untuk terpilih sebagai *parent*, namun individu dengan nilai *fitness* lebih rendah juga dapat terpilih sebagai *parent*, meskipun dengan probabilitas yang lebih rendah. Pada proses ini, nilai *fitness* keseluruhan populasi akan dihitung. Kemudian nomor acak akan dipilih antara 0 dan total nilai *fitness*. Selama iterasi pada populasi, total dari nilai *fitness* individu yang sedang

dipertimbangkan akan terus dihitung. Jika nilai *fitness* individu melebihi nomor acak yang dipilih sebelumnya, individu tersebut akan dipilih sebagai *parent*. Proses ini akan diulangi sampai dua *parent* terpilih untuk disilangkan (*Crossover*).

Tabel 9. Perhitungan total nilai *fitness*

| Kromosom (individu)        | Nilai <i>fitness</i> |
|----------------------------|----------------------|
| 1                          | 0,5                  |
| 2                          | 0,5                  |
| 3                          | 1                    |
| Total Nilai <i>Fitness</i> | 2                    |

Pada tabel ini terdapat nilai *fitness* dari masing-masing kromosom nya dan total dari nilai *fitness* nya. Berikutnya akan ditentukan nomor acak nya berdasarkan antara 0 dan total nilai *fitness* nya. Misalkan jika didapat nomor acak sebesar 0.7, maka akan dibandingkan nilai-nilai *fitness* nya dengan nomor acak tersebut. Jika nilai *fitness* individu melebihi nomor acak tersebut maka akan dipilih sebagai *parent*, yaitu pada tabel ini adalah kromosom nomor 3 dan untuk *parent* berikutnya akan dipilih antara kromosom 2 dan kromosom 1.

#### 4.4. Crossover

Proses *crossover* pada penelitian ini menggunakan metode *single-point Crossover* untuk menghasilkan individu *child* dari dua *parent*. Pada metode ini setiap gen pada satu individu *parent* akan dipertukarkan dengan gen pada individu *parent* lain untuk membuat individu baru melalui satu titik potong[19]. Proses pada *Crossover* ini diawali dengan gen dari *parent* pertama dan kedua disalin ke dalam variabel lokal. Kemudian, titik pisah dipilih secara acak antara 1 dan jumlah gen pada individu *parent* pertama. Selanjutnya, gen dari orang tua pertama sebelum titik pembagian akan ditambahkan ke individu *child* baru, lalu gen dari *parent* kedua setelah titik pembagiannya akan ditambahkan juga ke individu *child* baru.

Tabel 10. Crossover

| Gen | Parent 1   | Parent 2   | Child      |
|-----|------------|------------|------------|
| 1   | 2,13,2,5,3 | 5,12,4,1,3 | 2,13,2,5,3 |
| 2   | 1,20,2,4,3 | 4,14,1,3,3 | 1,20,2,4,3 |
| 3   | 3,24,4,5,4 | 3,18,3,4,1 | 3,18,3,4,1 |
| 4   | 5,10,4,5,1 | 1,23,1,3,3 | 5,12,4,5,1 |
| 5   | 4,8,4,4,2  | 2,4,1,3,3  | 4,8,4,4,2  |

Pada tabel ini terdapat gen 1 sampai gen 5 dari *parent* 1 dan *parent* 2 yang digunakan untuk mencari individu *child* nya melalui proses *crossover*. Pada *crossover* ini akan menggunakan *split point* acak yaitu 3, maka gen akan di swap atau di *crossover* dari *split point* sampai akhir, sehingga menghasilkan individu *child* dari penggabungan gen dari dua individu *parent* tersebut.

#### 4.5. Mutasi

Proses mutasi pada penelitian ini akan bertujuan untuk memperkenalkan variasi genetik ke dalam suatu

populasi melalui mutasi individu *child*. Proses ini diawali dengan setiap gen pada individu *child* akan diperiksa untuk mengetahui apakah ada mutasi pada gen tersebut. Untuk melakukan ini, nilai acak antara 0 dan 100 dihasilkan dan diperiksa apakah nilai ini lebih kecil dari nilai probabilitas *mutase*, yaitu *mutationRate*. Dalam hal ini terjadi mutasi pada gen. Gen yang bermutasi akan diperbarui menggunakan data acak dari kumpulan data yang sesuai, seperti ruang, waktu, asisten praktikum, dan hari. Proses ini dilakukan untuk setiap gen pada individu *child*, sehingga berpotensi menciptakan variasi genetik baru dalam populasi.

