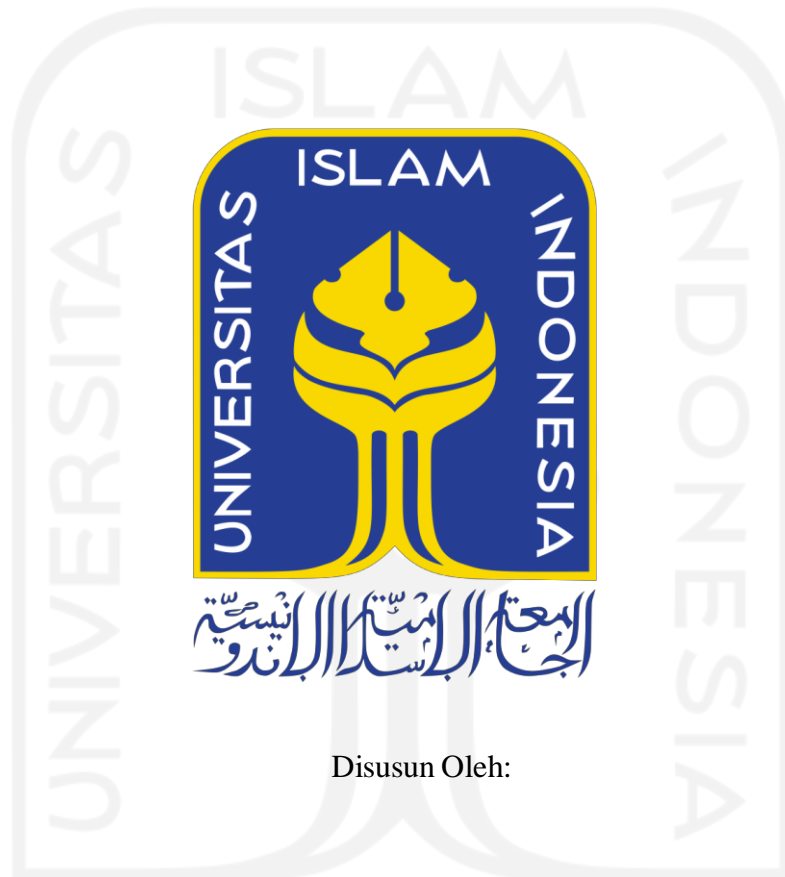


**SISTEM PENJADWALAN MATA KULIAH DARING UNTUK  
PROGRAM STUDI INFORMATIKA UNIVERSITAS ISLAM  
INDONESIA MENGGUNAKAN METODE ALGORITMA  
GENETIKA**



Disusun Oleh:

Nama : Hari Kurnia

NIM : 14523317

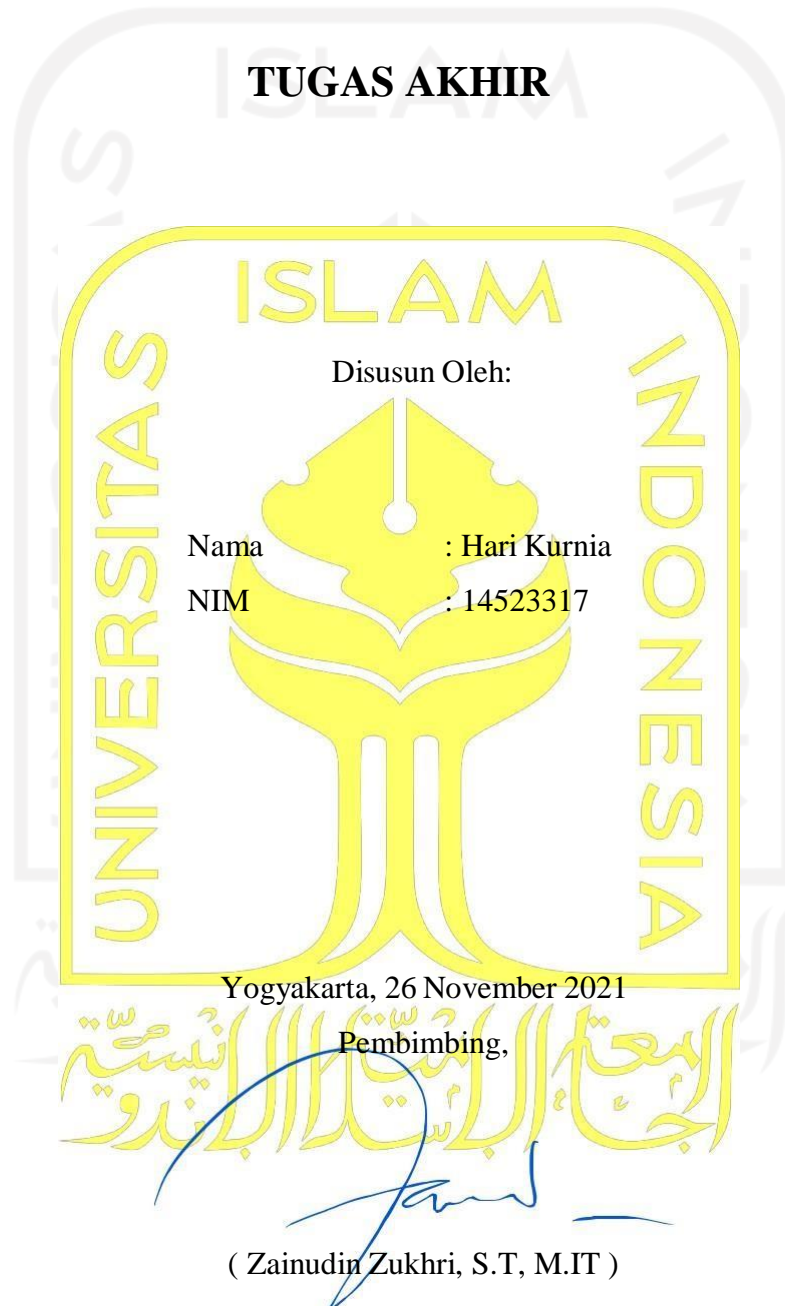
**PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA  
FAKULTAS TEKNOLOGI INDUSTRI  
UNIVERSITAS ISLAM INDONESIA**

**2021**

HALAMAN PENGESAHAN DOSEN PEMBIMBING

**SISTEM PENJADWALAN MATA KULIAH DARING UNTUK  
PROGRAM STUDI INFORMATIKA UNIVERSITAS ISLAM  
INDONESIA MENGGUNAKAN METODE ALGORITMA  
GENETIKA**

**TUGAS AKHIR**



## HALAMAN PENGESAHAN DOSEN PENGUJI

**SISTEM PENJADWALAN MATA KULIAH DARING UNTUK  
PROGRAM STUDI INFORMATIKA UNIVERSITAS ISLAM  
INDONESIA MENGGUNAKAN METODE ALGORITMA  
GENETIKA**

**TUGAS AKHIR**

Telah dipertahankan di depan sidang pengujian sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Informatika – Program Sarjana di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 27 Desember 2021

Tim Penguji

Zainudin Zukhri, S.T., M.IT

**Anggota 1**

Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D.

**Anggota 2**

Septia Rani, S.T., M.Cs.

Mengetahui,

Ketua Program Studi Informatika – Program Sarjana

Fakultas Teknologi Industri

Universitas Islam Indonesia



( Dr. Radeh Teduh Dirgahayu, S.T., M.Sc. )



## HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Hari Kurnia

NIM : 14523317

Tugas akhir dengan judul:

### **SISTEM PENJADWALAN MATA KULIAH DARING UNTUK PROGRAM STUDI INFORMATIKA UNIVERSITAS ISLAM INDONESIA MENGGUNAKAN METODE ALGORITMA GENETIKA**

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila di kemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung risiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 10 Desember 2021



(Hari Kurnia)

## HALAMAN PERSEMBAHAN

Puji dan syukur penulis panjatkan kepada Allah Subhanahuwata'ala yang telah memberikan penulis nikmat dan hidayah-Nya yang begitu besar dan tiada henti. Yang selalu ada jika penulis ingin bercerita tentang berbagai hal yang harus penulis lalui selama proses pembuatan Tugas Akhir ini, serta memberikan penulis kesehatan dan kekuatan sehingga dapat menyelesaikan Tugas Akhir ini.

Tugas Akhir ini penulis persembahkan untuk diri penulis sendiri, kedua orang tua, saudara, serta teman-teman yang selalu mendukung dan memberikan semangat kepada penulis sampai saat ini sehingga berkat dukungan dari berbagai pihak tersebut, penulis dapat menjadikannya motivasi untuk menyelesaikan Tugas Akhir ini. Penulis juga sangat berterima kasih kepada dosen-dosen Program Studi Informatika yang telah mencurahkan ilmu-ilmunya yang bermanfaat. Terkhusus juga penulis sampaikan terima kasih kepada Dosen Pembimbing Akademik penulis, Ibu Izzati Muhimmah, S.T., M.Sc., Ph.D. yang selalu mengingatkan penulis untuk segera menyelesaikan Tugas Akhir.

Terima kasih yang tak terhingga juga kepada Ibu Sri Mulyati, S.Kom.,M.Kom., selaku dosen penanggung jawab untuk angkatan 2014, Bapak Zainudin Zukhri, S.T, M.IT, selaku Dosen Pembimbing, dan Bapak Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D. yang telah membantu penulis dalam mengumpulkan informasi dan data-data terkait Tugas Akhir penulis.

## HALAMAN MOTO

*“Tiada kata terlambat sebelum menemui ajal”*

*“Motivasi terbesar tidak datang dari motivator handal, tapi harus dibangkitkan dari dalam diri sendiri”*

*“Lari dari tanggung jawab tidak akan menyelesaikan apapun”*



## KATA PENGANTAR

*Assalamu'alaikum Warahmatullahi Wabarakatuh.*

Alhamdulillah, puji dan syukur saya panjatkan kepada Allah Subhanahuwata'ala, berkat rahmat, karunia, dan hidayah-Nya saya dapat menyelesaikan Tugas Akhir yang berjudul “Sistem Penjadwalan Mata Kuliah Daring untuk Program Studi Informatika Universitas Islam Indonesia Menggunakan Metode Algoritma Genetika” sebagai syarat menyelesaikan masa pendidikan saya di jenjang sarjana Program Studi Informatika Universitas Islam Indonesia.

Dalam menyelesaikan Tugas Akhir ini, terdapat berbagai kendala yang harus saya lewati. Kendala yang sangat berarti adalah saya selalu menunda mengerjakannya selama kurang lebih 3(tiga) tahun. Hal ini dikarenakan saya terlalu menyibukkan diri dengan kegiatan-kegiatan yang tidak ada hubungannya dengan pengerjaan Tugas Akhir ini. Namun di akhir masa kuliah saya ini, dengan banyak bantuan moral, bimbingan, dan do'a dari berbagai pihak, Tugas Akhir ini dapat saya selesaikan.

Pada kesempatan ini, penulis ingin menyampaikan ucapan terima kasih kepada:

- a. Allah Subhanahuwata'ala yang telah memberikan saya nikmat yang tidak terhingga, terutama nikmat sehat dan Islam kepada penulis.
- b. Zahari Koto dan Rawismawati, selaku ayah dan mama penulis yang sangat penulis cintai dan banggakan. Berkat bimbingan dan pendampingan ayah dan mama yang sabar menghadapi penulis dalam menyelesaikan Tugas Akhir ini.
- c. Aulia Putrid an Tri Meilana, selaku saudara-saudara penulis yang telah memberikan bantuan moral dan semangat yang tiada henti-hentinya kepada penulis.
- d. Ibu Izzati Muhimmah, S.T., M.Sc., selaku Dosen Pembimbing Akademik penulis yang dengan sabar membimbing dan selalu mengingatkan penulis untuk segera menyelesaikan masa studi penulis.
- e. Bapak Zainudin Zuhri, S.T, M.IT, selaku Dosen Pembimbing Tugas Akhir penulis, yang telah bersedia meluangkan waktu dan pikiran dalam membimbing penulis sehingga penulis dapat menyelesaikan Tugas Akhir dengan baik.
- f. Dosen-dosen Program Studi Informatika UII yang telah berjasa dalam memberikan ilmu yang sangat bermanfaat selama masa perkuliahan.
- g. Elang Cergas Pembrani, seorang teman yang telah membantu penulis ketika dalam kesulitan selamu pengerjaan Tugas Akhir ini.



- h. Muhammad Khairi dan Jerina Novita Elpasari, sahabat terdekat penulis yang tanpa lelah selalu memberikan dukungan moral yang sangat besar kepada penulis selama 3(tiga) tahun belakangan.
- i. Jecky Fernando, teman seperjuangan yang berjuang juga dalam menyelesaikan Tugas Akhir yang selalu berbagi informasi kepada penulis terkait perkembangan Tugas Akhir, syarat-syarat dan jadwal sidang.
- j. Teman-teman penulis selama masa perkuliahan yang telah memberikan banyak ilmu dan pengalaman berharga kepada penulis.

Saya sebagai penulis, menyadari Tugas Akhir yang telah penulis kerjakan dengan maksimal ini masih mempunyai kekurangan sehingga penulis sangat membutuhkan kritik dan saran agar penelitian ini dapat dikembangkan lebih baik lagi bagi peneliti-peneliti yang di masa akan datang menjadikan Tugas Akhir penulis ini sebagai referensi penelitiannya. Semoga penelitian ini dapat bermanfaat untuk ke depannya.

*Wassalamu'alaikum Warahmatullahi Wabarakatuh.*

Yogyakarta, 10 Desember 2021



( Hari Kurnia )



## SARI

Pembuatan jadwal mata kuliah merupakan elemen penting untuk keberlangsungan proses pembelajaran yang akan dilaksanakan setiap semester pada sebuah institusi pendidikan. Penjadwalan mata kuliah yang efektif akan menciptakan proses belajar mengajar yang efisien, baik dari sisi mahasiswa maupun dosen. Mengingat, situasi pandemik virus COVID-19 yang sedang terjadi dalam kurun waktu hamper 2 tahun belakangan ini yang tidak memungkinkan untuk mengadakan proses belajar mengajar secara tatap muka, maka diterapkan perkuliahan secara daring.

Tidak beda halnya dengan penjadwalan mata kuliah tatap muka yang memiliki kompleksitas tinggi, penjadwalan mata kuliah daring juga dihadapkan dengan banyaknya faktor *constraint* dalam penentuan jadwal, misalnya banyak kelas, jumlah dosen yang mengajar, pengelompokan mata kuliah paket, hingga mata kuliah yang dibuka hanya di semester ganjil, genap, maupun tiap semester. Selain permasalahan penjadwalan, maka akan timbul juga permasalahan penjadwalan sesi kuliah pengganti. Hal ini dikarenakan dosen atas mata kuliah terkait tidak dapat hadir pada jadwal yang telah berlaku selama satu semester. kedua permasalahan di atas dapat diselesaikan menggunakan metode algoritma genetika.

Algoritma genetika merupakan metode optimasi dengan pendekatan komputasional yang terinspirasi dari teori genetika dan evolusi Mendell untuk menyelesaikan masalah pengoptimasian. Penggunaan istilah gen, kromosom, populasi, *crossover*, mutasi, dan regenerasi akan dijumpai pada tahapan pengoptimalan menggunakan algoritma genetika. Pada penelitian ini, implementasi algoritma genetika dalam penjadwalan mata kuliah daring adalah menemukan jadwal optimal yang dapat digunakan sebagai bahan KRS mahasiswa Program Studi S-1 Informatika Universitas Islam Indonesia.

Melalui beberapa kali percobaan, pada penelitian ini menggunakan jumlah populasi awal yang ideal digunakan adalah 100, titik *crossover* sebanyak 3 titik, probabilitas *crossover* ( $P_c$ ) sebesar 0,07 dan probabilitas mutasi ( $P_m$ ) sebesar 0,002 dan solusi optimal ditentukan berdasarkan nilai *fitness* terkecil. Melalui proses *generate* jadwal, sistem akan menghasilkan beberapa pilihan kombinasi jadwal melalui beberapa kromosom di dalam satu generasi terbaik yang memiliki nilai *fitness* yang sama sebagai calon solusi optimal. Kondisi ini memungkinkan *user* memilih salah satu dari beberapa hasil optimal yang dihasilkan.

Kata kunci: penjadwalan, optimasi, *fitness*, *crossover*, mutasi, Algoritma Genetika

## GLOSARIUM

Dalam penelitian ini, terdapat istilah-istilah yang memerlukan penjelasan. Berikut penjelasan dari istilah-istilah tersebut:

<i>Selection</i>	operasi untuk memilih beberapa kromosom sebagai calon induk yang akan menurunkan sifat-sifatnya kepada anaknya
<i>Fitness</i>	fungsi tujuan masalah yang hendak diselesaikan
<i>Crossover</i>	penyilangan dua buah kromosom untuk menghasilkan dua kromosom baru yang tetap memiliki sifat-sifat induknya
<i>Mutation</i>	operatot pendukung yang berperan mengubah struktur kromosom secara spontan yang menyebabkan terbentuknya suatu mutan
<i>Soft computing</i>	metode pengolahan suatu data yang memiliki aturan fleksibel atau dapat disesuaikan, tidak bersifat pasti
<i>Script</i>	salah satu bentuk bahasa pemrograman yang berperan sebagai penunjang atau pelengkap suatu program
<i>Dummy</i>	data yang dibuat menyerupai data aslinya, tetapi diisi dengan isian yang dibuat-buat

## DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN PENGESAHAN DOSEN PEMBIMBING.....	ii
HALAMAN PENGESAHAN DOSEN PENGUJI.....	iii
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR .....	iv
HALAMAN PERSEMBAHAN .....	v
HALAMAN MOTO .....	vi
KATA PENGANTAR.....	vii
SARI.....	ix
GLOSARIUM.....	x
DAFTAR ISI .....	xi
DAFTAR TABEL.....	xiii
DAFTAR GAMBAR .....	xiv
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah .....	3
1.3 Tujuan Penelitian .....	3
1.4 Manfaat Penelitian .....	3
1.5 Batasan Masalah .....	3
1.6 Sistematika Penulisan.....	4
BAB II LANDASAN TEORI .....	5
2.1 Penjadwalan.....	5
2.1.1 Penjadwalan Mata Kuliah Daring.....	5
2.1.2 Penjadwalan Kelas Pengganti Mata Kuliah .....	5
2.2 Algoritma Genetika.....	6
2.2.1 Mekanisme Algoritma Genetika .....	8
2.2.2 Struktur Umum Algoritma Genetika .....	9
2.2.3 Komponen Utama Algoritma Genetika .....	10
2.3 Penelitian Sejenis .....	16
BAB III METODOLOGI PENELITIAN.....	18
3.1 Studi Literatur .....	18
3.2 Sumber Data .....	18
3.3 Perancangan Algoritma Genetika .....	18

	xii
3.4 Diagram Blok Penjadwalan Mata Kuliah Daring .....	19
3.5 Teknik Pengkodean Kromosom .....	20
3.6 Inisialisasi Populasi .....	21
3.7 Evaluasi Nilai <i>Fitness</i> .....	22
3.8 Seleksi Induk.....	24
3.9 <i>Crossover</i> .....	25
3.10 Mutasi .....	25
3.11 Regenerasi dan Iterasi.....	26
3.12 Perancangan <i>Database</i> .....	26
<b>BAB IV PEMBAHASAN DAN HASIL .....</b>	<b>34</b>
4.1 Implementasi Program .....	34
4.1.1 Penjadwalan Mata Kuliah Daring.....	34
4.1.2 Penjadwalan Mata Kuliah Pengganti.....	46
4.1.3 Analisis Kelebihan dan Kekurangan Sistem .....	50
4.1.4 Evaluasi Sistem.....	51
<b>BAB V KESIMPULAN DAN SARAN.....</b>	<b>56</b>
5.1 Simpulan.....	56
5.2 Saran.....	56
<b>DAFTAR PUSTAKA .....</b>	<b>58</b>
<b>LAMPIRAN .....</b>	<b>59</b>



