

**IMPLEMENTASI ALGORITMA GENETIK
UNTUK SISTEM PENJADWALAN PERKULIAHAN OTOMATIS
BERBASIS WEB DI JURUSAN ILMU KOMPUTER UNIVERSITAS
LAMPUNG**

(Skripsi)

Oleh

**CINDY LORIA
NPM 2117051004**



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2025**

**IMPLEMENTASI ALGORITMA GENETIK
UNTUK SISTEM PENJADWALAN PERKULIAHAN OTOMATIS
BERBASIS WEB DI JURUSAN ILMU KOMPUTER UNIVERSITAS
LAMPUNG**

Oleh

CINDY LORIA

Skripsi

**Sebagai Salah Satu Syarat untuk Memperoleh Gelar
SARJANA KOMPUTER**

Pada

**Program Studi S1 Ilmu Komputer
Jurusani Ilmu Komputer**



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2025**

ABSTRAK

IMPLEMENTASI ALGORITMA GENETIK UNTUK SISTEM PENJADWALAN PERKULIAHAN OTOMATIS BERBASIS WEB DI JURUSAN ILMU KOMPUTER UNIVERSITAS LAMPUNG

Oleh

CINDY LORIA

Penjadwalan perkuliahan di perguruan tinggi merupakan proses kompleks yang melibatkan pengaturan waktu, ruang, dan dosen dengan berbagai kendala. Di Jurusan Ilmu Komputer Universitas Lampung, penjadwalan manual seringkali menyebabkan konflik dan ketidakefisienan penggunaan sumber daya. Penelitian ini bertujuan untuk menerapkan Algoritma Genetika dalam penjadwalan otomatis menggunakan Python, dan mengembangkan *web service* berbasis FastAPI untuk mengintegrasikan model optimasi tersebut. Metode penelitian mencakup pengumpulan data, pemodelan data, implementasi Algoritma Genetika menggunakan *library* PYGAD, serta pengujian sistem. Hasil menunjukkan bahwa Algoritma Genetika mampu menghasilkan jadwal optimal dengan waktu komputasi rata-rata di bawah 3 menit (180 detik) untuk satu periode akademik, serta memvalidasi semua constraints. *Web service* yang dikembangkan juga berhasil beroperasi dengan baik, memungkinkan integrasi antarmuka pengguna berbasis web. Kesimpulannya, solusi ini efektif dalam menyelesaikan masalah penjadwalan secara efisien dan dapat diadopsi untuk mendukung operasional akademik jurusan.

Kata kunci: Penjadwalan perkuliahan, Algoritma Genetika, FastAPI, optimasi, *web service*.

ABSTRACT

IMPLEMENTASI ALGORITMA GENETIK UNTUK SISTEM PENJADWALAN PERKULIAHAN OTOMATIS BERBASIS WEB DI JURUSAN ILMU KOMPUTER UNIVERSITAS LAMPUNG

By

CINDY LORIA

University course scheduling is a complex process involving the coordination of time, space, and lecturers with various constraints. At the Computer Science Department of Lampung University, manual scheduling frequently leads to conflicts and inefficient resource allocation. This study aims to implement a Genetic Algorithm for automatic scheduling using Python, and develop a FastAPI-based web service to integrate the optimization model. The research methodology includes data collection, data modeling, implementation of Genetic Algorithm using PYGAD library, and system testing. Results demonstrate that the Genetic Algorithm can generate optimal schedules with an average computation time of under 3 minutes (180 seconds) for one academic period while satisfying all constraints. The developed web service also functions effectively, enabling integration with web-based user interfaces. In conclusion, this solution proves efficient in resolving scheduling problems and can be adopted to support academic operations in the department.

Keywords: Course scheduling, Genetic Algorithm, FastAPI, optimization, web service.

Judul Skripsi

: IMPLEMENTASI ALGORITMA GENETIK
UNTUK SISTEM PENJADWALAN
PERKULIAHAN OTOMATIS BERBASIS
WEB DI JURUSAN ILMU KOMPUTER
UNIVERSITAS LAMPUNG

Nama Mahasiswa

: Cindy Toria

Nomor Pokok Mahasiswa

: 2117051004

Program Studi

: Ilmu Komputer

Fakultas

: Matematika dan Ilmu Pengetahuan Alam



Febi Eka Febriansyah, S.T., M.T.
NIP. 19800219 2006041 001

2. MENGETAHUI

Ketua Jurusan Ilmu Komputer
FMIPA Universitas Lampung

A blue ink signature of the name "Dwi Sakethi".

Dwi Sakethi, S.Si., M.Kom
NIP. 19680611 199802 1 001

Ketua Program Studi Ilmu Komputer
FMIPA Universitas Lampung

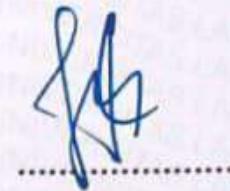
A blue ink signature of the name "Tristiyanto".