Tabel 11. Mutasi

| Sebelum    | Sesudah    |
|------------|------------|
| 1,20,2,4,3 | 2,20,2,4,3 |

Pada tabel ini terdapat gen 2 dari individu *child*. Gen 2 ini mengalami mutasi dikarenakan nilai acak dari 0 hingga 100 kurang dari atau sama dengan 10 (10% dari 100), sehingga atribut pada gen tersebut akan dimutasi, yang dalam gen ini adalah atribut mata kuliah praktikum.

#### 4.6. Regenerasi

Proses regenerasi pada penelitian ini akan menggunakan pendekatan elitism. Pendekatan elitism atau operator elitism adalah operator yang digunakan untuk mempertahankan individu terbaik[20]. Proses ini diawali dengan tiga individu teratas dari populasi sebelumnya tidak berubah dan pemilihan *parent* dilakukan untuk setiap pasangan dalam populasi saat ini. Melalui proses *Crossover*, gen dari kedua *parent* akan dipertukarkan untuk menghasilkan *child* baru, dan selanjutnya terjadi mutasi. Langkah-langkah ini akan diulangi hingga populasi baru mencapai jumlah yang diinginkan. Populasi baru ini terdiri dari individu-individu elit dari generasi sebelumnya dan keturunan yang dihasilkan dari *Crossover* dan mutasi, yang mewakili variasi genetik dalam regenerasi populasi.

#### 4.7. Proses findSolution

Proses *findSolution* pada penelitian ini bertujuan untuk mencari solusi atau penjadwalan terbaik dalam rentang jumlah generasi (*generations*) tertentu. Proses ini akan diawali dengan inisialisasi populasi awal menggunakan metode *initPopulation()*. Kemudian, pada setiap iterasi generasi (*generations*), dilakukan proses regenerasi untuk menghasilkan populasi baru melalui proses *Crossover* dan mutasi. Selama proses ini, jika ditemukan jadwal dengan nilai *fitness* lebih tinggi, jadwal optimal akan diperbarui. Pada akhir iterasi, solusi terbaik ditampilkan beserta generasi di mana solusi tersebut ditemukan, yaitu *bestgeneration*. Selanjutnya, solusi terbaik akan disimpan di tabel 'tb\_jadwal' menggunakan metode *saveBestSchedule()*. Dengan demikian, fungsi ini mengintegrasikan langkah-langkah pencarian solusi penjadwalan dengan regenerasi populasi dalam

mencari solusi penjadwalan terbaik untuk masalah penjadwalan yang dihadapi.

#### 4.8. Pengujian

Pada proses ini, akan dilakukan pengujian yang bertujuan untuk mengecek keberadaan penjadwalan dengan solusi terbaik atau optimal, yaitu bernilai *fitness* satu atau bernilai *fitness* mendekati satu, dengan menggunakan konfigurasi ukuran populasi (*Population Size*), generasi (*Generations*), dan tingkat mutasi (*Mutation Rate*). Pada pengujian pertama akan digunakan konfigurasi *Population Size* = 20, *Mutation Rate* = 10, dan *Generations* = 100, serta akan dilakukan *generate* jadwal sebanyak lima kali.

Tabel 12. Pengujian pertama

| Jadwal | Nilai <i>fitness</i> | Waktu eksekusi | Generasi terbaik |
|--------|----------------------|----------------|------------------|
| 1      | 1                    | 4,50 Detik     | 26               |
| 2      | 1                    | 5,52 Detik     | 18               |
| 3      | 0,5                  | 2,20 Detik     | 1                |
| 4      | 0,5                  | 3,62 Detik     | 2                |
| 5      | 1                    | 7,85 Detik     | 17               |

Berikutnya, pada pengujian kedua akan digunakan konfigurasi *Population Size* = 40, *Mutation Rate* = 20, dan *Generations* = 200, serta akan dilakukan *generate* jadwal sebanyak lima kali.