## DAFTAR TABEL

Tabel 3.1 Contoh <i>value decoding</i> sebuah kromosom.....	19
Tabel 3.2 Pasangan peraturan dengan konstanta <i>penalty</i> .....	22
Tabel 3.3 Struktur Tabel blok_jam.....	25
Tabel 3.4 Struktur Tabel dosen .....	26
Tabel 3.5 Struktur Tabel generate_jkd .....	26
Tabel 3.6 Struktur Tabel generate_jkd_dosen_mengajar_matkul.....	26
Tabel 3.7 Struktur Tabel generate_jkd_generation .....	27
Tabel 3.8 Struktur Tabel generate_jkp .....	27
Tabel 3.9 Struktur Tabel generate_jkd_generation .....	28
Tabel 3.10 Struktur Tabel generate_jkp_kelas.....	28
Tabel 3.11 Struktur Tabel generate_jkp_kelas_sesi.....	29
Tabel 3.12 Struktur Tabel generate_jkp_mahasiswa_mengambil_kelas.....	29
Tabel 3.13 Struktur Tabel kelas .....	29
Tabel 3.14 Struktur Tabel kluster_matkul .....	30
Tabel 3.15 Struktur Tabel mahasiswa .....	30
Tabel 3.16 Struktur Tabel mata_kuliah .....	31
Tabel 3.17 Struktur Tabel mata_kuliah_paket.....	31
Tabel 3.18 Struktur Tabel mata_kuliah_pilihan.....	32
Tabel 3.19 Struktur Tabel paket.....	32
Tabel 4.1 Ringkasan Percobaan Sistem.....	51



## DAFTAR GAMBAR

Gambar 2.1 Diagram Blok Algoritma Genetika.....	8
Gambar 2.2 Contoh <i>Binary Encoding</i> .....	10
Gambar 2.3 Contoh <i>Permutation Encoding</i> .....	10
Gambar 2.4 Contoh <i>Value Encoding</i> .....	10
Gambar 2.5 Contoh <i>Encoding Tree</i> .....	11
Gambar 2.6 Contoh <i>one-point crossover</i> .....	14
Gambar 2.7 Contoh <i>two-point crossover</i> .....	14
Gambar 3.1 Diagram Blok Algoritma Genetika.....	18
Gambar 3.2 Contoh <i>value decoding</i> sebuah kromosom.....	19
Gambar 4.1 Tangkapan Layar Halaman Login.....	34
Gambar 4.2 Tangkapan Layar Halaman Dashboard.....	34
Gambar 4.3 Tangkapan Layar Menu Master Data.....	35
Gambar 4.4 Tangkapan Layar Halaman Daftar Dosen.....	35
Gambar 4.5 Menu <i>Generate</i> .....	36
Gambar 4.6 Halaman Generate Mata Kuliah Daring.....	36
Gambar 4.7 Proses Penambahan Periode Perkuliahan.....	37
Gambar 4.8 Tangkapan Layar Halaman <i>Input</i> Dosen Pengajar Mata Kuliah.....	37
Gambar 4.9 Tangkapan Layar Proses <i>Input</i> Data Dosen Pengajar.....	38
Gambar 4.10 Tangkapan Layar Daftar Dosen Mengampu Mata Kuliah.....	39
Gambar 4.11 Tangkapan Layar <i>Input</i> Argumen Algoritma Genetika.....	39
Gambar 4.12 Tangkapan Layar Tambah Generasi.....	40
Gambar 4.13 Proses Pembentukan Generasi Baru.....	40
Gambar 4.14 Generasi Baru.....	41
Gambar 4.15 Daftar Pelanggaran.....	41
Gambar 4.16 Tangkapan Layar Tabel Jadwal Mata Kuliah Daring.....	42
Gambar 4.17 Argumen Hasil Optimal.....	43
Gambar 4.18 Generasi dengan Nilai <i>Fitness</i> Buruk.....	44
Gambar 4.19 Jadwal Tetap Mata Kuliah.....	46
Gambar 4.20 Tangkapan Layar <i>Input</i> Argumen.....	47
Gambar 4.21 <i>Input</i> Kelas Mata Kuliah.....	47
Gambar 4.22 <i>Input</i> Jumlah Generasi.....	48
Gambar 4.23 Proses <i>Generate</i> Jadwal Pengganti.....	48

Gambar 4.24 Tangkapan Layar Jadwal Pengganti Mata Kuliah ..... 49



# BAB I PENDAHULUAN

## 1.1 Latar Belakang

Penjadwalan mata kuliah merupakan sebuah proses awal yang harus dilakukan oleh setiap perguruan tinggi, termasuk sebuah jurusan dalam sebuah perguruan tinggi tersebut. Penjadwalan mata kuliah memiliki peran dalam menghasilkan proses belajar mengajar yang efektif, baik dari sisi mahasiswa maupun dosen pengajar. Dari sisi mahasiswa, adanya kemungkinan bahwa mahasiswa tidak dapat mengambil mata kuliah yang seharusnya dapat diambil dikarenakan penjadwalan yang tidak efektif. Dari sisi dosen, adanya kemungkinan bahwa dosen dapat mengajar lebih dari satu mata kuliah pada waktu yang bersamaan. Selain itu, penjadwalan mata kuliah yang kaitannya dengan syarat-syarat penentuan sesi kelas juga menjadi perhatian penting. Seperti, mata kuliah yang berada pada kluster yang sama tidak dijadwalkan pada waktu yang bersamaan, mata kuliah yang dibuka tiap semester tidak dijadwalkan pada waktu yang bersamaan dengan mata kuliah paket semester.

Selain proses penjadwalan mata kuliah, fitur yang juga harus diperhatikan ketika jadwal mata kuliah sudah ditetapkan selama satu semester adalah sistem penjadwalan ulang untuk sesi kuliah pengganti. Fitur ini mengakomodir dinamisasi jadwal perkuliahan yang tidak bisa dilaksanakan karena dosen terkait berhalangan hadir pada jadwal yang sudah ada. Penjadwalan sesi kuliah pengganti disesuaikan dengan jadwal kosong dari dosen terkait dan juga memaksimalkan kehadiran mahasiswa pada jadwal baru yang akan dibuat. Memaksimalkan kehadiran mahasiswa yang dimaksud adalah karena adanya kemungkinan beberapa mahasiswa yang tidak dapat menghadiri sesi kuliah pengganti. Hal ini dikarenakan adanya kemungkinan jadwal kuliah pengganti diadakan pada waktu yang bersamaan dengan jadwal kuliah tetap dari beberapa mahasiswa.

Selama ini penjadwalan mata kuliah yang dilakukan oleh Jurusan Informatika Universitas Islam Indonesia berupa penjadwalan mata kuliah tatap muka antara dosen dan mahasiswa secara langsung di ruangan kelas yang tersedia. Akan tetapi, mengingat kondisi global yang belakangan ini sedang menghadapi sebuah pandemi virus Covid-19 terhitung sejak tahun 2020, proses belajar mengajar pun dilaksanakan secara *online* (daring). Oleh karena itu, proses penjadwalan mata kuliah juga harus diperbaharui menyesuaikan kondisi terkini. Untuk saat ini penjadwalan kuliah daring belum memiliki sebuah sistem yang terprogram secara

otomatis oleh Jurusan Informatika Universitas Islam Indonesia, termasuk penjadwalan sesi kuliah pengganti. Perbedaan penjadwalan mata kuliah daring dengan penjadwalan mata kuliah tatap muka terletak pada penggunaan ruangan. Penjadwalan mata kuliah daring tidak membutuhkan parameter kelas sebagai optimasi penjadwalan.

Pada penelitian ini, sistem penjadwalan dibuat dengan melibatkan parameter-parameter yang terdiri dari mata kuliah, waktu, dan dosen. Salah satu metode yang sering digunakan untuk kasus optimasi seperti kasus di atas adalah metode algoritma genetika. Algoritma genetika adalah suatu teknik pencarian (*searching technique*) dan teknik optimasi yang cara kerjanya meniru proses evolusi dari perubahan struktur genetika pada makhluk hidup (Yandra Arkeman, 2012). Dalam seleksi alam, individu-individu bersaing untuk mempertahankan hidup dan melakukan reproduksi. Individu-individu yang lebih "*fit*" akan memiliki peluang untuk terus bertahan hidup (*survive*) dan melakukan reproduksi (menghasilkan keturunan), begitu juga sebaliknya. Dengan prinsip seperti ini, penerapan algoritma genetika dapat menghasilkan optimasi dalam penjadwalan mata kuliah daring maupun penjadwalan sesi mata kuliah pengganti.

Syahputra et.al. (2018) dalam penelitian yang berjudul "Genetic Algorithm to Solve the Problems of Lectures and Practicums Scheduling" mengemukakan bahwa proses penjadwalan mata kuliah dan praktikum dilakukan dengan menggunakan Algoritma Genetika yang terdiri dari proses inialisasi populasi, seleksi, *crossover*, dan mutasi. Kromosom yang digunakan pada penelitian ini adalah mata kuliah, dosen, *credit value*, tingkat kelas, ruang kelas, hari, dan interval waktu. Proses penyeleksian menggunakan metode *Roulette Wheel Selection* untuk setiap individu. Pada proses mutasi menggunakan metode *Exchange Mutation*. Hasil dari proses penjadwalan menggunakan Algoritma Genetika adalah jadwal yang paling optimal yang sesuai dengan waktu yang tersedia, ruang kelas, mata kuliah dan jadwal dosen.

Ahmad et.al. (2018) dalam penelitian yang berjudul "A Heuristics Approach for Classroom Scheduling Using Genetic Algorithm Technique" mengemukakan bahwa proses penjadwalan kelas dilakukan dengan menggunakan Algoritma Genetika yang terdiri dari proses inialisasi populasi, seleksi, *crossover*, dan mutasi. Kromosom yang digunakan pada penelitian ini adalah dosen, waktu rapat, mata kuliah, ruangan dan departemen.

Berdasarkan latar belakang di atas, maka penulis mencoba untuk menggunakan algoritma genetika dalam optimasi penjadwalan mata kuliah daring dan penjadwalan sesi mata kuliah pengganti dengan membuat penelitian dengan judul "Sistem Penjadwalan Mata Kuliah Daring

untuk Program Studi Informatika Universitas Islam Indonesia Menggunakan Algoritma Genetika”.

## **1.2 Rumusan Masalah**

Adapun masalah yang akan diselesaikan berdasarkan latar belakang masalah yang telah dijelaskan, sebagai berikut:

- a. Bagaimana implementasi Algoritma Genetika dalam optimasi penjadwalan mata kuliah daring dan penjadwalan sesi kuliah pengganti?
- b. Bagaimana hasil optimasi yang diperoleh dengan menggunakan Algoritma Genetika dalam optimasi penjadwalan mata kuliah daring dan penjadwalan sesi kuliah pengganti?

## **1.3 Tujuan Penelitian**

Tujuan dari penelitian ini adalah untuk mengoptimalkan penjadwalan mata kuliah daring dan penjadwalan sesi kuliah pengganti yang diterapkan pada Jurusan Informatika Universitas Islam Indonesia.

## **1.4 Manfaat Penelitian**

Manfaat yang diharapkan dari penelitian ini antara lain:

- a. Dapat membantu Jurusan Informatika UII untuk memperoleh solusi optimal dalam proses penjadwalan mata kuliah daring dan penjadwalan sesi kuliah pengganti.
- b. Dapat menjadi tolak ukur dan referensi bagi penelitian selanjutnya.

## **1.5 Batasan Masalah**

Beberapa batasan masalah dalam penelitian ini, adalah sebagai berikut:

- a. Penjadwalan dilakukan untuk tingkat Strata-1 (S1) pada Program Studi Informatika Universitas Islam Indonesia.
- b. Penjadwalan dilakukan tanpa mempertimbangkan ketersediaan waktu dosen pengajar.
- c. Seluruh sesi kuliah di hari selasa ditiadakan sebagai jadwal perkuliahan.
- d. Pemodelan optimasi penjadwalan mata kuliah yang dilakukan hanya pada Mata Kuliah Wajib Keprodian(MKWK) Jurusan Informatika UII, serta tidak mengoptimasi jadwal praktikum.



## **1.6 Sistematika Penulisan**

Sistematika penulisan memuat metode penulisan yang digunakan dalam penelitian. Berikut merupakan sistematika penulisan penelitian yang terbagi menjadi lima bab.

### **BAB I PENDAHULUAN**

Berisi latar belakang masalah, rumusan masalah, tujuan penelitian, manfaat penelitian, Batasan masalah, dan sistematika penulisan.

### **BAB II LANDASAN TEORI**

Berisi mengenai teori-teori yang digunakan dalam penelitian yang diambil dari berbagai sumber terutama penelitian-penelitian yang pernah dilakukan. Diantaranya ialah teori berdasarkan penelitian sebelumnya, teori mengenai Travelling Salesman Problem, dan teori mengenai Algoritma yang digunakan pada penelitian yaitu Algoritma Genetika dan Tabu Search.

### **BAB III METODE PENELITIAN**

Berisi tentang uraian langkah-langkah yang dilakukan untuk menyelesaikan penelitian.

### **BAB IV HASIL DAN PEMBAHASAN**

Berisi tentang pemaparan hasil implementasi dan hasil pengujian berdasarkan skenario pengujian pada BAB III.

### **BAB V KESIMPULAN DAN SARAN**

Berisi tentang kesimpulan dari hasil penelitian dan saran untuk penelitian selanjutnya.

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Penjadwalan**

Penjadwalan adalah proses penyusunan daftar pekerjaan yang akan dilakukan untuk mencapai atau mewujudkan suatu tujuan tertentu yang juga memuat tabel waktu pelaksanaannya. Pengertian jadwal menurut Kamus Besar Bahasa Indonesia adalah pembagian waktu berdasarkan rencana pengeturan kerja, daftar atau tabel kegiatan atau rencana kegiatan dengan pembagian waktu pelaksanaan terperinci. Sedangkan pengertian penadwalan adalah proses, cara pembuatan penjadwalan atau memasukkan dalam jadwal.

Jadwal dari suatu kegiatan merupakan hal yang penting dilakukan agar kegiatan tersebut berjalan dengan baik. Penyusunan jadwal kegiatan berkaitan juga dengan berbagai syarat yang harus terpenuhi sehingga memerlukan beberapa pertimbangan agar syarat-syarat yang ada dapat terpenuhi sebagai mana mestinya.

##### **2.1.1 Penjadwalan Mata Kuliah Daring**

Mata kuliah merupakan salah satu komponen penting dalam keberlangsungan proses belajar mengajar. Proses ini bergantung pada mata kuliah yang diadakan pada rentang waktu tertentu, biasanya berlaku untuk sekali semester. Proses belajar mengajar dilakukan pada waktu dan lokasi yang sudah disepakati bersama oleh pihak penyelenggara perkuliahan.

Penjadwalan mata kuliah daring(*online*) merupakan penyusunan jadwal perkuliahan yang dilakukan secara *online*. Kondisi ini tidak mengikutsertakan lokasi proses belajar mengajar, seperti ruangan yang biasanya digunakan untuk melakukan proses perkuliahan. Parameter ruangan dapat diganti dengan mengakses ruangan virtual yang disediakan oleh perusahaan-perusahaan teknologi yang memiliki produk berupa aplikasi *virtual room meeting*. Dengan adanya kondisi seperti ini, proses perkuliahan dapat dilakukan tanpa mempertimbangkan ketersediaan ruangan dan kapasitas setiap ruangan.

##### **2.1.2 Penjadwalan Kelas Pengganti Mata Kuliah**

Mata kuliah pengganti merupakan jadwal perkuliahan yang diadakan sebagai pengganti jadwal mata kuliah yang sudah ditetapkan selama semester tertentu. Jadwal pengganti adalah jawaban dari jadwal perkuliahan yang dinamis, yang disebabkan adanya jadwal kuliah yang

kosong akibat dosen berhalangan hadir pada jadwal awal yang sudah ditetapkan di awal semester.

Penjadwalan pengganti waktu mata kuliah dilakukan dengan pertimbangan waktu yang tersedia pada dosen terkait dan persentase kehadiran mahasiswa pada waktu yang akan ditetapkan sebagai jadwal pengganti dari kelas mata kuliah yang diampu oleh dosen terkait. Selain itu, ketentuan melakukan jadwal pengganti adalah dengan melakukan penjadwalan pada minggu yang sama dengan kelas mata kuliah yang akan dirubah jadwalnya.

## 2.2 Algoritma Genetika

Algoritma genetika pertama kali dikembangkan pada tahun 1970-an oleh John Holland, seorang professor di University of Michigan, Amerika. Tujuan yang ingin dicapai Holland pada saat itu adalah mengabstraksi proses-proses evolusi yang terjadi di alam dan mendesain suatu *software* yang prinsip kerjanya meniru proses-proses evolusi. Hasilnya, algoritma genetika ternyata dapat menyelesaikan masalah-masalah yang tidak dapat diselesaikan dengan menggunakan perhitungan matematika biasa. Algoritma genetika juga secara khusus dapat menyelesaikan masalah-masalah optimasi dengan baik. Suatu kejadian yang sangat penting dari penemuan ini adalah bahwa algoritma genetika hanya melibatkan operasi-operasi matematika sederhana.

Algoritma genetika adalah suatu teknik pencarian (*searching technique*) dan teknik optimasi yang cara kerjanya meniru proses evolusi dan perubahan struktur genetik pada makhluk hidup. Algoritma genetika juga disebut sebagai *soft computing*. Prinsip utama dari cara kerja algoritma genetika ini diilhami oleh proses seleksi alam dan prinsip-prinsip ilmu genetika. Dalam seleksi alam, individu-individu bersaing untuk mempertahankan hidup dan melakukan reproduksi individu-individu yang lebih "*fit*" akan memiliki peluang untuk terus bertahan hidup (*survive*) dan menghasilkan keturunan. Sebaliknya, individu-individu yang kurang "*fit*" akan mati dan punah (prinsip ini dinamakan "*survival of the fittest*"). Selanjutnya, dalam seleksi alam ini, beberapa individu baru yang lebih "*fit*" dari kedua orang tuanya akan dilahirkan melalui proses yang disebut penyilangan (*crossover*) dan mutasi. Kedua proses ini terjadi pada kromosom-kromosom individu yang melakukan reproduksi. Proses seleksi dan reproduksi (penyilangan dan mutasi) ini berlangsung berulang-kali, sampai individu yang paling "*fit*" dihasilkan. Individu yang paling *fit* inilah yang merupakan solusi dari masalah yang dihadapi.

Dalam proses implementasi algoritma genetika, langkah awal yang dilakukan adalah membangkitkan populasi awal secara acak. Setelah populasi dibangkitkan, setiap kromosom yang terdapat di dalam populasi tersebut diberikan nilai *fitness* sesuai aturan yang digunakan. Jika ditemukan solusi, maka proses algoritma akan langsung berhenti dengan ketentuan kondisi berhenti yang didefinisikan di awal. Jika kondisi belum memenuhi solusi yang dibutuhkan, maka kromosom yang terdapat pada populasi tersebut dilakukan proses penyeleksian (*selection*) untuk menghasilkan kromosom-kromosom yang akan bertindak sebagai calon kromosom induk (*parent*) melalui metode seleksi yang ada. Setelah didapatkan kromosom calon induk, dipilih beberapa induk yang akan dilakukan proses *crossover* untuk menghasilkan kromosom anak (*offspring*) dan selanjutnya mutasi. Tahapan selanjutnya adalah, regenerasi, yang merupakan representasi dari seleksi alam. Kromosom-kromosom yang memiliki *fitness* yang lebih baik yang akan bertahan pada generasi selanjutnya. Proses ini akan berlangsung melalui iterasi perulangan sampai pada kondisi tertentu sehingga solusi terbaik yang mungkin didapat ditemukan.

Pada algoritma genetika, terdapat beberapa istilah penting yang harus diketahui sebelum melakukan proses implementasi pada kasus yang akan diselesaikan menggunakan algoritma ini. Berikut istilah-istilah tersebut:

a. Gen

Gen adalah nilai yang menyatakan satuan dasar yang membentuk suatu arti tertentu dalam satu kesatuan gen yang dinamakan kromosom.

b. Kromosom atau Individu

Kromosom merupakan gabungan dari beberapa gen yang membentuk suatu nilai tertentu dan menyatakan solusi yang mungkin terjadi dari suatu permasalahan.

c. Populasi

Populasi merupakan sekumpulan individu atau kromosom yang akan diproses bersama dalam satu satuan siklus evolusi.

d. Fungsi *Fitness*

Fungsi *fitness* merupakan suatu fungsi yang akan merepresentasikan seberapa baik nilai dari suatu kromosom dari suatu populasi yang didapatkan.

e. Seleksi/*Selection*

Seleksi merupakan proses penyeleksian kromosom-kromosom dalam sebuah populasi yang digunakan sebagai calon induk yang baik.

f. Penyilangan/*Crossover*

Penyilangan merupakan proses pertukaran atau kawin silang gen-gen dari dua induk tertentu.

g. Mutasi/*Mutation*

Mutasi merupakan proses pergantian salah satu gen yang terpilih dengan nilai tertentu.

h. Generasi

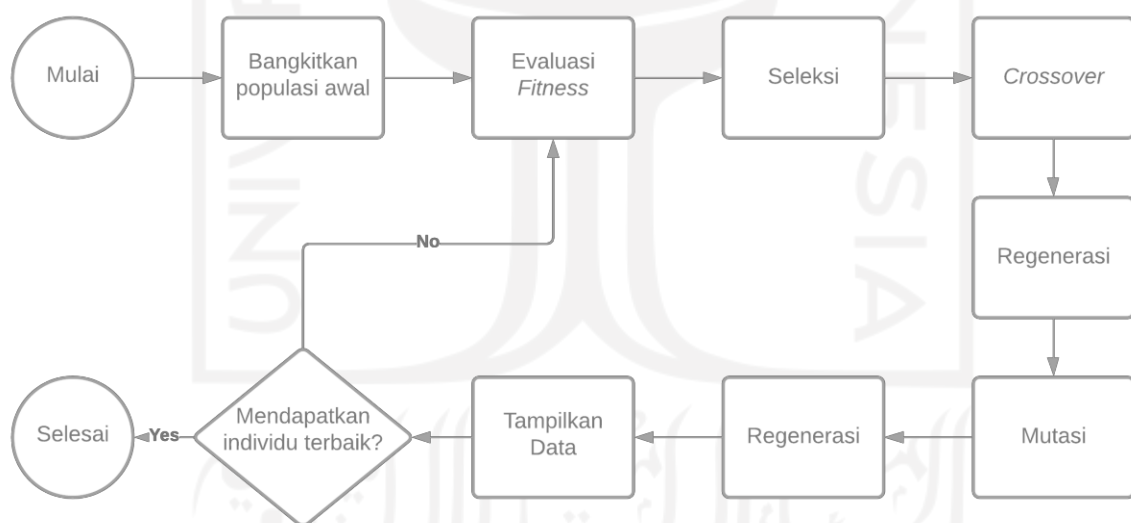
Generasi merupakan urutan iterasi tempat beberapa kromosom bergabung

i. *Offspring*

*Offspring* merupakan kromosom baru yang dihasilkan, biasa disebut kromosom anak.

### 2.2.1 Mekanisme Algoritma Genetika

Algoritma genetika dapat dipahami melalui bagan alur atau *flowchart* yang menampilkan langkah-langkah dan keputusan dalam pengimplementasiannya. Bagan alur algoritma genetika dapat dilihat pada Gambar 2.1 berikut:



Gambar 2.1. Diagram Blok Algoritma Genetika

Sebelum sebuah kasus yang menggunakan algoritma genetika dijalankan, maka perlu didefinisikan terlebih dahulu sebuah fungsi *fitness* sebagai masalah yang ingin dioptimalkan. Jika nilai *fitness* semakin baik, maka sistem yang dihasilkan semakin baik. Pendefinisian fungsi *fitness* untuk mencapai hasil optimal dapat ditentukan sesuai keinginan *programmer*. Jika hasil optimal diinginkan dengan nilai *fitness* yang semakin besar, maka tujuan akhirnya adalah



mendapatkan nilai *fitness* terbesar sebagai solusi permasalahan optimasi. Begitu juga sebaliknya, jika hasil optimal didapatkan melalui nilai *fitness* yang semakin kecil, maka akan dibutuhkan nilai *fitness* terkecil sebagai solusi permasalahan optimasi.

Kemudian, dapat dilihat pada Gambar 2.1 di atas, proses algoritma genetika dapat dikerjakan berdasarkan urutan langkah yang ada. Iterasi yang dilakukan untuk menghasilkan generasi selanjutnya bergantung pada seberapa baik nilai *fitness* yang didapatkan. Jika dalam sekali operasi algoritma genetika belum mendapatkan individu terbaik yang akan menjadi solusi optimasi, maka proses iterasi generasi akan terus berlangsung.

Algoritma Genetika merupakan pilihan yang tepat untuk digunakan dalam menyelesaikan masalah optimasi yang kompleks dan sukar diselesaikan dengan menggunakan metode konvensional. Sebagaimana halnya proses evolusi di alam, suatu Algoritma Genetika yang sederhana umumnya terdiri dari tiga operasi yaitu: operasi reproduksi, operasi *crossover* (persilangan), dan operasi mutasi. Sebelum melakukan proses algoritma genetika yang menggunakan operator.

### 2.2.2 Struktur Umum Algoritma Genetika

Struktur umum dari suatu algoritma genetika dapat didefinisikan dengan langkah sebagai berikut (Suhartono, 2015):

a. Membangkitkan populasi awal

Populasi awal dibangkitkan secara acak sehingga didapatkan solusi awal berupa sekumpulan individu atau kromosom yang tergabung ke dalam populasi tersebut.

b. Membentuk generasi baru

Membentuk generasi baru melalui proses seleksi dan dua operator genetika yaitu operator operator *crossover* dan mutasi. Tahapan tersebut dilakukan secara berulang-ulang sehingga diperoleh generasi baru yang memiliki kromosom yang dapat merepresentasikan dari solusi baru. Generasi baru ini dikenal dengan istilah kromosom anak (*offspring*).

c. Evaluasi solusi

Pada setiap generasi, masing-masing kromosom akan diukur dengan fungsi *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 hingga memenuhi kriteria berhenti. Beberapa kriteria berhenti yang sering digunakan antara lain:

1. Berhenti pada generasi tertentu.

2. Berhenti setelah mendapatkan nilai *fitness* terbaik pada beberapa generasi berturut-turut yang nilai yang sama.
3. Berhenti pada generasi tertentu pada kondisi nilai *fitness* terbaik dari generasi baru tidak mengalami perubahan dari beberapa generasi sebelumnya.

### 2.2.3 Komponen Utama Algoritma Genetika

Terdapat 6 komponen utama yang digunakan dalam algoritma genetika (Suhartono, 2015), yaitu sebagai berikut:

#### a. Teknik Pengodean (*Encoding*)

Suatu teknik yang digunakan dalam mengkodekan gen dari sebuah individu. Teknik ini berfungsi untuk mendefinisikan populasi awal sebagai calon solusi dari suatu masalah. Beberapa macam pengodean dalam algoritma genetika, yaitu sebagai berikut (Saptono, 2007):

1. *Binary encoding*, pengodean yang paling sering sederhana dan digunakan dalam algoritma genetika, pengodean ini terdiri dari bit 0 dan 1 pada setiap gen dalam suatu kromosom.

0	1	1	0	0	1
---	---	---	---	---	---

Gambar 2.2 Contoh *Binary Encoding*

2. *Permutation encoding*, pengodean kromosom dengan suatu angka yang merepresentasikan urutan. Kode yang terdapat pada setiap gen bersifat unik.

1	5	2	6	4	3
---	---	---	---	---	---

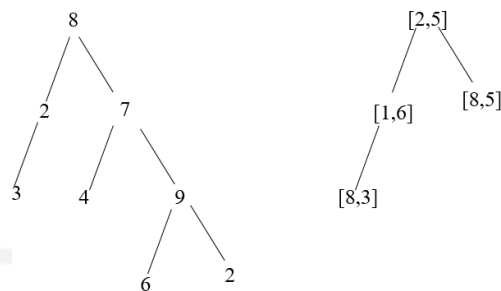
Gambar 2.3 Contoh *Permutation Encoding*

3. *Value encoding*, setiap kromosom adalah string dari suatu nilai yang merepresentasikan suatu masalah, seperti karakter, desimal, ataupun bilangan bulat.

Kromosom A	1.24, 2.46, 7.41, 9.82
Kromosom B	ABC, CBA, CBC, BCA
Kromosom C	<i>Forward, Backward, Right, Left</i>

Gambar 2.4 Contoh *Value Encoding*

4. *Encoding tree*, biasa digunakan dalam menyusun program *genetic programming*. Dalam *encoding tree* ini, setiap kromosom dinyatakan sebagai sebuah pohon.



Gambar 2.5 Contoh *Encoding Tree*

b. Prosedur Inisialisasi

Membangkitkan sejumlah individu secara acak yang tergabung pada sebuah populasi awal. Ukuran dari populasi dipengaruhi oleh suatu masalah yang akan diselesaikan dan jenis operator genetik yang akan diterapkan pada pemecahan masalah tersebut. Setelah ukuran populasi ditetapkan, dilakukan proses inisialisasi pada kromosom-kromosom yang terdapat pada populasi tersebut. Penginisialisasian ini dilakukan secara acak dengan tetap memperhatikan daerah solusi yang ingin dihasilkan dan kendala permasalahan diberlakukan.

c. Fungsi Evaluasi

Setiap individu dilakukan evaluasi berdasarkan sebuah fungsi matematika tertentu sehingga dapat dijadikan ukuran tingkat kebugaran (*fitness*) individu tersebut. Fungsi evaluasi juga dapat disebut fungsi *fitness*. Fungsi *fitness* dapat sama atau merupakan hasil modifikasi terhadap fungsi tujuan dari masalah yang akan diselesaikan. Individu yang memiliki nilai *fitness* yang baik akan dipertahankan, sedangkan individu dengan nilai *fitness* buruk akan digantikan oleh individu lain. Fungsi *fitness* tidak memiliki rumus baku, melainkan dapat disesuaikan dengan permasalahan tertentu dari representasi yang digunakan. Berikut merupakan salah satu pendefinisian fungsi *fitness* yang digunakan (Denny, 2020).

$$\text{fitness} = \frac{1}{1 + \sum_{i=1}^n x_i} \quad (2.1)$$

$x_i$  adalah jumlah *penalty* yang ada pada setiap kromosom

$$x_i = x_1 + x_2 + x_3 + \dots + x_n$$

$$x_1 = C + D + JK + KK + P + R + S$$

$$x_2 = C + D + JK + KK + P + R + S$$

Keterangan:

$C$  = Mata Kuliah ( $c_1, c_2, c_3, \dots, c_n$ )

$D$  = Dosen ( $d_1, d_2, d_3, \dots, d_n$ )

$JK$  = Jenis Kelas ( $j_1, j_2, j_3, \dots, j_n$ )

$KK$  = Kapasitas Kelas ( $k_1, k_2, k_3, \dots, k_n$ )

$P$  = Program Studi ( $p_1, p_2, p_3, \dots, p_n$ )

$R$  = Ruang ( $r_1, r_2, r_3, \dots, r_n$ )

$S$  = Semester ( $s_1, s_2, s_3, \dots, s_n$ )

#### d. Seleksi

Proses seleksi memiliki tujuan untuk memilih beberapa individu sebagai calon induk yang akan dilakukan *crossover* dan mutasi. Oleh sebab itu, diharapkan calon induk tersebut dapat menghasilkan keturunan yang baik. Sebelum memilih individu-individu tersebut, masing-masing individu harus diinisialisasikan nilai kebugarannya terlebih dahulu melalui fungsi *fitness* yang sudah ditetapkan sebelumnya. Setelah melakukan proses seleksi menggunakan metode yang ada, masing-masing calon induk akan menerima probabilitas reproduksi yang tergantung pada nilai objektif dirinya sendiri terhadap nilai objektif semua individu yang terdapat di dalam wadah seleksi tersebut. Terdapat beberapa teknik seleksi yang dapat digunakan, antara lain:

1. Teknik seleksi cakram rolet (*roulette wheel selection*)

Teknik seleksi ini diilustrasikan sebagai teknik pemutaran cakram rolet. Setiap kromosom yang terdapat di dalam populasi menempati suatu slot pada cakram rolet. Besarnya ukuran slot merupakan nilai yang sama dengan rasio antara nilai *fitness* suatu kromosom dengan total *fitness* keseluruhan kromosom pada populasi tersebut. Untuk menghasilkan sejumlah calon kromosom induk, suatu rolet diputar sebanyak ukuran populasi yang ada.

2. Teknik seleksi turnamen (*tournament selection*)

Teknik seleksi ini diilustrasikan sebagai proses kompetisi dari beberapa kromosom dalam sebuah kelompok kecil sampai tersisa satu individu pemenang. Individu yang memenangkan turnamen akan bertahan dan dijadikan sebagai calon

kromosom induk yang selanjutnya akan dilakukan proses penyilangan (*crossover*). Dalam skala kelompok terkecil, teknik ini memilih 2 kromosom secara acak kemudian melakukan evaluasi nilai *fitness* keduanya. Kromosom dengan nilai *fitness* paling baik yang akan memenangkan kompetisi.

#### e. Operator Algoritma Genetika

Penggunaan algoritma genetika bertujuan agar sekumpulan kromosom atau calon solusi yang dibangkitkan secara acak pada penginisialisasian populasi awal dapat berkembang biak dengan sendirinya melalui beberapa generasi hingga konvergen menghasilkan suatu nilai *fitness* terbaik yang akan merepresentasikan nilai optimal pada sebuah permasalahan. Kromosom yang terbentuk pada generasi baru disebut sebagai kromosom anak (*offspring*). Sebuah kromosom anak dapat terbentuk melalui dua proses utama, yaitu dengan menggabungkan elemen-elemen antara dua kromosom induk menggunakan operator penyilangan, atau dengan memodifikasi sebuah kromosom induk menggunakan operator mutasi.

Dalam suatu populasi, kedua proses di atas dapat berlangsung secara berurutan (*sequential*) maupun paralel. Berlangsung secara berurutan artinya proses penyilangan antar kedua kromosom induk terjadi terlebih dahulu, lalu dilanjutkan proses mutasi pada kedua kromosom anak yang baru terbentuk. Proses ini disebut dengan *mutaion embedded within crossover*. Sedangkan secara paralel artinya proses penyilangan dan mutasi terjadi secara terpisah pada kromosom-kromosom induk saja, tidak pada kromosom anak yang baru dihasilkan. Kedua operator genetika di atas dapat dijelaskan lebih lanjut melalui pembahasan di bawah ini.

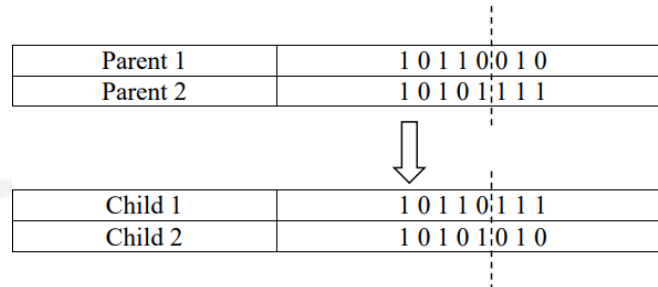
##### 1. Penyilangan (*crossover*)

Penyilangan (*crossover*) merupakan operator utama dalam algoritma genetika. Operator ini bekerja pada sepasang kromosom induk untuk menghasilkan dua buah kromosom anak dengan cara menukarkan beberapa elemen (*gen*) yang dimiliki masing-masing kromosom induk. Proses penyilangan ini bertujuan menciptakan keberagaman baru pada populasi dan generasi selanjutnya. Pada operator *crossover*, terdapat beberapa metode yang sering digunakan, antara lain:

###### (a). *One-point crossover*

Penyilangan dilakukan pada dua buah kromosom dengan memilih tepat satu titik secara acak yang akan dijadikan pemisah panjang kromosom menjadi 2 bagian.

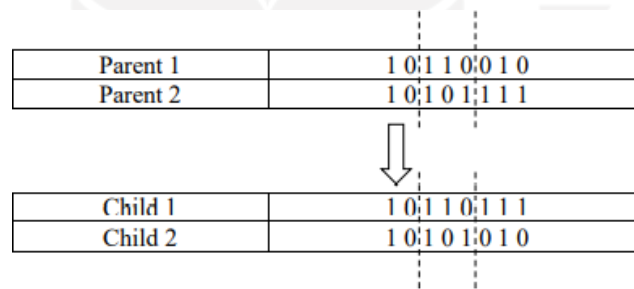
Selanjutnya elemen-elemen pada salah satu sisinya dilakukan penukan dengan elemen-elemen lain pada kromosom lain yang juga sudah dilakukan pemisahan yang sama. Gambar 2.6 adalah ilustrasi dari *one-point crossover*.



Gambar 2.6 Contoh *one-point crossover*

(b). *Two-point crossover*

Teknik penyilangan yang hampir sama dengan metode *one-point crossover*, akan tetapi titik *crossover* yang dipilih adalah 2 buah. Gambar 2.7 merupakan ilustrasi dari *two-point crossover*.



Gambar 2.7 Contoh *two-point crossover*

(c). *Partially-mapped crossover*

Teknik ini dipandang sebagai perbaikan dari teknik *two-point crossover*. pernaikan yang terdapat pada teknik ini disebut dengan metode *repairing procedure* untuk mencegah munculnya kromosom ilegal.

(d). *Position-based crossover*

Teknik penyilangan dilakukan dengan membangkitkan sebuah nilai yang akan merepresentasikan banyaknya elemen yang akan dilakukan proses penyilangan. Pemilihan elemen-elemen ini akan dilakukan secara acak sehingga memenuhi banyak titik yang ingin dilakukan penyilangan.