Tabel 13. Pengujian kedua

| Jadwal | Nilai <i>fitness</i> | Waktu eksekusi | Generasi terbaik |
|--------|----------------------|----------------|------------------|
| 1      | 1                    | 36,79 Detik    | 76               |
| 2      | 1                    | 27,10 Detik    | 13               |
| 3      | 1                    | 45,0 Detik     | 13               |
| 4      | 1                    | 40,77 Detik    | 27               |
| 5      | 1                    | 39,66 Detik    | 52               |

Berdasarkan kedua pengujian ini, digunakan konfigurasi yang berbeda. Pada pengujian pertama, menggunakan konfigurasi *Population Size* = 20, *Mutation Rate* = 10, dan *Generations* = 100. Hasilnya diperoleh nilai *fitness* antara 0,5 hingga 1, dengan waktu rata-rata untuk menghasilkan jadwal adalah 4,73 detik, dengan jumlah generasi yang bervariasi. Lalu, pada pengujian kedua akan menggunakan konfigurasi *Population Size* = 40, *Mutation Rate* = 20, dan *Generations* = 200.

Hasilnya diperoleh nilai *fitness* keseluruhan 1, dengan waktu rata-rata untuk menghasilkan jadwal adalah 37,86 detik, dengan jumlah generasi yang bervariasi juga. Dengan demikian dapat diperoleh bahwa ukuran populasi (*Population Size*) dan generasi (*Generations*) mempengaruhi waktu dari eksekusinya. Berdasarkan pengujian ini jika ukuran populasi dan generasinya semakin besar maka waktu eksekusinya menjadi lebih lama dan sebaliknya. Selain itu, ukuran populasi, generasi dan tingkat mutasi juga mempengaruhi kemungkinan untuk menemukan hasil yang optimal.

## 5. KESIMPULAN DAN SARAN

Diambil kesimpulan bahwa, Algoritma Genetika mampu menyelesaikan permasalahan penjadwalan yang rumit, dengan digunakannya algoritma genetika pada penjadwalan praktikum didapatkan hasil bahwa algoritma genetika dapat memberikan jadwal praktikum yang akurat dan efisien sehingga tidak terjadinya bentrok antar jadwal dari segi hari, waktu, ruang dan asisten praktikum yang mengajar. Dengan demikian, penelitian ini dapat menjadi acuan pendidikan dalam pengembangan algoritma genetika khususnya di dalam bidang penjadwalan.