## 2. Mutasi (*mutation*)

Operator mutasi merupakan operator sekunder dalam algoritma genetika. Mutasi bertujuan untuk mengubah salah satu atau beberapa gen dari suatu kromosom. Proses ini bertujuan untuk mengganti gen yang hilang akibat proses seleksi yang memungkinkan munculnya kembali gen yang hilang sebelumnya pada inisialisasi populasi. Akan tetapi, proses mutasi yang terlalu banyak akan mengakibatkan individu yang sudah memiliki nilai *fitness* baik akan kembali memburuk akibat adanya penggantian gen yang sudah ada. Hal ini dikarenakan gen yang akan dimutasi dipilih secara acak melalui argumen kontrol yang ditetapkan. Berikut beberapa operator mutasi yang dapat digunakan, antara lain mutasi invers, mutasi insersi, *displacement mutation* dan *reciprocal exchange mutation*.

## f. Argumen Kontrol

Argumen kontrol dalam algoritma genetika dibutuhkan untuk mengendalikan operator-operator seleksi maupun mengontrol ukuran populasi yang ingin dibangkitkan. Argumen-argumen tersebut adalah:

### 1. Probabilitas *Crossover* ( $P_c$ )

Probabilitas *crossover* akan mengendalikan operator penyilangan dalam setiap generasi. Pemilihan kromosom induk yang sudah melalui proses seleksi akan dikenakan probabilitas *crossover* untuk memilih jumlah kromosom yang akan mengalami *crossover*. Penetapan nilai probabilitas *crossover* yang tinggi akan menciptakan keanekaragaman kromosom anak atau individu-individu baru yang tinggi ke dalam populasi, begitu juga sebaliknya. Nilai probabilitas *crossover* yang biasanya digunakan cukup tinggi, dapat bernilai 0,6 sampai 1. Tingkat penyilangan yang semakin tinggi akan menyebabkan semakin besarnya algoritma genetika mengeksplorasi ruang pencarian sehingga akan berdampak pada semakin cepat ditemukannya solusi optimal dari suatu permasalahan.

### 2. Probabilitas Mutasi ( $P_m$ )

Tingkat mutasi atau probabilitas mutasi adalah rasio antara jumlah gen yang diharapkan mengalami mutasi pada setiap generasi dengan jumlah total gen dalam sebuah populasi. Nilai probabilitas mutasi yang digunakan biasanya cukup rendah, dapat berkisar 0,001 sampai 0,2. Nilai yang tidak baku, akan tetapi dapat disesuaikan



dengan jumlah gen yang terdapat pada satu generasi. Hal ini mengacu pada seleksi alam murni, mutasi jarang sekali muncul sehingga operator mutasi pada algoritma genetika tidak selalu terjadi dalam pembentukan generasi baru.

3. *Popsize* (ukuran populasi)

Ukuran populasi menunjukkan jumlah kromosom yang terdapat dalam satu generasi, yang mana akan selalu sama pada setiap generasi. Ukuran populasi juga memiliki peran penting dalam mencapai tujuan yang diinginkan. Semakin banyak populasi yang diinisialisasikan, akan menghasilkan tingkat keanekaragaman kromosom yang semakin besar juga.

### 2.3 Penelitian Sejenis

Penggunaan algoritma genetika dalam menyelesaikan permasalahan optimasi sudah dilakukan pada banyak penelitian sehingga penelitian-penelitian tersebut dapat menjadi referensi dalam mengembangkan sistem penjadwalan mata kuliah daring. Berikut beberapa penelitian yang mengimplimentasikan algoritma genetika dalam menyelesaikan masalah-masalah yang ada.

- a. Erni Masdupi (2014), pada penelitiannya yang berjudul “Optimalisasi Penjadwalan Perkuliahan di Fakultas Ekonomi Universitas Negeri Padang Menggunakan Algoritma Genetika”. Pada penelitian ini, penulis mengklasifikasikan *constraint* ke dalam dua tipe, yaitu *hard constraint* dan *soft constraint*. Pengklasifikasian ini memungkinkan menghasilkan hasil optimal dengan tidak melanggar *hard constraint* dan terjadi beberapa pelanggaran *soft constraint* yang sulit dihindari.
- b. Sam'ani (2012), Rancang Bangun Sistem Penjadwalan Perkuliahan dan Ujian Akhir Semester dengan Pendekatan Algoritma Genetika. Penelitian ini mengambil kasus penjadwalan perkuliahan di STMIK Palangkaraya. Hasil yang didapat adalah penjadwalan dinilai menjadi lebih optimal setelah iterasi yang relatif pendek, yaitu 5 sampai 10 generasi.
- c. Algoritma Genetika telah digunakan untuk menghasilkan nilai maksimum luas *coverage area* dan biaya minimum operasional penempatan armada kapal TNI AL di kawasan timur Indonesia. Sistem yang dibuat diuji dengan penempatan armada kapal sebanyak 27 buah untuk ditempatkan pada 28 pangkalan yang ada. Hasil yang didapatkan adalah luas *coverage area* 1.942.929 mil<sup>2</sup> dan biaya operasional Rp2.853.447.000 dengan nilai fitness terbaik 6,6330. Jika dibandingkan dengan data

lapangan yang ada yaitu total luas area yang harus diamankan sekitar 1.688.765 mil<sup>2</sup> dengan biaya Rp5.000.000.000 maka hasil yang didapatkan lebih efektif dan efisien (Sam'ani, 2012).

- d. Gibbon Mardame Parsaroan Tamba (2004), Sistem Penjadwalan Matakuliah menggunakan Algoritma Genetika (Studi Kasus Fakultas Matematika dan IPA IPB). Dalam penelitian ini, dilakukan metode perbandingan penyusunan jadwal secara manual dengan penyusunan jadwal menggunakan algoritma genetika pada jadwal kuliah Fakultas MIPA IPB. Kesimpulan dari penelitian ini adalah metode Algoritma Genetika dinyatakan efektif untuk menyusun jadwal perkuliahan secara optimal.
- e. Algoritma Genetika juga telah digunakan untuk pengolahan pengenalan citra huruf kapital dengan metode pembangkitan nilai *fitness* berdasarkan transformasi digital seperti translasi, dilatasi dan rotasi. Dalam penelitian tersebut dihasilkan pengenalan yang optimal dengan generasi di atas 2000 (Saputro, 2003).
- f. Sebagai sistem pendukung keputusan teknik perdagangan dalam masalah TSP (*Traveling Salesman Product*). Metode ini telah diteliti oleh Franklin Allen dan Risto Karjalainen dalam jurnal yang berjudul "*Using Genetic Algorithm to Find Technical Trading*" (Allen, 1999).

Berdasarkan beberapa penelitian di atas, terdapat beberapa perbedaan dan kemiripan sistem yang diterapkan pada penelitian ini. Perbedaan tersebut adalah pada penelitian ini, iterasi yang akan berlangsung dapat lebih sedikit. Jumlah iterasi yang sedikit juga harus didukung dengan penetapan argumen kontrol yang tepat. Selain itu, perbedaan yang mendasar adalah, penelitian ini dapat memberikan beberapa pilihan solusi optimal, sehingga kombinasi jadwal yang akan digunakan akan lebih banyak sehingga *user* dapat memilih kromosom yang akan dijadikan sebagai solusi terbaik.

Selain terdapat perbedaan, penelitian ini juga memiliki kemiripan penerapan konsep yang hampir sama dengan penelitian sebelumnya. Pengklasifikasian *constraint* dilakukan berfungsi untuk mengelompokkan *constraint* dengan bobot pelanggaran yang berbeda-beda sehingga solusi optimal dapat dicapai sebagaimana yang diharapkan. Perbedaan yang terdapat pada pengklasifikasian *constraint* adalah jumlah klasifikasi yang digunakan. terdapat dua kelompok peraturan pada penelitian yang dilakukan Erni Masdupi (2014), sedangkan pada penelitian ini, terdapat tiga kelompok peraturan dengan bobot yang berbeda-beda.

## **BAB III**

### **METODOLOGI PENELITIAN**

#### **3.1 Studi Literatur**

Studi literatur dilakukan untuk mencari informasi sebanyak-banyaknya yang berhubungan dengan permasalahan yang akan diteliti dan metode yang akan digunakan untuk menyelesaikan masalah tersebut. Penelitian-penelitian yang relevan akan membantu proses penelitian ini berjalan sebagaimana mestinya. Referensi informasi dan penelitian yang digunakan dapat berupa buku, jurnal, maupun situs internet yang berkaitan dengan algoritma genetika dan implementasinya.

#### **3.2 Sumber Data**

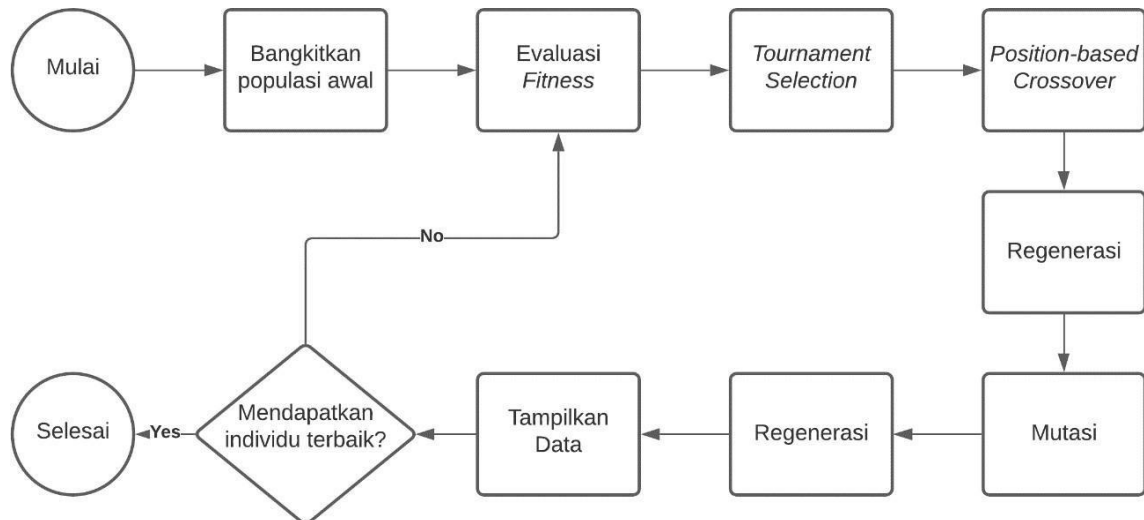
Penelitian mengenai penjadwalan mata kuliah daring ini mengambil kasus pada Jurusan Informatika Universitas Islam Indonesia. Informasi yang berkaitan dengan penjadwalan mata kuliah beserta *constraints* di dalamnya didapatkan melalui Sekretaris Jurusan Informatika Universitas Islam Indonesia, yaitu Bapak Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D. Informasi studi kasus penjadwalan pada Jurusan Informatika UII didapatkan melalui wawancara bersama Beliau, mulai dari sumber data, seperti data mata kuliah, data dosen, data paket mata kuliah dan lainnya dapat diakses melalui *website* resmi Jurusan Informatika UII hingga persyaratan-persyaratan apa saja yang harus diperhatikan dalam proses pembuatan jadwal perkuliahan.

#### **3.3 Perancangan Algoritma Genetika**

Sistem yang akan dibuat adalah pengoptimalan penjadwalan mata kuliah daring pada Program Studi Informatika Universitas Islam Indonesia. Komponen-komponen yang terlibat di dalam proses penjadwalan yang menggunakan metode algoritma genetika adalah penyandian kromosom, inisialisasi populasi awal, fungsi *fitness*, proses seleksi, penerapan operator genetika, dan penetapan argumen kontrol. Pada penelitian ini, pengimplementasian algoritma genetika pada dua buah kasus penjadwalan, yaitu penjadwalan mata kuliah daring dan penjadwalan kelas pengganti mata kuliah. Penjadwalan kelas pengganti mata kuliah merupakan fitur tambahan dalam sistem yang akan dikembangkan. Sistem penjadwalan kelas pengganti mata kuliah merupakan jawaban dari jadwal perkuliahan yang dinamis.

### 3.4 Diagram Blok Penjadwalan Mata Kuliah Daring

Bagan alur penjadwalan mata kuliah daring menggunakan metode algoritma genetika dapat dilihat secara singkat melalui Gambar 3.1 di bawah ini.



Gambar 3.1 Diagram Blok Algoritma Genetika

Merujuk pada Gambar 2.1 yang merupakan bentuk umum metode algoritma genetika, Gambar 3.1 merupakan bentuk kustomisasi yang dilakukan pada penelitian ini. Penyesuaian proses yang diterapkan pada penelitian ini merupakan bentuk implementari nyata dari penggunaan metode algoritma genetika dalam penyelesaian kasus optimasi penjadwalan mata kuliah daring. Berdasarkan Gambar 3.1, langkah-langkah yang akan dilakukan adalah sebagai berikut:

- a. Mulai, melakukan persiapan langkah-langkah algoritma genetika
- b. Bangkitkan populasi awal, proses pembentukan populasi awal yang terdiri dari kromosom-kromosom
- c. Evaluasi *Fitness*, melakukan evaluasi nilai *fitness* untuk setiap kromosom yang ada di dalam populasi
- d. *Tournament selection*, melakukan penyeleksian calon kromosom induk menggunakan metode *tournament selection*.
- e. *Position-based crossover*, menjalankan proses *crossover* menggunakan teknik *position-based crossover*
- f. Regenerasi, proses pembentukan populasi kembali secara utuh setelah didapatkan kromosom anak (*offspring*) dengan jumlah populasi yang tetap dengan populasi awal,

dengan cara mengikutsertakan secara acak beberapa kromosom yang tidak terseleksi pada proses seleksi turnamen.

- g. Mutasi, proses mutasi gen-gen terpilih yang terdapat pada populasi hasil proses regenerasi sebelumnya.
- h. Regenerasi, proses penyusunan ulang kromosom-kromosom baru setelah dilakukan proses mutasi
- i. Menampilkan data, menampilkan data kromosom-kromosom setelah melalui proses regenerasi sebelumnya.
- j. Pengecekan terhadap hasil optimal, jika belum mendapatkan hasil optimal, maka akan kembali ke langkah 3.
- k. Selesai. Proses penjadwalan telah menghasilkan solusi optimal.

### 3.5 Teknik Pengkodean Kromosom

Pembuatan representasi kromosom akan disandikan menggunakan *value decoding*. Sebuah kromosom yang terdiri dari gen-gen yang telah disandikan berupa, mata kuliah, dosen pengajar, hari dan jam perkuliahan. Pada kasus penjadwalan mata kuliah daring ini, parameter ruangan tidak diperlukan. Berikut contoh pengkodean sebuah kromosom.

Tabel 3.1 Contoh *value decoding* sebuah kromosom

GdM	GdM	EP	EP	P-KSIM	P-KSIM
[A]	[B]	[A]	[B]	[A]	[B]
DHANI	AFFAN	A.FATHAN	ARI.S	A.RAFIE	RIDHO
Jumat	Rabu	Jumat	Senin	Senin	Kamis
14:40	08:40	10:05	12:30	08:40	08:40
17:10	11:10	11:45	14:10	11:10	11:10

Tabel 3.1 merupakan contoh potongan kromosom yang akan diinisialisai pada penelitian ini. Berikut beberapa keterangan terkait contoh potongan kromosom di atas:

- a. Baris pertama pada potongan kromosom tersebut (GdM, EP, P-KSIM) merupakan pengkodean terhadap mata kuliah.

- b. Baris kedua ([A]DHANI, [B]AFFAN, ...) merupakan pengkodean terhadap kelas mata kuliah yang akan dijadwalkan pada semester tertentu. Setiap mata kuliah akan memiliki minimal satu kelas yang akan dibuka.
- c. Baris ketiga (Jumat 14:40 17:10, Rabu 8:40 11:10, ...) merupakan pengkodean blok jam yang merepresentasikan jadwal penyelenggaraan kelas mata kuliah.

Berdasarkan potongan kromosom di atas, sebuah kromosom dapat didefinisikan dalam skema sebagai berikut:

$$\text{SesiKelas}[i] = [\text{kodematkul}[i], \text{kodkelasdosen}[i], \text{kodblokjam}[i]]; \quad (3.1)$$

Pengkodean kromosom di atas akan memiliki pengkodean yang akan selalu sama pada setiap kromosom di masing-masing populasi. Pada kasus penjadwalan mata kuliah daring di penelitian ini, pengkodean blok jam akan berperan sebagai gen pada setiap kromosom yang akan mengalami langkah-langkah algoritma genetika. Panjang kromosom ditentukan oleh jumlah mata kuliah yang dibuka pada setiap semester, jumlah kelas yang dibuka untuk tiap mata kuliah, dan bobot sks yang dimiliki tiap mata kuliah. Panjang kromosom dapat dirumuskan sebagai berikut.

$$\text{Panjang Kromosom} = 2 \sum_{i=1}^n x_i + \sum_{i=1}^m y_i \quad (3.2)$$

Keterangan:

$x_i$  : banyak kelas yang dibuka untuk mata kuliah yang memiliki lebih dari 3 sks

$y_i$  : banyak kelas yang dibuka untuk mata kuliah yang memiliki paling besar 3 sks

$n$  : nilai berhenti yang mengacu pada jumlah mata kuliah yang lebih dari 3 sks

$m$  : nilai berhenti yang mengacu pada jumlah mata kuliah yang lebih dari 3 sks

### 3.6 Inisialisasi Populasi

Setelah melakukan pengkodean pada kromosom, populasi pertama akan dibangkitkan secara acak dari seluruh solusi yang ada dan dengan *popsiz*e tertentu. Kromosom akan menyatakan salah satu alternatif solusi yang mungkin terjadi. Jumlah populasi awal yang akan dibangkitkan akan berpengaruh pada keberagaman pada generasi awal itu sendiri. Pemilihan secara random menyebabkan populasi awal kecil kemungkinan akan sama dalam setiap kali

percobaan. Hal ini dikarenakan banyaknya kemungkinan solusi yang akan dihasilkan pada tiap kali pembangkitan populasi awal.

### 3.7 Evaluasi Nilai *Fitness*

Selanjutnya, kromosom akan menampung nilai *fitness* yang didapat melalui fungsi *fitness* yang telah ditetapkan. Nilai *fitness* dihitung melalui jumlah pelanggaran (*penalty*) dari peraturan-peraturan yang ditetapkan yang digunakan untuk mencapai solusi optimal. Berikut adalah daftar peraturan yang diterapkan pada penjadwalan mata kuliah daring:

- (*p*<sub>1</sub>) Mata kuliah yang memiliki lebih dari 3 sks akan dibagi menjadi 2 sesi yang dijadwalkan tidak pada hari yang sama
- (*p*<sub>2</sub>) Mata kuliah yang memiliki lebih dari 3 sks akan dibagi menjadi 2 sesi yang dijadwalkan tidak pada hari yang berurutan
- (*p*<sub>3</sub>) Mata kuliah bergerak tidak dijadwalkan pada waktu yang bersamaan dengan mata kuliah paket di semester 4
- (*p*<sub>4</sub>) Mata kuliah bergerak tidak dijadwalkan pada waktu yang bersamaan dengan mata kuliah paket di semester 3
- (*p*<sub>5</sub>) Mata kuliah bergerak diusahakan tidak dijadwalkan pada waktu yang bersamaan dengan mata kuliah paket di semester 5
- (*p*<sub>6</sub>) Mata kuliah metodologi penelitian tidak dijadwalkan pada waktu yang bersamaan dengan mata kuliah pilihan
- (*p*<sub>7</sub>) Mata kuliah metodologi penelitian tidak dijadwalkan pada waktu yang bersamaan dengan mata kuliah paket di semester 5
- (*p*<sub>8</sub>) Dosen tidak dijadwalkan mengajar lebih dari satu mata kuliah di waktu yang bersamaan
- (*p*<sub>9</sub>) Mata kuliah pilihan tidak dijadwalkan pada waktu yang bersamaan dengan mata kuliah paket di semester 5
- (*p*<sub>10</sub>) Mata kuliah pilihan yang tergolong ke dalam kluster yang sama tidak dijadwalkan pada waktu yang bersamaan
- (*p*<sub>11</sub>) Mata kuliah yang terdapat pada tahun pertama (semester 1 dan 2) dengan kelas yang sama tidak dijadwalkan pada waktu yang bersamaan

Terdapat sebelas peraturan yang menjadi parameter sebuah individu memiliki tingkat kebugaran tertentu. Peraturan-peraturan ini akan dikelompokkan ke dalam tingkat kefatalan pelanggaran yang akan ditimbulkan jika terjadi pelanggaran. Dengan adanya perbedaan pengaruh yang ditimbulkan oleh pelanggaran pada masing-masing peraturan, tentunya juga akan menyebabkan nilai konstanta *penalty* untuk masing-masing peraturan berbeda-beda.



Pelanggaran peraturan tertentu yang tidak dapat ditolerir akan memiliki konstanta *penalty* yang lebih besar, seperti seorang dosen mengajar lebih dari satu kelas mata kuliah pada waktu yang bersamaan. Pasangan peraturan dan konstanta *penalty* dapat dilihat pada tabel 3.1 di bawah ini.

Tabel 3.2 Pasangan peraturan dengan konstanta *penalty*

No.	Peraturan ( $p_i$ )	Konstanta <i>penalty</i> ( $k_i$ )
1	Peraturan 1 ( $p_1$ )	1
2	Peraturan 2 ( $p_2$ )	50
3	Peraturan 3 ( $p_3$ )	50
4	Peraturan 4 ( $p_4$ )	50
5	Peraturan 5 ( $p_5$ )	1
6	Peraturan 6 ( $p_6$ )	1
7	Peraturan 7 ( $p_7$ )	50
8	Peraturan 8 ( $p_8$ )	5000
9	Peraturan 9 ( $p_9$ )	50
10	Peraturan 10 ( $p_{10}$ )	50
11	Peraturan 11 ( $p_{11}$ )	5000

Berdasarkan pasangan peraturan dan konstanta *penalty* yang akan diberikan jika melakukan pelanggaran pada peraturan tertentu, maka fungsi *fitness* dapat didefinisikan ke dalam operasi matematika sebagai berikut:

$$Fitness = \sum_{i=1}^n k_i p_i \quad (3.3)$$

Keterangan:

$k_i$  = koefisien basis *pinalti* untuk pelanggaran peraturan  $p_i$

$p_i$  = banyaknya kemunculan pelanggaran peraturan ke –  $i$

$i$  = nilai yang menyatakan urutan ke –  $i$  (1, 2,3, ... )

$n$  = banyak peraturan yang diterapkan

Penetapan konstanta *penalty* di atas didasarkan pada seberapa fatal sebuah peraturan jika dilanggar. Peraturan-peraturan yang memiliki konstanta *penalty* 5000 memiliki arti bahwa pelanggaran pada peraturan tersebut akan menjadikan sebuah kromosom tidak memenuhi syarat sebagai solusi optimal secara mutlak. Peraturan-peraturan yang memiliki konstanta *penalty* 50 memiliki arti bahwa pelanggaran pada peraturan tersebut menjadikan kromosom tetap dapat diterima sebagai calon solusi optimal pada kondisi tertentu. Terakhir, peraturan-peraturan yang memiliki konstanta *penalty* 1 memiliki arti bahwa pelanggaran pada peraturan tersebut tidak memiliki dampak signifikan pada sebuah kromosom sebagai calon solusi optimal.

Kromosom yang memiliki nilai *fitness* semakin kecil merepresentasikan tingkat kebugaran yang semakin baik sehingga besar kemungkinan akan diikutsertakan pada proses seleksi, *crossover* dan mutasi. Semakin banyak akumulasi *penalty* yang terjadi pada sebuah kromosom berarti semakin menjauhkan kromosom tersebut dari solusi optimal yang ingin dicapai.

### 3.8 Seleksi Induk

Setelah semua kromosom memperoleh nilai *fitness* masing-masing, langkah selanjutnya adalah melakukan proses seleksi. Melalui proses seleksi, akan dipilih beberapa kromosom yang awalnya dijadikan calon kromosom induk dengan diberlakukan operator genetika *crossover*. Teknik seleksi yang digunakan adalah *tournament selection*. Metode ini merupakan penyeleksian sederhana. Seperti namanya, seleksi ini akan diadakan turnamen pada sekumpulan kelompok kecil kromosom, kromosom yang memiliki nilai *fitness* terbaik; yang direpresentasikan pada nilai *fitness* terkecil, akan memenangkan turnamen dan menjadi calon kromosom induk. Pada kasus ini, penyeleksian akan dilakukan pada sepasang kromosom yang dipilih secara acak sehingga akan terpilih minimal 50% dari jumlah populasi awal.

Setelah didapatkan beberapa calon kromosom induk, maka argumen kontrol akan berperan pada tahapan pemilihan kromosom induk. Argumen yang diterapkan adalah probabilitas *crossover* ( $P_c$ ). Nilai probabilitas (rentang 0 sampai 1) akan dibangkitkan sehingga terpilih kromosom induk dengan jumlah sebanyak probabilitas *crossover* dikalikan dengan jumlah calon kromosom induk. Pemilihan ini mengambil kromosom-kromosom terbaik. Nilai  $P_c$  yang disarankan adalah 0,6 sampai 1 sehingga dengan jumlah induk yang semakin banyak yang akan dilakukan proses *crossover*, diharapkan menghasilkan kromosom baru yang lebih baik.

### 3.9 Crossover

Berdasarkan hasil seleksi, kromosom-kromosom terpilih akan dikenai proses *crossover*. Proses ini bekerja pada sepasang kromosom yang dipilih secara acak pada himpunan kromosom induk. Teknik penyilangan yang dipakai adalah *position-based crossover*. Teknik ini dimulai dengan menentukan berapa banyak posisi atau berapa banyak gen yang akan dilakukan penyilangan. Setelah banyak posisi penyilangan dimiliki, kemudian pemilihan posisi akan dilakukan secara acak sebagai nilai posisi penyilangan yang didapatkan sebelumnya. Kemudian elemen dari kedua kromosom akan ditukarkan satu sama lain sehingga menghasilkan dua buah kromosom baru.

Setelah diperoleh kromosom baru yang jumlahnya sama dengan kromosom induk, maka akan dilakukan regenerasi populasi terlebih dahulu dengan cara mengambil semua kromosom induk, kromosom anak dan calon kromosom induk yang sebelumnya tidak terpilih dilakukan operasi *crossover*. Akan tetapi, langkah ini belum memenuhi jumlah kromosom seperti semula. Maka akan dipilih dari kromosom-kromosom yang kalah dalam proses seleksi turnamen secara acak. Banyak kromosom yang dipilih adalah  $(1 - P_c)$  dikalikan dengan jumlah kromosom yang kalah turnamen. Dengan begitu, jumlah kromosom akan sama seperti semula.

### 3.10 Mutasi

Tahapan yang dilakukan setelah terjadinya regenerasi setelah proses *crossover* adalah proses mutasi. Sebelum melakukan proses mutasi, harus dibangkitkan probabilitas mutasi (dinotasikan sebagai  $P_m$ ). Probabilitas mutasi mengacu pada rasio antara jumlah gen yang diharapkan mengalami proses mutasi pada setiap generasi dengan jumlah gen total dalam sebuah populasi. Karena mutasi merupakan operator sekunder dalam algoritma genetika, probabilitas mutasi diharapkan bernilai cukup rendah. Jika tingkat mutasi terlalu besar, maka jumlah gen yang mengalami mutasi juga semakin banyak. Hal ini menyebabkan banyak mutan yang akan muncul sehingga mengakibatkan banyak sifat kromosom induk yang kemungkinan hilang pada generasi selanjutnya. Sedangkan pemilihan kromosom induk sudah dilakukan dengan memilih kromosom-kromosom terbaik.

Selanjutnya dibangkitkan nilai acak antara 0 sampai 1 pada masing-masing gen. Ketentuan gen yang akan mengalami mutasi adalah nilai acak yang diberikan pada setiap gen harus lebih kecil dari nilai probabilitas mutasi. Jika terdapat gen yang memiliki nilai acak lebih kecil dari nilai probabilitas mutasi, maka gen tersebut akan ditukar secara acak. Tujuan dari proses ini

adalah diharapkan dapat mengembalikan gen-gen yang hilang pada generasi sebelumnya, atau memunculkan gen baru yang belum pernah muncul.

### 3.11 Regenerasi dan Iterasi

Setelah sebuah populasi awal dilakukan pengerjaan langkah-langkah algoritma genetika dalam sekali pemrosesan, maka proses tersebut akan menghasilkan generasi baru yang masih memiliki beberapa sifat dari populasi sebelumnya. Selanjutnya, generasi baru ini akan menjadi populasi awal pada iterasi selanjutnya. Proses perulangan ini akan berlangsung hingga mencapai generasi yang diharapkan *user*.

### 3.12 Perancangan Database

Untuk menampung data-data yang akan digunakan dalam proses penjadwalan mata kuliah dibuat *database* menggunakan MySQL. Berikut adalah daftar tabel yang berisi atribut di dalamnya beserta hubungan relasi yang dimiliki oleh setiap tabel dengan tabel lainnya.

#### a. Tabel blok\_jam

Tabel blok\_jam merupakan tabel untuk menyimpan data terkait jadwal jam perkuliahan yang diadakan. Tabel ini memiliki relasi 1:M (*One-to-Many*) dengan tabel generate\_jkp\_kelas\_sesi. Artinya, setiap blok jam dapat memiliki banyak *generate* jadwal kuliah pengganti sesi kelas. Berikut merupakan struktur tabel blok\_jam:

Tabel 3.3 Struktur Tabel blok\_jam

Nama Kolom	Tipe	Keterangan
id	int(11)	Primary Key
hari	enum('Senin','Selasa','Rabu','Kamis','Jumat','Sabtu','Minggu')	No
jam_mulai	time	No
jam_selesai	time	No

#### b. Tabel dosen

Tabel dosen merupakan tabel untuk menyimpan data terkait informasi dosen. Tabel ini memiliki relasi 1:M (*One-to-Many*) dengan tabel generate\_jkp\_kelas dan tabel generate\_jkd\_deosen\_mengajar\_matkul. Berikut struktur tabel dosen:

Tabel 3.4 Struktur Tabel dosen

Nama Kolom	Tipe	Keterangan
id	int(11)	Primary Key
nik	varchar(20)	No
nama_dosen	varchar(255)	No
nickname	varchar(255)	No

## c. Tabel generate\_jkd

Tabel generate\_jkd merupakan tabel yang berisi data untuk proses *generate* jadwal mata kuliah daring. Tabel ini memiliki relasi 1:M (*One-to-Many*) dengan tabel generate\_jkd\_generation dan tabel generate\_jkd\_deosen\_mengajar\_matkul. Berikut struktur tabel generate\_jkd:

Tabel 3.5 Struktur Tabel generate\_jkd

Nama Kolom	Tipe	Keterangan
id	int(11)	Primary Key
created_at	datetime	Index
tahun	int(11)	No
semester	enum('ganjil','genap')	Index

## d. Tabel generate\_jkd\_dosen\_mengajar\_matkul

Tabel generate\_jkd\_dosen\_mengajar\_matkul merupakan tabel yang berisi data mata kuliah yang akan dilakukan *generate* beserta dosen-dosen yang mengampu setiap mata kuliah tersebut. Tabel ini memiliki relasi 1:M (*One-to-Many*) dengan tabel mata\_kuliah, tabel generate\_jkd dan tabel dosen. Berikut struktur tabel generate\_jkd\_mengajar\_matkul:

Tabel 3.6 Struktur Tabel generate\_jkd\_dosen\_mengajar\_matkul

Nama Kolom	Tipe	Keterangan
id	int(11)	Primary Key
id_generate	int(11)	Index
id_matkul	int(11)	Index

id_dosen	int(11)	Index
kelas	int(11)	No

e. Tabel generate\_jkd\_generation

Tabel generate\_jkd\_generation merupakan tabel yang berisi data generasi dari proses generate jadwal mata kuliah daring. Tabel ini memiliki relasi 1:M (*One-to-Many*) dengan tabel generate\_jkd. Berikut struktur tabel generate\_jkd:

Tabel 3.7 Struktur Tabel generate\_jkd\_generation

Nama Kolom	Tipe	Keterangan
id	int(11)	Primary Key
id_generate_jkd	int(11)	Index
created_at	datetime	No
generation_number	int(11)	Index
parent_generation	int(11)	Index
generation_data	longtext	No

f. Tabel generate\_jkp

Tabel generate\_jkp merupakan tabel yang berfungsi untuk menyimpan data proses generate jadwal kuliah pengganti. Tabel ini memiliki relasi 1:M (*One-to-Many*) dengan tabel generate\_jkp\_generation dan tabel generate\_jkp\_kelas.. Di bawah ini struktur tabel generate\_jkp:

Tabel 3.8 Struktur Tabel generate\_jkp

Nama Kolom	Tipe	Keterangan
id	int(11)	Primary Key
created_at	datetime	No
tahun	int(11)	Index
semester	enum('ganjil', 'genap')	Index

## g. Tabel generate\_jkp\_generation

Tabel generate\_jkp\_generation merupakan tabel yang berisi data generasi dari proses generate jadwal mata kuliah pengganti. Tabel ini memiliki relasi 1:M (*One-to-Many*) dengan tabel generate\_jkp. Berikut struktur tabel generate\_jkp\_generation:

Tabel 3.9 Struktur Tabel generate\_jkd\_generation

Nama Kolom	Tipe	Keterangan
id	int(11)	Primary Key
id_generate_jkp	int(11)	Index
created_at	datetime	No
generation_number	int(11)	Index
parent_generation	int(11)	Index
generation_data	longtext	No

## h. Tabel generate\_jkp\_kelas

Tabel generate\_jkp\_kelas merupakan tabel yang berfungsi untuk menyimpan data generate jadwal kuliah pengganti. Tabel ini memiliki relasi 1:M (*One-to-Many*) dengan tabel generate\_jkp, tabel mata\_kuliah, tabel dosen dan tabel generate\_jkp\_kelas\_sesi. Di bawah ini struktur tabel generate\_jkp\_kelas:

Tabel 3.10 Struktur Tabel generate\_jkp\_kelas

Nama Kolom	Tipe	Keterangan
id	int(11)	Primary Key
id_generate	int(11)	Index
id_matkul	int(11)	Index
id_dosen	int(11)	Index
kelas	int(11)	No



## i. Tabel generate\_jkp\_kelas\_sesi

Tabel generate\_jkp\_kelas\_sesi merupakan tabel yang berfungsi untuk menyimpan data hasil *generate* jadwal kuliah pengganti. Tabel ini memiliki relasi 1:M (*One-to-Many*) tabel generate\_jkp\_kelas dan tabel blok\_jam dengan tabel kelas dan tabel blok\_jam. Di bawah ini struktur tabel generate\_jkp\_kelas\_sesi:

Tabel 3.11 Struktur Tabel generate\_jkp\_kelas\_sesi

Nama Kolom	Tipe	Keterangan
id	int(11)	Primary Key
id_kelas	int(11)	Index
sesi	int(11)	Index
id_blok_jam	int(11)	Index

## j. Tabel generate\_jkp\_mahasiswa\_mengambil\_kelas

Tabel generate\_jkp\_mahasiswa\_mengambil\_kelas merupakan tabel yang berisi info mahasiswa beserta kelas yang diambilnya. Tabel ini memiliki relasi 1:M (*One-to-Many*) dengan tabel generate\_jkp\_kelas dan tabel mahasiswa. Di bawah ini struktur tabel generate\_jkp\_mahasiswa\_mengambil\_kelas:

Tabel 3.12 Struktur Tabel generate\_jkp\_mahasiswa\_mengambil\_kelas

Nama Kolom	Tipe	Keterangan
id	int(11)	Primary Key
nim_mahasiswa	int(11)	Index
id_kelas	int(11)	Index

## k. Tabel kelas

Tabel kelas merupakan tabel yang berfungsi untuk menyimpan data kelas mata kuliah yang diselenggarakan. Tabel ini tidak memiliki relasi apapun dengan tabel yang ada di *database*. Di bawah ini struktur tabel kelas:

Tabel 3.13 Struktur Tabel kelas

Nama Kolom	Tipe	Keterangan
------------	------	------------

id	int(11)	Primary Key
id_mengajar	int(11)	Index
kelas	int(11)	Index

l. Tabel kluster\_matkul

Tabel kluster\_matkul merupakan tabel yang berfungsi untuk menyimpan data kluster mata kuliah pilihan yang berisi beberapa mata kuliah pilihan dengan kluster yang sama. Tabel ini tidak memiliki relasi apapun dengan tabel yang ada di *database*. Di bawah ini struktur tabel kluster\_matkul:

Tabel 3.14 Struktur Tabel kluster\_matkul

Nama Kolom	Tipe	Keterangan
id	int(11)	Primary Key
nama_kluster	varchar(255)	No

m. Tabel mahasiswa

Tabel mahasiswa merupakan tabel yang berfungsi untuk menyimpan data mahasiswa aktif. Tabel ini hanya memiliki relasi 1:M (*One-to-Many*) dengan tabel generate\_jkp\_mahasiswa\_mengambil\_kelas. Di bawah ini struktur tabel mahasiswa:

Tabel 3.15 Struktur Tabel mahasiswa

Nama Kolom	Tipe	Keterangan
id	int(11)	Primary Key
nama_mahasiswa	varchar(255)	No
angkatan	int(11)	No

n. Tabel mata\_kuliah

Tabel mata\_kuliah merupakan tabel yang berfungsi untuk menyimpan data mata kuliah wajib Jurusan Informatika UII. Tabel ini memiliki relasi 1:M (*One-to-Many*) dengan tabel generate\_jkp\_kelas dan tabel generate\_jkd\_dosen\_mengajar\_matkul. Di bawah ini struktur tabel mata\_kuliah:

Tabel 3.16 Struktur Tabel mata\_kuliah

<b>Nama Kolom</b>	<b>Tipe</b>	<b>Keterangan</b>
id	int(11)	Primary Key
nama_matkul	varchar(255)	No
singkatan	varchar(255)	No
kode	varchar(20)	Index
sks	int(11)	No
semester	enum('ganjil','genap','ganjil_genap')	Index
jenis	enum('reguler','pilihan')	Index
Syarat_minimal_sks	int(11)	No

## o. Tabel mata\_kuliah\_paket

Tabel mata\_kuliah\_paket adalah tabel relasi antara tabel paket dengan tabel mata\_kuliah. Tabel ini memiliki relasi 1:M (*One-to-Many*) dengan tabel mata\_kuliah. Artinya, setiap paket mata kuliah di tiap semester dapat memiliki banyak mata kuliah. Sebaliknya, tiap mata kuliah tidak dapat memiliki banyak paket yang direfleksikan pada setiap semesternya. Struktur tabel mata\_kuliah\_paket dapat dilihat di bawah ini:

Tabel 3.17 Struktur Tabel mata\_kuliah\_paket

<b>Nama Kolom</b>	<b>Tipe</b>	<b>Keterangan</b>
id	int(11)	Primary Key
id_paket	int(11)	Index
id_matkul	int(11)	Index

## p. Tabel mata\_kuliah\_pilihan

Tabel mata\_kuliah\_pilihan adalah tabel relasi antara tabel mata\_kuliah dengan tabel kluster. Tabel ini memiliki relasi 1:M (*One-to-Many*) dengan tabel mata\_kuliah. Artinya, setiap paket mata kuliah di tiap semester dapat memiliki banyak mata kuliah. Struktur tabel mata\_kuliah\_pilihan dapat dilihat di bawah ini:

Tabel 3.18 Struktur Tabel mata\_kuliah\_pilihan

<b>Nama Kolom</b>	<b>Tipe</b>	<b>Keterangan</b>
id	int(11)	Primary Key
id_matkul	int(11)	Index
id_kluster	int(11)	Index

## q. Tabel paket

Tabel paket adalah tabel yang berisi data paket mata kuliah di setiap semester. Tabel ini memiliki relasi 1:M (*One-to-Many*) dengan tabel mata\_kuliah. Artinya, setiap paket mata kuliah di tiap semester dapat memiliki banyak mata kuliah. Struktur tabel mata\_kuliah\_paket dapat dilihat di bawah ini:

Tabel 3.19 Struktur Tabel paket

<b>Nama Kolom</b>	<b>Tipe</b>	<b>Keterangan</b>
id	int(11)	Primary Key
semester	int(11)	Index

## BAB IV

### PEMBAHASAN DAN HASIL

#### 4.1 Implementasi Program

Implementasi merupakan hasil dari rancangan yang telah dibuat sebelumnya. Rancangan yang ada dituangkan ke dalam *script* sehingga aplikasi dapat dioperasikan secara nyata. Program dibuat menggunakan komputer dengan spesifikasi perangkat keras dan perangkat lunak seperti di bawah ini.

a. Kebutuhan Perangkat Keras (*hardware*)

Pemrograman dilakukan menggunakan *desktop computer* dengan spesifikasi sebagai berikut:

- Processor Intel(R) Core(TM) i3-4150 CPU @ 3.50GHz
- Memory (RAM) 8.00 GB
- Harddisk SSD 120 GB
- VGA 2 GB
- Sistem Operasi Windows 10

b. Kebutuhan Perangkat Lunak (*software*)

*Software* yang digunakan dalam membangun program ini adalah sebagai berikut:

- XAMPP sebagai *web server* Apache dan *database* MySQL
- Visual Studio Code sebagai aplikasi pemrograman
- Chrome sebagai aplikasi untuk menampilkan hasil pemrograman
- PHP merupakan bahasa pemrograman yang digunakan

Modul utama yang dimiliki sistem adalah sistem penjadwalan mata kuliah daring dan sistem penjadwalan pengganti jadwal mata kuliah. Setelah melakukan pemrograman sistem yang akan dibangun, hasil dari pemrograman tersebut dapat dilihat pada tangkapan layar di bawah ini.

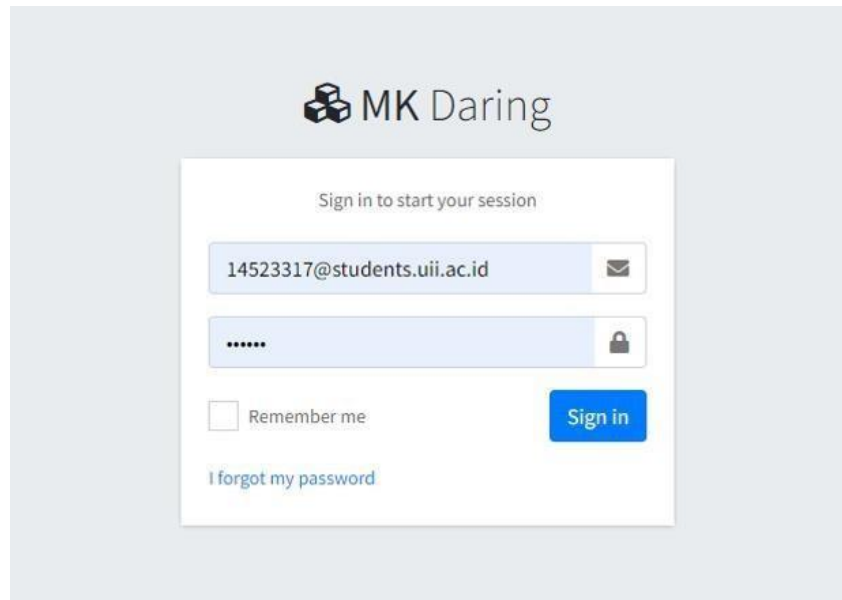
##### 4.1.1 Penjadwalan Mata Kuliah Daring

Pada bagian ini, akan ditunjukkan langkah-langkah yang disusun secara terstruktur dalam melakukan penjadwalan mata kuliah daring. Pada implementasi penjadwalan mata kuliah

daring, akan dilakukan 2 proses, yang pertama adalah, langkah-langkah *generate* penjadwalan, dan yang kedua adalah pengujian algoritma genetika.

a. Langkah-Langkah Penjadwalan Mata Kuliah Daring

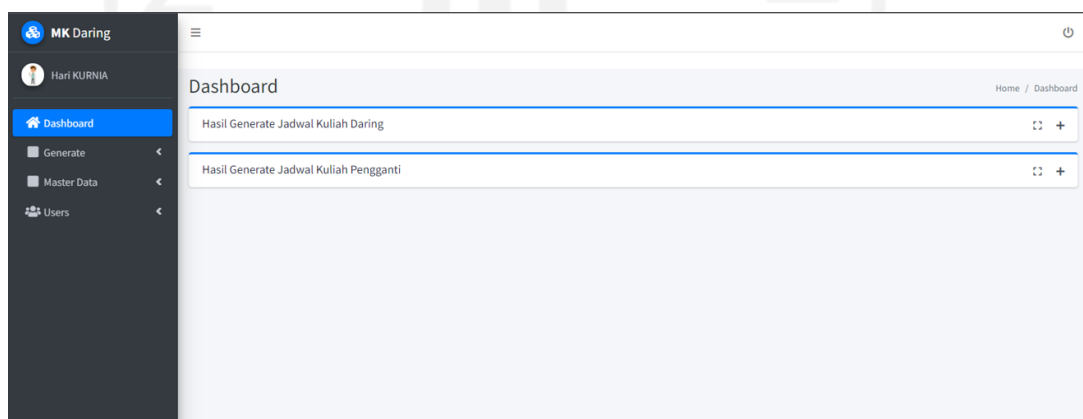
1. Halaman *Login*



Gambar 4.1 Tangkapan Layar Halaman *Login*

Halaman *login* berfungsi untuk gerbang awal dalam mengakses sistem secara menyeluruh. Pengguna dapat melakukan *input email* dan *password* jika sudah terdaftar di dalam sistem sebagai *user*.

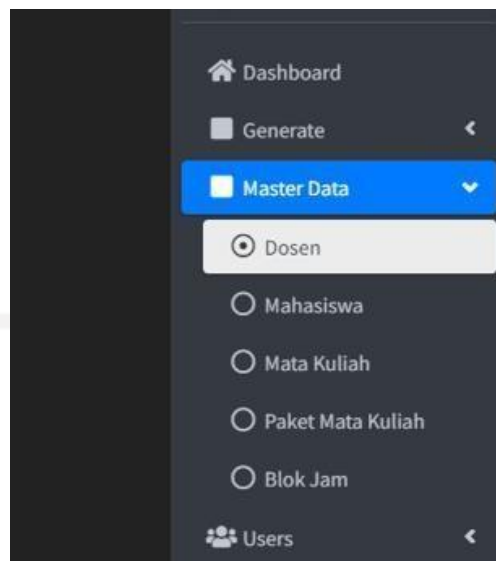
2. Halaman *Dashboard*



Gambar 4.2 Tangkapan Layar Halaman *Dashboard*

Halaman *dashboard* merupakan halama awal yang akan dijumpai ketika berhasil melakukan proses *login*. Halaman ini berisi riwayat hasil dari proses *generate* yang telah dilakukan, baik *generate* jadwal kuliah daring, maupun jadwal pengganti mata kuliah.

### 3. Menu *Master Data*



Gambar 4.3 Tangkapan Layar Menu *Master Data*

Menu ini memiliki beberapa halaman yang dapat diakses di dalamnya, seperti dilihat pada Gambar 4.3, terdapat *sub-menu* Dosen, Mahasiswa, Paket Mata Kuliah, dan Blok Jam. Berikut salah satu contoh tangkapan layar dari daftar tersebut.

Dosen Master Data Home / Master Data / Dosen

Data Dosen

[+ Tambah](#) Cari:

Tampilan 10 entri

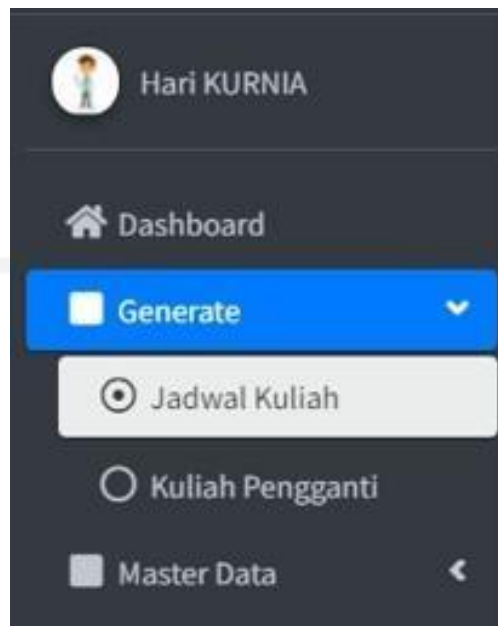
NIK	Nama	Action
000523045	Zainudin Zukhri, S.T., MIT.	<a href="#">Edit</a> <a href="#">Delete</a>
000523044	Taufiq Hidayat, S.T., MCS.	<a href="#">Edit</a> <a href="#">Delete</a>
000523043	Sri Mulyati, S.Kom., M.Kom.	<a href="#">Edit</a> <a href="#">Delete</a>
000523042	Sheila Nurul Huda, S.Kom., M.Cs.	<a href="#">Edit</a> <a href="#">Delete</a>
000523041	Septia Rani, S.T., M.Cs.	<a href="#">Edit</a> <a href="#">Delete</a>
000523040	Rian Adam Rajagede, S.Kom., M.Cs.	<a href="#">Edit</a> <a href="#">Delete</a>
000523039	Rahadian Kurniawan, S.Kom., M.Kom.	<a href="#">Edit</a> <a href="#">Delete</a>

Gambar 4.4 Tangkapan Layar Halaman Daftar Dosen

Halaman ini berisi informasi dosen yang mengajar di Program Studi Informatika Universitas Islam Indonesia. Bagian atas halaman ini, terdapat tombol “Tambah” yang berfungsi untuk menambahkan data ke dalam tabel dosen di bawahnya. Kolom “Action” yang terdapat pada tabel dosen berfungsi sebagai menu sunting informasi yang ada pada tabel dosen, baik mengubah informasi, maupun menghapus data tertentu. Struktur halaman ini juga dimiliki oleh *sub-menu* lainnya yang terdapat pada *menu Master Data* sehingga dapat diwakilkan dengan halaman dosen di atas.



#### 4. Menu *Generate*



Gambar 4.5 Menu *Generate*

Menu ini memiliki 2 buah *sub-menu*, yaitu Jadwal Kuliah dan Kuliah Pengganti. Pada sistem penjadwalan mata kuliah daring, menu yang dipilih adalah menu Jadwal Kuliah. Setelah memilih menu ini, akan tampil halaman seperti berikut.

Jadwal Kuliah Daring Generate Home / Generate / Jadwal Kuliah Daring

List Generate Jadwal Kuliah Daring

[+ Tambah](#) Cari:

Tampilan  entri

ID	Tahun	Semester	Tanggal Buat	Action
1	2022	genap	2021-05-28 18:48:24	<a href="#">Edit</a> <a href="#">Delete</a>
2	2022	genap	2021-05-28 20:31:41	<a href="#">Edit</a> <a href="#">Delete</a>
4	2022	ganjil	2021-06-02 19:35:53	<a href="#">Edit</a> <a href="#">Delete</a>
5	2022	genap	2021-06-02 19:49:18	<a href="#">Edit</a> <a href="#">Delete</a>

Gambar 4.6 Halaman *Generate* Mata Kuliah Daring

Berdasarkan Gambar 4.6 di atas, halaman awal yang muncul ketika *user* menekan tombol “Kuliah Daring” pada *sub-menu Generate* adalah daftar proses membangkitkan jadwal kuliah daring yang diinginkan. Halaman ini menjadi tahapan awal bagi *user* dalam penjadwalan mata kuliah daring, karena pada halaman ini, *user*

harus menambahkan periode perkuliahan yang diinginkan dan akan dilakukan penjadwalan. Proses penjadwalan mata kuliah daring akan dimulai pada tahapan-tahapan di bawah ini.

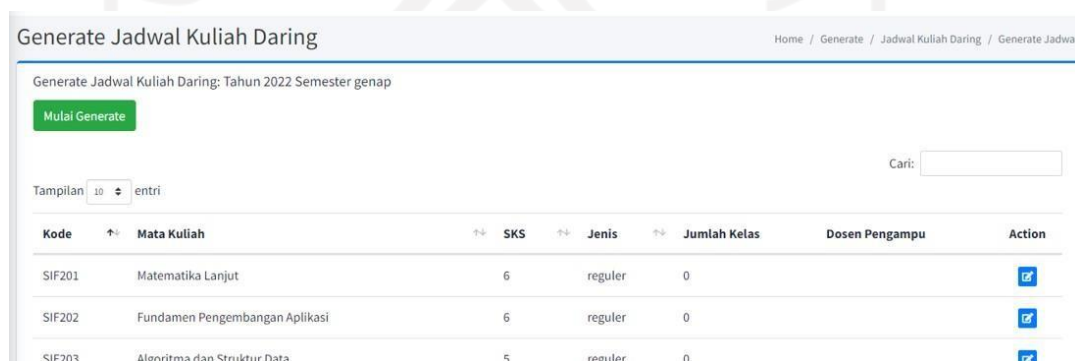
(a). *Input* Periode Perkuliahan






Gambar 4.7 Proses Penambahan Periode Perkuliahan

Setelah melakukan *input* periode perkuliahan, data akan ditampilkan pada tabel yang ada di halaman tersebut. Selain informasi periode yang berisi tahun ajar dan semester, proses ini juga akan menyimpan informasi daftar mata kuliah yang akan dilaksanakan pada periode tersebut sehingga pada tahapan selanjutnya akan dijumpai informasi terkait.

(b). *Input* Dosen Pengajar Mata Kuliah



Kode	Mata Kuliah	SKS	Jenis	Jumlah Kelas	Dosen Pengampu	Action
SIF201	Matematika Lanjut	6	reguler	0		
SIF202	Fundamen Pengembangan Aplikasi	6	reguler	0		
SIF203	Algoritma dan Struktur Data	5	reguler	0		

Gambar 4.8 Tangkapan Layar Halaman *Input* Dosen Pengajar Mata Kuliah

Halaman ini merupakan hasil dari memilih tombol “Edit” pada kolom *Action* di tabel periode penjadwalan mata kuliah daring. Halaman ini menampilkan daftar mata kuliah yang akan dilangsungkan pada periode perkuliahan yang dipilih pada proses

sebelumnya. Sebelum melakukan proses *generate* jadwal mata kuliah daring melalui tombol “Mulai *Generate*”, dibutuhkan informasi pengajar yang akan mengampu setiap mata kuliah yang ada pada daftar. Proses memasukkan informasi pengajar dapat dilakukan melalui tombol “Action” yang pada pada kolom terakhir di tabel mata kuliah. Proses tersebut dapat dilihat di bawah ini.



Gambar 4.9 Tangkapan Layar Proses *Input* Data Dosen Pengajar

Berdasarkan Gambar 4.9, sistem telah menyediakan kolom-kolom kelas yang akan dibuka untuk mata kuliah tertentu, pengguna dapat menambah dan menghapus kolom tersebut dengan menyesuaikan banyak kelas yang akan dibuka. Setiap kelas yang dibuka, akan dimasukkan nama dosen yang akan mengampu kelas tersebut. Setelah menekan tombol “Simpan” pada setiap mata kuliah yang ada di daftar, daftar dosen yang mengajar pada tiap mata kuliah akan tampil pada tabel, seperti tangkapan layar di bawah ini.

Mata Kuliah	SKS	Jenis	Jumlah Kelas	Dosen Pengampu
Matematika Lanjut	6	reguler	5	KELAS A : Feri Wijayanto, M.T. KELAS B : Rahadian Kurniawan, S.Kom., M.Kom. KELAS C : Ahmad M. Raf'ie Pratama, S.T., MIT., Ph.D. KELAS D : Ahmad Luthfi, S.Kom., M.Kom. KELAS E : Kholid Haryono, S.T., M.Kom.
Fundamen Pengembangan Aplikasi	6	reguler	5	KELAS A : Kurniawan Dwi Irianto, S.T., M.Sc. KELAS B : Rahadian Kurniawan, S.Kom., M.Kom. KELAS C : Dr. Sri Kusumadewi, S.Si., M.T. KELAS D : Hanson Prihantoro Putro, M.T. KELAS E : Lizda Iswari, S.T., M.Sc.
Algoritma dan Struktur Data	5	reguler	5	KELAS A : Ahmad M. Raf'ie Pratama, S.T., MIT., Ph.D. KELAS B : Affan Mahtarami, S.Kom., M.T. KELAS C : Erika Ramadhani, S.T., M.Eng.

Gambar 4.10 Tangkapan Layar Daftar Dosen Mengampu Mata Kuliah

Setelah semua mata kuliah yang ada pada tabel dilakukan *input* dosen pengampu, *user* dapat menekan tombol “Mulai *Generate*” untuk melangkah ke tahapan selanjutnya.

(c). *Input* Argumen Algoritma Genetika

Generate Jadwal Kuliah  
Daring: Argumen Algoritma Genetika

Home / Generate / Jadwal Kuliah Daring  
/ Generate Jadwal / Argumen

Argumen Algoritma Genetika

Jumlah Populasi:

Probabilitas Crossover:   
Probabilitas harus antara 0 s/d 1

Banyak Titik Crossover:

Probabilitas Mutasi:   
Probabilitas harus antara 0 s/d 1

Gambar 4.11 Tangkapan Layar *Input* Argumen Algoritma Genetika

Halaman ini akan menyediakan modul *input* argumen-argumen yang diperlukan dalam proses algoritma genetika. Terdapat 4 argumen, di antaranya jumlah populasi, banyak titik *crossover*, probabilitas *crossover*, dan probabilitas mutasi. Argumen jumlah populasi adalah banyaknya kromosom yang akan dibangkitkan pada populasi awal. Banyak titik *crossover* adalah jumlah titik penyilangan yang akan dilakukan menggunakan operator *crossover*. Probabilitas *crossover* adalah rasio pemilihan kromosom dari calon induk yang telah melalui proses seleksi menjadi kromosom

induk sehingga dapat dilakukan proses *crossover* dan menghasilkan kromosom baru. Probabilitas mutasi adalah rasio yang dibutuhkan dalam menentukan jumlah gen yang akan dilakukan proses mutasi terhadap keseluruhan jumlah gen yang terdapat di dalam populasi.