## DAFTAR PUSTAKA

- [1] A. Rinaldi and A. A. Rismayadi, "Optimasi Penjadwalan Proyek Dengan Metode Algoritma Genetika," *eProsiding Tek. Inform.*, vol. 3, no. 1, pp. 107–120, 2022.
- [2] I. H. Sugeha, R. L. Inkiriwang, and P. A. K. Pratatis, "Optimasi Penjadwalan Menggunakan Metode Algoritma Genetika Pada Proyek Rehabilitasi Puskesmas Minanga," *J. Sipil Statik*, vol. 7, no. 12, pp. 1669–1680, 2019, [Online]. Available: <https://ejournal.unsrat.ac.id/v3/index.php/jss/article/view/26145>
- [3] D. Asmarajati, M. F. Asnawi, and R. D. Akmal, "Implementasi Algoritma Genetika Pada Penjadwalan Sistem informasi XYZ TV," *Riau J. Comput. Sci.*, vol. 06, no. 01, pp. 12–22, 2020.
- [4] Y. Pratama, "Optimalisasi Penjadwalan Karyawan Paruh Waktu Berdasarkan Nilai Fitness Terbaik Menggunakan Algoritma Genetika," *J. Nas. Inform.*, vol. 2, no. 2, pp. 114–142, 2021.
- [5] A. P. S. Iskandar, "Optimasi Penjadwalan Ujian Tugas Akhir Dengan Menggunakan Algoritma Genetika," *J. Comput. Sci. Informatics Eng.*, vol. 5, no. 1, pp. 40–48, 2021, doi: 10.29303/jcosine.v5i1.379.
- [6] I. A. Abduljabbar and S. M. Abdullah, "An evolutionary algorithm for solving academic courses timetable scheduling problem," *Baghdad Sci. J.*, vol. 19, no. 2, pp. 399–408, 2022, doi: 10.21123/BSJ.2022.19.2.0399.
- [7] P. Rpp, D. Pendekatan, and M. Realistik, "Pegguruang: Conference Series," vol. 1, no. September, 2019.
- [8] D. Wahyuningsih and E. Helmud, "Penerapan Algoritma Genetika Untuk Optimasi Penjadwalan pada MTS Negeri 1 Pangkalpinang," *J. Sisfokom (Sistem Inf. dan Komputer)*, vol. 9, no. 3, pp. 435–441, 2020, doi: 10.32736/sisfokom.v9i3.994.
- [9] D. Ghaniyyu Muqsit, S. Achmadi, and Y. Agus Pranoto, "Penerapan Algoritma Genetika Untuk Penjadwalan Mata Pelajaran," *JATI (Jurnal Mhs. Tek. Inform.)*, vol. 7, no. 5, pp. 2959–2966, 2024, doi: 10.36040/jati.v7i5.7569.
- [10] M. Elliot, S. Gbenga, and E. J. Mnisi, "Enhanced Heuristic Teaching Timetabling Algorithm Using Genetic algorithm," *Int. J. Sci. Technol. Res.*, vol. 9, no. 4, pp. 3804–3814, 2020, [Online]. Available: [www.ijstr.org](http://www.ijstr.org)
- [11] B. Tağtekin, M. U. Öztürk, and M. K. Sezer, "A Case Study: Using Genetic Algorithm for Job Scheduling Problem," 2021, [Online]. Available: <http://arxiv.org/abs/2106.04854>
- [12] H. Ardiansyah and M. B. S. Junianto, "Penerapan Algoritma Genetika untuk Penjadwalan Mata Pelajaran," *J. Media Inform. Budidarma*, vol. 6, no. 1, p. 329, 2022, doi: 10.30865/mib.v6i1.3418.
- [13] A. Wahid Abdul, "Analisis Metode Waterfall Untuk Pengembangan Sistem Informasi," *J. Ilmu-ilmu Inform. dan Manaj. STMIK*, no. November, pp. 1–5, 2020.
- [14] F. Mone and J. E. Simarmata, "Aplikasi Algoritma Genetika Dalam Penjadwalan Mata Kuliah," *BAREKENG J. Ilmu Mat. dan Terap.*, vol. 15, no. 4, pp. 615–628, 2021, doi: 10.30598/barekengvol15iss4pp615-628.
- [15] A. P. Pambudi *et al.*, "1051-Article Text-3263-1-10-20210914," vol. 8, no. 3, 2021.
- [16] A. Penjadwalan, M. Pelajaran, M. Furqan, and R. Ananda, "476-968-1-Sm," vol. 6, no. September, pp. 591–600, 2022.
- [17] I. Ismail and J. Efendi, "Black-Box Testing : Analisis Kualitas Aplikasi Source Code Bank Programming," *J. JTIK (Jurnal Teknol. Inf. dan Komunikasi)*, vol. 4, no. 2, p. 1, 2020, doi: 10.35870/jtik.v5i1.148.
- [18] P. A. Rizki and Y. Hendriyani, "Rancang Bangun Aplikasi Penjadwalan Mata Pelajaran SMK Muhammadiyah 1 Pekanbaru Berbasis Web Menggunakan Algoritma Genetika," vol. 7, pp. 24787–24798, 2023.
- [19] Eka Yulia Sari, Dina Yulina H, and Titik Rahmawati, "Pemodelan Sistem Optimasi Penjadwalan Matakuliah Dengan Algoritma Genetika," *Tek. Teknol. Inf. dan Multimed.*, vol. 4, no. 1, pp. 70–78, 2023, doi: 10.46764/teknimedia.v4i1.97.
- [20] S. Cokrowibowo, Ismail, and Indra, "Multiple Traveling Salesman Problem Menggunakan Algoritma Ant Colony Optimization dengan Operasi Elitism," *J. Comput. Inf. Syst. ( J-CIS )*, vol. 1, no. 2, pp. 23–28, 2019, doi: 10.31605/jcis.v1i2.619.