Setelah melakukan *input* keempat argumen di atas, *user* dapat memulai proses pembentukan generasi selanjutnya dengan proses *generate* yang telah dirancang melalui *script* PHP pada aplikasi Visual Studio Code.

#### (d). *Input* Jumlah Generasi

Generasi Algoritma Genetika: Progress Generasi

Jumlah Populasi 100 Probabilitas Crossover 0.6

Banyak Titik Crossover 5 Probabilitas Mutasi 0.008

Jumlah Generasi Baru

**Tambah Generasi**

---

Generasi Algoritma Genetika: Best Kromosom

---

Generasi Algoritma Genetika: Generasi Terakhir

[INDEX: k1] [FITNESS: 66774] [Lihat Hasil](#) [Export Hasil ke Jadwal Kuliah Pengganti](#)

ML					FPA					ASD															
[A]	[B]	[C]	[D]	[E]	[A]	[B]	[C]	[D]	[E]	[A]	[B]	[C]													
FERI	RAHADIAN	A.RAFIE	A.LUTHFI	KHOLID	KURNIAWAN	RAHADIAN	CICI	HANSON	LIZDA	A.RAFIE	AFFAN	ERIK													
Jumat	Kamis	Senin	Senin	Kamis	Kamis	Rabu	Kamis	Rabu	Rabu	Senin	Kamis	Jumat	Jumat	Kamis	Jumat	Rabu	Senin	Senin	Rabu	Jumat	Rabu	Senin	Jumat	Rabu	R
14:40	09:30	08:40	07:00	08:40	12:30	12:30	09:30	12:30	08:40	07:50	12:30	06:45	07:35	07:00	07:35	08:40	07:00	08:40	07:00	06:45	12:30	09:30	13:00	12:30	0

Gambar 4.12 Tangkapan Layar Tambah Generasi

Setelah menekan tombol “Mulai *Generate*”, sistem akan melakukan proses pembangkitan populasi awal yang berisi informasi-informasi dari argumen yang sudah ditetapkan pada halaman sebelumnya. Pada halaman ini, akan ditampilkan juga semua kromosom yang dibangkitkan. Pembangkitan populasi awal ini dilakukan secara acak dengan memperhatikan tujuan calon solusi yang diinginkan. Selain itu, kromosom terbaik pada setiap generasi juga ditampilkan sehingga dapat menjadi referensi perkembangan nilai *fitness* pada setiap generasi.

Setelah tombol “Tambah Generasi” ditekan dengan informasi jumlah generasi yang diinginkan dimasukkan pada *insert section box*, maka proses pembentukan generasi akan dilakukan secara *backend* dan akan kembali ke halaman populasi awal. Berikut proses menambah generasi sesuai *input* yang dilakukan oleh *user*.

```
PS C:\Users\Jarkumande\Documents\lama kali TA nya\nih TA nya> php artisan queue:work --queue=jadwal_kuliah_daring --timeout=7200
[2021-11-29 04:23:42][67] Processing: App\Jobs\ProcessGenerateJKD
[2021-11-29 04:38:30][67] Processed: App\Jobs\ProcessGenerateJKD
```

Gambar 4.13 Proses Pembentukan Generasi Baru

Setelah proses selesai, halaman generasi akan menampilkan jumlah generasi yang dihasilkan. Seperti gambar di bawah ini, telah diciptakan 200 generasi, dan akan ditampilkan kromosom-kromosom pada generasi ke-200.

[INDEX: k200.1] [FITNESS: 5] [PREVIOUS INDEX: k199.1]

ML										FPA					
[A]		[B]		[C]		[D]		[E]		[A]		[B]		[C]	
FERI		RAHADIAN		A.RAFIE		A.LUTHFI		KHOLID		KURNIAWAN		RAHADIAN		CICI	
Senin	Rabu	Senin	Jumat	Senin	Jumat	Senin	Jumat	Kamis	Rabu	Jumat	Senin	Rabu	Kamis	Kamis	Rabu
07:00	07:00	07:00	14:40	12:30	14:40	09:30	07:35	12:30	09:30	14:40	12:30	07:00	12:30	08:40	12:30
09:30	09:30	09:30	17:10	15:00	17:10	12:00	10:05	15:00	12:00	17:10	15:00	09:30	15:00	11:10	15:00
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

[INDEX: k200.2] [FITNESS: 5] [PREVIOUS INDEX: c(k199.10:k199.4)]

Gambar 4.14 Generasi Baru

Selain itu, dengan menekan nilai *fitness* yang akan muncul, maka informasi pelanggaran yang terjadi pada kromosom tersebut akan ditampilkan di layar. Informasi tersebut akan merepresentasikan nilai *fitness* yang dimiliki oleh sebuah kromosom, berikut ilustrasinya.

Fitness: 5

```

Array
(
    [Initial Fitness] => 0
    [Dosen Sama Beda Sesi Tabrakan [5000]] => 0
    [Tahun Pertama Kelas Sama Tabrakan [5000]] => 0
    [Mata Kuliah Sesi Lebih (> 3 sks) Hari-Berurutan (Hari Sama => [50], Hari Beda => [1])] => 5
    [Mata Kuliah Bergerak Tabrakan dengan Paket (Semester 3 => [50], 4 => [50], 5 => [1])] => 0
    [Mata Kuliah Pilihan Tabrakan dengan Paket (Semester 5 => [50])] => 0
    [Mata Kuliah Metapel Tabrakan (Paket => [50], Pilihan => [1])] => 0
    [Mata Kuliah Kluster Sama Tabrakan [50]] => 0
)

```

Kuliah Pengganti [INDEX: k200.1] [FITNESS: 5] [PREVIOUS INDEX: k199.1]

Gambar 4.15 Daftar Pelanggaran

Kromosom dengan jumlah pelanggaran paling sedikit, maka nilai *fitness* yang dimilikinya akan semakin mendekati angka 0 (nol), begitu juga sebaliknya. Berdasarkan fungsi tujuan tersebut, solusi optimal ditemukan melalui kromosom dengan nilai *fitness* terkecil yang dihasilkan.

(e). Mencetak Hasil *Generate*

Generasi terbaik yang direpresentasikan oleh kromosom terbaik di generasinya dapat digunakan sebagai solusi optimal dari proses penjadwalan mata kuliah daring. Kromosom yang digunakan merupakan representasi dari jadwal yang akan diadakan dan ditawarkan kepada mahasiswa ketika masa KRS dimulai. Hasil optimal yang didapat dapat ditampilkan dalam bentuk tabel jadwal mata kuliah yang lebih mudah dibaca dan dipahami. Tabel hasil optimal dapat dilihat pada gambar dibawah ini.

Keamanan Siber dan Informasi	3	A	Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D.	Rabu	09:30 - 12:00
		B	Beni Suranto, M.SoftEng.	Rabu	07:50 - 10:20
Forensika Digital	3	A	Ari Sujarwo, MIT(Hons)	Jumat	13:50 - 16:20
Internet of Things	3	A	Aridhanyati Arifin, S.T., M.Cs.	Kamis	07:00 - 09:30
		B	Dr. Mukhammad Andri Setiawan, S.T., M.Sc.	Rabu	12:30 - 15:00
Teknologi Blockchain	3	A	Aridhanyati Arifin, S.T., M.Cs.	Rabu	12:30 - 15:00
		B	Dr. Ing. Ridho Rahmadi, S.Kom., M.Sc.	Senin	08:40 - 11:10
Manajemen Teknologi Informasi	3	A	Andhik Budi Cahyono, M.T.	Kamis	07:00 - 09:30
		B	Chanifah Indah Ratnasari, S.Kom., M.Kom.	Kamis	07:00 - 09:30
Teknologi Informasi untuk Pembangunan (ICT4D)	3	A	Beni Suranto, M.SoftEng.	Jumat	08:25 - 10:55
		B	Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D.	Senin	08:40 - 11:10
Metodologi Penelitian	3	A	Zainudin Zuhri, S.T., MIT.	Senin	12:30 - 15:00
		B	Beni Suranto, M.SoftEng.	Senin	07:00 - 09:30

Gambar 4.16 Tangkapan Layar Tabel Jadwal Mata Kuliah Daring

Tabel mata kuliah daring yang dihasilkan melalui algoritma genetika sebelumnya akan ditampilkan pada halaman “*Dashboard*”. Proses *generate* akan



tercatat berdasarkan periode urutan penjadwalan yang dilakukan menggunakan metode algoritma genetika.

## b. Pengujian

Pengujian adalah proses pengecekan terhadap program yang telah dibuat. Pengujian ini dilakukan berulang kali untuk mencapai solusi optimal yang diinginkan. Berdasarkan langkah-langkah penjadwalan mata kuliah daring di atas, dapat dilihat bahwa, hasil *generate* akan ditentukan oleh argumen-argumen yang diberlakukan pada setiap kali melakukan *generate* jadwal. Terdapat 2 kasus hasil yang didapat dengan memodifikasi atau menyesuaikan argumen-arguman yang ada.

### 1. Hasil optimal ditemukan

Pada pengujian yang dilakukan dengan pengisian argumen yang tepat, akan dihasilkan solusi optimal. Berikut gambaran argumen yang dimasukkan secara tepat.

Generasi Algoritma Genetika: Progress Generasi

Jumlah Populasi	100	Probabilitas Crossover	0.7
Banyak Titik Crossover	3	Probabilitas Mutasi	0.002

Jumlah Generasi Baru:

Gambar 4.17 Argumen Hasil Optimal

Berdasarkan ilustrasi Gambar 4.17 di atas, terdapat beberapa faktor yang mempengaruhi hasil yang optimal. Faktor-faktor tersebut sebagai berikut.

- (a). Jumlah populasi yang besar. Jumlah populasi awal yang dibangkitkan akan berpengaruh terhadap perolehan solusi optimal. Semakin besar jumlah populasi yang dibangkitkan, akan menghasilkan keberagaman kromosom yang diciptakan sehingga akan menghasilkan calon solusi yang lebih banyak dan beragam pula.
- (b). Jumlah titik *crossover*. Selain jumlah populasi yang besar, titik persilangan juga mempengaruhi nilai *fitness* generasi baru yang akan diciptakan. semakin sedikit titik *crossover*, maka keberagaman generasi baru juga akan semakin kecil. Karena kromosom baru akan memiliki sifat yang hampir menyerupai induknya. Akan tetapi, hal ini tidak menutup kemungkinan menghasilkan solusi optimal,

akan tetapi akan dibutuhkan lebih banyak generasi dan memperpanjang proses yang akan dilakukan oleh sistem.

- (c). Probabilitas *crossover*. Rasio induk yang akan dipilih dalam proses *crossover* juga berpengaruh pada pembentukan generasi baru yang beragam. Probabilitas *crossover* yang semakin tinggi akan menghasilkan kromosom induk yang semakin banyak untuk dilakukan *crossover*, sehingga dianjurkan untuk menggunakan argumen probabilitas *crossover* yang tinggi pula.
- (d). Probabilitas mutasi. Operator mutasi merupakan operator sekunder dalam algoritma genetika sehingga probabilitas mutasi disarankan bernilai kecil. Besar dan kecilnya probabilitas mutasi dapat dianalisa berdasarkan total gen yang ada pada satu generasi. Hal ini dilakukan karena akan dibangkitkan nilai acak pada setiap gen secara merata dengan *range* (0-1) sehingga gen yang mendapat nilai acak lebih kecil dari probabilitas mutasi yang akan mengalami mutasi. Dengan probabilitas yang kecil, diharapkan tidak banyak gen yang mengalami mutasi. Karena juga probabilitas mutasi bernilai besar, maka akan banyak gen yang akan bermutasi, dan kemungkinan akan menghasilkan kembali kromosom dengan *fitness* besar, yang artinya akan semakin tidak *fit*.

## 2. Hasil optimal tidak ditemukan

Dalam pengimplementasian algoritma genetika pada penyelesaian kasus optimasi penjadwalan mata kuliah daring ini, ditemukan bahwa generasi terbaik belum pasti dihasilkan dengan sekali proses *generate*. Penetapan argumen-argumen kontrol sangat berpengaruh selama proses *generate*. Berikut merupakan contoh hasil generasi yang tidak menemukan solusi optimal.

Generasi Algoritma Genetika: Generasi Terakhir

[INDEX: **k500.1**] [FITNESS: **1450**] [PREVIOUS INDEX: k499.9]

ML									
[A] FERI		[B] RAHADIAN		[C] A.RAFIE		[D] A.LUTHFI		[E] KHOLID	
Kamis 12:30 15:00	Rabu 09:30 12:00	Jumat 13:50 16:20	Kamis 08:40 11:10	Rabu 12:30 15:00	Kamis 07:00 09:30	Kamis 09:30 12:00	Rabu 12:30 15:00	Senin 12:30 15:00	Kamis 09:30 12:00
1	2	3	4	5	6	7	8	9	10

Gambar 4.18 Generasi dengan Nilai *Fitness* Buruk

Temuan ini merupakan bentuk penetapan argumen-argumen algoritma genetika yang kurang tepat nilainya sehingga jika dilihat pada Gambar 4.18, argumen-argumen yang ditetapkan untuk menghasilkan generasi ke-500 belum mendekati solusi optimal. Hal ini didukung juga oleh individu-individu yang telah tercipta pada generasi ke-500 ini sudah cenderung identik. Jadi, jika terus dilakukan iterasi sebanyak apapun, maka tidak akan menghasilkan generasi-generasi yang optimal.

#### 4.1.2 Penjadwalan Mata Kuliah Pengganti

Penjadwalan mata kuliah pengganti mereferensi pada jadwal yang sudah berlaku selama satu periode semester tertentu. Jadwal yang berlaku selama satu semester tidak dapat berubah sewaktu-waktu, kecuali dosen tertentu yang mengajar kelas mata kuliah tertentu berhalangan hadir pada jadwal yang sebenarnya.

Oleh karena itu, sistem yang dibangun ini akan mengakomodasi jadwal kuliah yang dinamis seperti kasus di atas. Pengganti jadwal kelas mata kuliah juga dibatasi pada minggu ketika dosen terkait tidak bisa mengajar pada jadwal yang sudah ada. Penjadwalan kelas pengganti juga tidak mengganggu jadwal kelas mata kuliah yang lain sehingga penjadwalan ulang kelas pengganti akan menyeleksi waktu kosong dosen terkait saja. Selain itu, faktor kehadiran mahasiswa juga berpengaruh, karena jadwal baru yang akan dibuat tentu saja memiliki kemungkinan diadakan di waktu yang bersamaan dengan kelas mata kuliah yang lain. Hal ini memungkinkan adanya mahasiswa yang akan memiliki jadwal yang sama pada kelas mata kuliah yang berbeda.

Dengan kondisi seperti di atas, penjadwalan kelas mata kuliah pengganti harus mempertimbangkan jadwal kosong dosen dan persentasi kehadiran mahasiswa pada jadwal baru yang akan menggantikan jadwal tetap. Penjadwalan waktu pengganti ini berlaku hanya pada minggu ketika dosen mengajukan untuk tidak bisa hadir pada jadwal tetap. Jadwal mata kuliah untuk minggu selanjutnya akan berlangsung sesuai jadwal tetap selama semester terkait.

Langkah-langkah penjadwalan di bawah ini menggunakan data *dummy* sebagai pengganti informasi pengambilan mata kuliah oleh setiap mahasiswa. Data ini diperlukan sebagai parameter penjadwalan kelas mata kuliah pengganti sehingga dapat ditentukan jadwal yang akan dipilih dengan tingkat kehadiran mahasiswa tertinggi.

Jadwal yang digunakan adalah salah satu jadwal hasil *generate* pada proses penjadwalan mata kuliah pengganti sebelumnya. Langkah awal untuk mengakses sistem penjadwalan mata

kuliah pengganti sama dengan penjadwalan mata kuliah daring. Berikut tahapan *generate* jadwal mata kuliah pengganti.

a. Mengambil data jadwal tetap semester

Generate Jadwal Kuliah Pengganti

Home / Generate / Jadwal Kuliah Pengganti / Generate Jadwal

Generate Jadwal Kuliah Pengganti: Tahun 2022 Semester genap

[Input Jadwal Mahasiswa](#)

Cari:

Tampilan  entri

Kode ↑↓	Mata Kuliah	SKS ↑↓	Jenis ↑↓	Jumlah Kelas	Dosen Pengampu
SIF201	Matematika Lanjut	6	reguler	5	KELAS A : Feri Wijayanto, M.T. KELAS B : Rahadian Kurniawan, S.Kom., M.Kom. KELAS C : Ahmad M. Raf'ie Pratama, S.T., MIT., Ph.D. KELAS D : Ahmad Luthfi, S.Kom., M.Kom. KELAS E : Kholid Haryono, S.T., M.Kom.
SIF202	Fundamen Pengembangan Aplikasi	6	reguler	5	KELAS A : Kurniawan Dwi Irianto, S.T., M.Sc. KELAS B : Rahadian Kurniawan, S.Kom., M.Kom. KELAS C : Dr. Sri Kusumadewi, S.Si., M.T. KELAS D : Hanson Prihantoro Putro, M.T. KELAS E : Lizda Iswari, S.T., M.Sc.

Gambar 4.19 Jadwal Tetap Mata Kuliah

Ilustrasi Gambar 4.18 di atas merupakan jadwal mata kuliah tetap yang berlaku selama satu semester. Jadwal ini diambil dari hasil optimal penjadwalan mata kuliah daring. Setelah itu dapat menekan tombol “Input Jadwal Mahasiswa”. Langkah ini dilakukan sebagai implementasi dari data *dummy* mahasiswa.

b. Mulai *Generate*

## Generate Jadwal Kuliah Pengganti: Argumen Algoritma Genetika

Home / Generate / Jadwal Kuliah Pengganti / Generate Jadwal / Argumen

### Argumen Algoritma Genetika

<b>Jumlah Populasi</b>	<input type="text" value="100"/>	<b>Probabilitas Crossover</b>	<input type="text" value="0.5"/> <small>Probabilitas harus antara 0 s/d 1</small>
<b>Banyak Titik Crossover</b>	<input type="text" value="1"/>	<b>Probabilitas Mutasi</b>	<input type="text" value="0.05"/> <small>Probabilitas harus antara 0 s/d 1</small>




[+ Tambah Kelas](#)

[Mulai Generasi Pertama](#)

Gambar 4.20 Tangkapan Layar *Input* Argumen

Sama halnya dengan *input* argumen pada penjadwalan mata kuliah daring, penjadwalan mata kuliah pengganti juga membutuhkan argumen-argumen seperti di atas.

- c. *Input* kelas yang akan diganti jadwalnya

Mata Kuliah	<input type="text"/>
Kelas 	<input type="text"/>
Mata Kuliah	<input type="text"/>
Kelas 	<input type="text"/>
Mata Kuliah	<input type="text"/>
Kelas 	<input type="text"/>

[+ Tambah Kelas](#)

[Mulai Generasi Pertama](#)

Gambar 4.21 *Input* Kelas Mata Kuliah

Proses ini sebagai bentuk dari pengganti jadwal yang akan dilakukan bagi tepat seorang dosen atau lebih. Tersedia pilihan mata kuliah yang akan diganti beserta kelas dosen yang akan diganti. Setelah itu, akan dilakukan proses pembangkitan populasi awal dengan panjang kromosom yang direpresentasikan oleh gen-gen kelas yang akan diganti jadwalnya.

d. *Input jumlah generasi*

The screenshot shows a web interface for inputting the number of generations. At the top, there is a text input field labeled 'Jumlah Generasi Baru' with a dropdown arrow, and a green button labeled 'Tambah Generasi'. Below this, there are two summary tables for the genetic algorithm process.

The first table is titled 'Generasi Algoritma Genetika: Best Kromosom' and has two columns: 'Generasi' and 'Fitness'. The value for 'Generasi' is 0 and for 'Fitness' is 0.

The second table is titled 'Generasi Algoritma Genetika: Generasi Terakhir' and shows the results for the final generation. It includes a status bar: '[INDEX: k1] [FITNESS: 0] Lihat Hasil'. The table has four columns representing different classes and their sessions, with sub-columns for 'ORIGINAL' and 'BARU' (New) schedules.

EP ERIKA KELAS B SESI 1		P-PCD YUDI KELAS B SESI 1		P-DL IZZATI KELAS A SESI 1		Metopel BENI KELAS B SESI 1	
Senin 12:30 14:10	Kamis 07:50 09:30	Rabu 08:40 11:10	Senin 09:30 12:00	Senin 12:30 15:00	Jumat 07:35 10:05	Senin 08:40 11:10	Rabu 08:40 11:10
ORIGINAL	BARU	ORIGINAL	BARU	ORIGINAL	BARU	ORIGINAL	BARU
1		2		3		4	

Gambar 4.22 *Input Jumlah Generasi*

Jumlah generasi yang dibutuhkan tidak terlalu besar, hal ini dikarenakan sebuah kromosom hanya memiliki beberapa gen saja. Keberagaman yang akan terbentuk juga didapat dengan jumlah populasi yang tidak terlalu besar.

```
PS C:\Users\Jarkumande\Documents\Lama kali TA nya\nih TA nya> php artisan queue:work --queue=jadwal_kuliah_pengganti --timeout=7200
[2021-11-29 05:56:11][68] Processing: App\Jobs\ProcessGenerateJKP
[2021-11-29 05:56:11][68] Processed: App\Jobs\ProcessGenerateJKP
```

Gambar 4.23 *Proses Generate Jadwal Pengganti*

e. *Menampilkan hasil generate*

Hasil *generate* jadwal mata kuliah pengganti akan ditampilkan pada halaman *Dashboard*. Dapat dilihat pada ilustrasi di bawah ini.

Hasil Generate Jadwal Kuliah Pengganti

ID: 3 ID: 4

ID: 3

No.	Mata Kuliah	Dosen	Kelas	Sesi	Jadwal Lama	Jadwal Baru
1	Algoritma dan Struktur Data	Ahmad M. Raf'ie Pratama, S.T., MIT., Ph.D.	A	2	Senin 15:30 - 17:10	Rabu 15:30 - 17:10
2	Kapita Selektta Informatika Medis	Arrie Kurniawardhani, S.Si., M.Kom.	A	1	Rabu 12:30 - 15:00	Jumat 09:15 - 11:45
3	Komputasi Lunak	Ahmad Luthfi, S.Kom., M.Kom.	B	1	Kamis 07:00 - 09:30	Jumat 13:00 - 15:30
4	Gim Serious	Affan Mahtarami, S.Kom., M.T.	B	1	Kamis 12:30 - 15:00	Senin 07:50 - 10:20
5	Arsitektur Perangkat Lunak	Taufiq Hidayat, S.T., MCS.	B	1	Jumat 06:45 - 09:15	Rabu 12:30 - 15:00
6	Metodologi Penelitian	Beni Suranto, M.SoftEng.	B	1	Senin 08:40 - 11:10	Rabu 08:40 - 11:10

Gambar 4.24 Tangkapan Layar Jadwal Pengganti Mata Kuliah

Jadwal ini dapat digunakan hanya pada minggu terkait. Oleh karena itu, setiap minggu akan dilakukan proses *generate* jadwal pengganti mata kuliah juga terdapat minimal seorang dosen dengan kelas mata kuliah tertentu ingin mengganti jadwalnya.

#### 4.1.3 Analisis Kelebihan dan Kekurangan Sistem

Dalam pengembangan sistem penjadwalan mata kuliah daring ini, terdapat kelebihan dan kekurangan sistem. Berikut adalah kelebihan dan kekurangan sistem.

##### a. Kelebihan Sistem

1. Sistem penjadwalan mata kuliah daring ini dapat membantu Jurusan Informatika UII dalam menyusun jadwal mata kuliah daring beserta penjadwalan kelas pengganti mata kuliah sebagai bentuk jadwal mata kuliah yang dinamis.
2. Memberikan lebih dari satu pilihan solusi optimal jika sudah menghasilkan generasi terbaik.



3. Memberikan informasi pelanggaran yang terjadi pada setiap kromosom yang tercipta.
  4. Dengan penetapan argumen-argumen kontrol yang tepat, sistem tidak membutuhkan banyak generasi untuk mendapatkan hasil optimal.
- b. Kekurangan Sistem
1. Untuk mencapai solusi optimal dibutuhkan penetapan nilai argumen yang tepat.
  2. Pengimplementasian program menggunakan bahasa pemrograman PHP menyebabkan proses untuk mencapai solusi optimal membutuhkan waktu yang lama dan membutuhkan alokasi memori dan penyimpanan yang cukup besar.
  3. Sistem yang dibangun fokus pada penyusunan jadwal mata kuliah daring saja, tidak terintegrasi dengan sistem-sistem perkuliahan lainnya.
  4. Sistem tidak selalu menghasilkan solusi optimal tiap kali melakukan *generate* jadwal mata kuliah daring maupun jadwal pengganti sesi mata kuliah.

#### 4.1.4 Evaluasi Sistem

Dalam penggunaan sistem penjadwalan mata kuliah daring yang sudah dirancang, dilakukan banyak percobaan sistem sehingga dapat memvalidasi sistem ini berjalan sesuai yang diinginkan. Berikut adalah tabel ringkasan percobaan yang telah dilakukan pada sistem penjadwalan mata kuliah daring.

Tabel 4.1 Tabel Ringkasan Percobaan Sistem

No.	Nilai Argumen Kontrol	Generasi yang dihasilkan	Hasil
1.	<ul style="list-style-type: none"> <li>- Jumlah populasi: 10</li> <li>- Jumlah titik <i>crossover</i>: 3</li> <li>- <math>P_c</math>: 0,7</li> <li>- <math>P_m</math>: 0,002</li> </ul>	200 generasi	Generasi ke-200 memiliki banyak individu yang identik dengan nilai <i>fitness</i> yang masih tinggi, sedangkan yang diharapkan adalah nilai <i>fitness</i> yang dibutuhkan sebagai solusi optimal adalah nol atau yang mendekati nol. Hal ini dipengaruhi oleh keberagaman individu yang dibangkitkan pada populasi awal hanya berjumlah 10, sedangkan panjang kromosom 105 gen

			dan memiliki nilai <i>fitness</i> yang sangat besar. Hal ini menyebabkan <i>offspring</i> yang dihasilkan sulit untuk lepas dari sifat induknya. Jika iterasi terus dilakukan, maka tidak akan adanya perbaikan individu pada generasi-generasi selanjutnya.
2.	<ul style="list-style-type: none"> <li>- Jumlah populasi: 100</li> <li>- Jumlah titik <i>crossover</i>: 3</li> <li>- <math>P_c</math>: 0,4</li> <li>- <math>P_m</math>: 0,002</li> </ul>	500 generasi	Pada generasi ke-300an, perbaikan nilai <i>fitness</i> terlihat berjalan dengan baik, akan tetapi setelah itu, individu yang dihasilkan perlahan mulai identik dengan nilai <i>fitness</i> yang cukup kecil. Akan tetapi, belum bisa dijadikan sebagai calon solusi optimal. Hal ini disebabkan oleh nilai argumen kontrol $P_c$ yang terlalu kecil. Pemilihan kromosom induk yang akan dilakukan <i>crossover</i> dengan probabilitas kecil terhadap jumlah populasi akan menghasilkan peluang tercipta <i>offspring offspring</i> dengan tingkat keberagaman yang rendah sehingga butuh lebih banyak generasi untuk mencapai solusi optimal dan kemungkinan menemukan solusi optimal juga masih rendah.
3.	<ul style="list-style-type: none"> <li>- Jumlah populasi: 100</li> <li>- Jumlah titik <i>crossover</i>: 3</li> <li>- <math>P_c</math>: 0,6</li> <li>- <math>P_m</math>: 0,3</li> </ul>	1000 generasi	Penetapan nilai probabilitas mutasi yang cukup besar memiliki kekurangan pada sistem. Pada dasarnya, proses mutasi diharapkan menciptakan kromosom baru yang belum pernah ada pada generasi-generasi sebelumnya. Tetapi dengan probabilitas mutasi yang

			<p>besar, maka akan semakin banyak mutan tercipta. Hal ini dapat menyebabkan hilangnya sifat induk yang ditandai dengan nilai <i>fitness</i> yang tidak membaik, tetapi semakin buruk. Pada generasi-generasi awal, generasi baru yang dihasilkan menunjukkan perbaikan nilai <i>fitness</i> yang cukup baik. Akan tetapi, setelah mencapai generasi ke-600, nilai <i>fitness</i> terbaik setiap generasi terus memburuk hingga terjadi ketidakstabilan perbaikan nilai <i>fitness</i>. Hal ini menyebabkan pemrosesan yang semakin lama untuk mencapai solusi optimal dan menciptakan kromosom-kromosom identik dengan nilai <i>fitness</i> yang masih jauh dari yang diharapkan.</p>
4.	<ul style="list-style-type: none"> <li>- Jumlah populasi: 100</li> <li>- Jumlah titik <i>crossover</i>: 3</li> <li>- <math>P_c</math>: 0,7</li> <li>- <math>P_m</math>: 0,002</li> </ul>	650 generasi	<p>Penetapan nilai argumen kontrol yang mengarah pada hasil optimal. Jumlah populasi yang besar akan menghasilkan keberagaman yang tinggi sehingga nilai <i>fitness</i> yang dimiliki setiap kromosom juga bervariasi. Jumlah titik <i>crossover</i> belum dilakukan perubahan, akan tetapi dengan jumlah titik <i>crossover</i> sebanyak 3 buah masih memenuhi hasil optimal yang diinginkan. Nilai probabilitas <i>crossover</i> yang cukup besar akan memungkinkan menciptakan keberagaman <i>offspring</i> juga. Nilai probabilitas mutasi yang</p>

			<p>kecil dengan pertimbangan panjang kromosom dan jumlah populasi awal yang dibangkitkan akan memungkinkan terciptanya mutan semakin kecil pula. Pada percobaan kali ini, terdapat 100 populasi awal dan panjang masing-masing generasi adalah 105 gen sehingga terdapat 10.500 gen pada setiap generasi. Dengan demikian, nilai <math>P_m</math> sebesar 0,002 memungkinkan hanya akan menghasilkan kira-kira 21 mutasi gen dengan pertimbangan nilai acak (antara 0-1) yang dibangkitkan pada setiap gen. dengan nilai setiap argumen kontrol yang sesuai seperti yang telah diberikan, generasi yang dihasilkan secara stabil menunjukkan perbaikan hingga mencapai generasi ke-650. Pada generasi ini, sudah memiliki persebaran nilai <i>fitness</i> yang tidak lebar dan individu-individu yang identik dengan nilai <i>fitness</i> yang sudah mendekati nol. Pada generasi ini, <i>user</i> sudah dapat menggunakan individu terbaik sebagai solusi optimal dengan memilih salah satu dari beberapa kromosom yang memiliki nilai <i>fitness</i> yang sama dan sudah mendekati nol.</p>
5.	<ul style="list-style-type: none"> <li>- Jumlah populasi: 100</li> <li>- Jumlah titik <i>crossover</i>: 1</li> <li>- <math>P_c</math>: 0,7</li> <li>- <math>P_m</math>: 0,002</li> </ul>	450 generasi	Tidak ada perubahan yang mencolok jika jumlah titik <i>crossover</i> adalah 1 dengan argumen kontrol yang lain memiliki nilai yang sama pada

			percobaan yang menghasilkan solusi optimal sebelumnya.
6.	<ul style="list-style-type: none"> <li>- Jumlah populasi: 100</li> <li>- Jumlah titik <i>crossover</i>: 7</li> <li>- <math>P_c</math>: 0,7</li> <li>- <math>P_m</math>: 0,002</li> </ul>	300 generasi	Tidak ada perubahan yang signifikan dengan menambahkan titik <i>crossover</i> . Hanya saja jumlah iterasi yang dilakukan tidak sebanyak percobaan sebelumnya yang menerapkan jumlah titik <i>crossover</i> 1 maupun 3 titik. Hal ini menunjukkan bahwa. Akan tetapi, percobaan yang dilakukan menggunakan argumen kontrol yang sama bisa menghasilkan solusi optimal pada generasi yang berbeda. Tidak terlihat konsistensi pada peningkatan jumlah titik <i>crossover</i> akan menghasilkan generasi yang selalu lebih sedikit dibandingkan menggunakan jumlah titik <i>crossover</i> yang sedikit.

Berdasarkan Tabel 4.1 di atas, penggunaan nilai argumen kontrol yang tepat akan memungkinkan solusi optimal ditemukan. Akan tetapi, dengan menggunakan nilai argumen kontrol yang sudah memenuhi syarat ditemukannya hasil optimal tidak menjamin semua percobaan akan berjalan sesuai keinginan. Walaupun menggunakan argumen kontrol yang tepat, kondisi yang dapat terjadi adalah jumlah generasi yang harus dihasilkan tidak selalu sama sehingga waktu pemrosesan juga dapat berubah-ubah tergantung iterasi yang menghasilkan generasi terbaik. Hal ini disebabkan oleh algoritma genetika merupakan algoritma pencarian acak sehingga hasil yang didapatkan pada setiap kali sistem yang menggunakan algoritma genetika selalu berubah-ubah. Akan tetapi, dengan menggunakan nilai argumen kontrol yang tepat, perubahan yang terjadi tidak terlalu mencolok.

## BAB V

### KESIMPULAN DAN SARAN

#### 5.1 Simpulan

Dari semua persiapan, perancangan, hingga implementasi dan pengujian program yang dibuat untuk melakukan penjadwalan mata kuliah daring dan penjadwalana kelas pengganti mata kuliah di atas, dapat disimpulkan bahwa:

- a. Pembangkitan populasi awal sangat berpengaruh terhadap solusi optimal yang ingin diperoleh. Keberagaman individu atau kromosom akan menghasilkan proses yang lebih singkat dalam menerapkan algoritma genetika ke dalam permasalahan optimasi ini.
- b. Hasil optimal didapat bukan berdasarkan dari iterasi yang ditentukan di awal, akan tetapi menganalisa perkembangan masing-masing generasi secara bertahap.
- c. Generasi terbaru, baik dengan argumen yang tepat maupun tidak, akan selalu mengarah kepada kromosom-kromosom yang identik yang dilihat dari nilai *fitness* masing-masingnya. Hal ini mengakibatkan apabila nilai *fitness* yang tak kunjung membaik setelah menciptakan banyak generasi maka proses *generate* harus diulang dengan kecocokan argumen yang dipilih.
- d. Sistem pembuatan jadwal dengan algoritma genetika ini dapat menangani operasi olah data, melakukan pembuatan jadwal mata kuliah daring dan menampilkannya dalam bentuk tabel yang hanya bisa diakses di dalam sistem. Belum terdapat modul mengambil hasilnya ke dalam bentuk \*.xlsx ataupun bentuk *file* sejenis.
- e. Jadwal mata kuliah daring yang dihasilkan merupakan representasi dari sebuah kromosom yang kualitasnya diukur dari nilai *fitness*.
- f. Meskipun sudah dirancang sedemikian fleksibel, namun sistem ini masih memiliki kekurangan dari berbagai sisi sehingga diperlukan penyempurnaan di dalamnya.

#### 5.2 Saran

Dari hasil pengujian perangkat lunak yang telah dibuat, terdapat beberapa kekurangan dalam sistem. Untuk itu perlu dikembangkan lebih lanjut agar tercipta sistem yang lebih baik. Saran pengembangan aplikasi ini adalah:

- a. Diharapkan ke depannya, sistem dapat dikembangkan lebih luas lagi dengan menyesuaikan kasus-kasus penjadwalan yang berbeda-beda.
- b. Penjadwalan yang dilakukan dapat memfasilitasi ketersediaan waktu mengajar yang dimiliki oleh setiap dosen.
- c. Sistem dapat memperluas jadwal yang digunakan, di antaranya dengan mengikutsertakan jadwal praktikum dan jadwal mata kuliah universitas.
- d. Sistem dikembangkan ke dalam aplikasi *desktop* untuk meminimalkan waktu pemrosesan.
- e. Sistem sudah menangani nilai argumen-argumen yang tepat, sehingga pengguna awam dapat mengoperasikan sistem tanpa harus mengerti algoritma genetika yang bekerja di dalamnya.





**DAFTAR PUSTAKA**

- Kusumadewi, S. &. (2005). *Penyelesaian Masalah optimasi dengan Teknik-teknok Heuristik*. Yogyakarta: Graha Ilmu.
- Ladjamudin, A. B. (2006). *Rekayasa Perangkat Lunak*. Yogyakarta: Graha Ilmu.
- McConnell, S. (2004). *Code Complete*. Redmond: Microsoft Press.
- Mitsuo Gen, Runwei Cheng. (2000). *Genetic Algorithm and Engineering Optimization*. New York: John Willey & Sons, Inc.
- Abdulkarim, H. A., & Alshammari, I. F. (2015). Comparison of Algorithms for Solving Traveling Salesman Problem. *International Journal of Engineering and Advanced Technology*
- Albar, M. A. (2013). Algoritma Genetik Tabu Search Dan Memetika Pada Permasalahan Penjadwalan Kuliah. *Semnasteknomedia Online*
- Amri, F., Nababan, E. B., & Syahputra, M. F. (2012). Artificial Bee Colony Algorithm untuk Menyelesaikan Travelling Salesman Problem. *Jurnal Dunia Teknologi Informasi*
- Andri, Suyandi, & WinWin. (2013). Aplikasi Travelling Salesman Problem dengan Metode Artificial Bee Colony.
- Aristi, G. (2014). Perbandingan Algoritma Greedy, Algoritma Cheapest Insertion Heuristics Dan Dynamic Programming Dalam Penyelesaian Travelling Salesman Problem. *Paradigma*, XVI(2)
- Darmawan, I., & Siliwangi, U. (2011). Hibridisasi Genetic-Tabu Search Algorithm Untuk Penjadwalan. 2011(Snati), 17–18. Dini, N. S. (2015). Produk Air Minum Kemasan Galon Menggunakan Kombinasi Algoritma Genetika Dan Pencarian Tabu Di Depot Air Minum Isi Ulang Banyu Belik.
- Nielsen, J. (1993). *Usability Engineering*. New York: Morgan Kaufmann. Pengenalan Pemrograman Java 1. (2007). Jakarta: J.E.N.I
- Sam'ani. (2012). Rancang Bangun Sistem Penjadwalan Perkuliahan dan Ujian Akhir Semester dengan Pendekatan Algoritma Genetika.

LAMPIRAN