Tristiyanto, S.Kom., M.I.S., Ph.D.
NIP. 19810414 200501 1 001

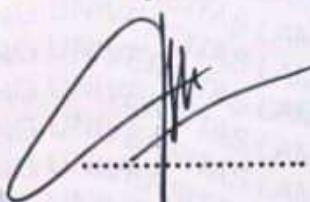
MENGESAHKAN

1. Tim Pengaji

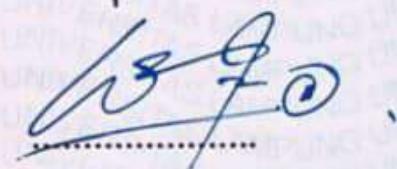
Ketua : Febi Eka Febriansyah, S.T., M.T.



Pengaji I : Didik Kurniawan, S.Si., M.T.



Pengaji II : Wartariyus, S.Kom., M.T.I



2. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam



Dr. Eng. Heri Satria, S.Si., M.Si.

NIP. 197110012005 01 1 002

Tanggal Lulus Ujian Skripsi: **2 Juli 2025**

PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Cindy Loria

NPM : 2117051004

Menyatakan bahwa skripsi saya yang berjudul '**Implementasi Algoritma Genetik Untuk Sistem Penjadwalan Perkuliahan Otomatis Berbasis Web Di Jurusan Ilmu Komputer Universitas Lampung**' merupakan karya saya sendiri dan bukan merupakan karya orang lain. Semua isi tulisan yang tertuang di skripsi ini telah mengikuti kaidah penulisan karya ilmiah Universitas Lampung. Apabila di kemudian hari terbukti skripsi saya merupakan hasil penjiplakan atau dibuat orang lain, maka saya bersedia menerima sanksi berupa pencabutan gelar yang telah saya terima.

Bandar Lampung, 29 Juli 2025



Cindy Loria
NPM. 2117051004

RIWAYAT HIDUP



Penulis dilahirkan di Pekalongan, Lampung Timur pada tanggal 20 Mei 2004 sebagai anak pertama dari 3 bersaudara, dari Bapak Yanto Hendro Lukito dan Ibu Lianty. Pendidikan yang sudah ditempuh oleh penulis diantaranya, menyelesaikan pendidikan dasar di SD Negeri 4 Metro Timur pada tahun 2016.

Penulis menyelesaikan pendidikan menengah pertama di SMP Negeri 1 Metro pada tahun 2019. Kemudian penulis menyelesaikan pendidikan menengah atas di SMA Negeri 1 Metro pada tahun 2021. Untuk jenjang selanjutnya penulis melanjutkan Pendidikan di Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung melalui jalur Seleksi Nilai Masuk Perguruan Tinggi (SNMPTN) pada tahun 2021. Selama menjadi mahasiswa, penulis aktif mengikuti beberapa kegiatan antara lain:

1. Anggota bidang Keilmuan Himakom Universitas Lampung
2. Asisten Dosen mata kuliah Matematika dan Matematika Diskrit.
3. Pada tahun 2024 melakukan Kuliah Kerja Nyata (KKN) di desa Hanura, Kecamatan Teluk Pandan, Kabupaten Pesawaran.
4. Pada tahun 2024 mendapatkan kesempatan untuk mengikuti kegiatan magang melalui program Magang dan Studi Independen Bersertifikat (MSIB) di Dinas Komunikasi, Informasi, Statistik, dan Persandian di Kabupaten Pesawaran sebagai *Front-end Developer*.

MOTTO

"Janganlah takut, sebab Aku menyertai engkau, janganlah bimbang, sebab Aku ini Allahmu; Aku akan meneguhkan, bahkan akan menolong engkau; Aku akan memegang engkau dengan tangan kanan-Ku yang membawa kemenangan."

(Yesaya 41:10)

"Karena masa depan sungguh ada, dan harapanmu tidak akan hilang"

(Amsal 23:18)

PERSEMBAHAN

Puji Syukur kepada Tuhan Yesus Kristus atas berkat kasih dan anugerahnya
sehingga skripsi ini dapat terselesaikan.

Kupersembahkan karya ini kepada:

Kedua Orang Tuaku dan Adik-adikku Tercinta

Yang senantiasa memberikan segala yang terbaik yang selalu mendoakan,
memberikan pesan, motivasi, dukungan serta kasih sayang hingga saat ini.

Seluruh Keluarga Besar Ilmu Komputer 2021

Yang selalu membantu dan memberikan dukungan.

SANWACANA

Puji Syukur kepada Tuhan Yang Maha Esa, yang telah melimpahkan berkat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul “Implementasi Algoritma Genetik Untuk Sistem Penjadwalan Perkuliahan Otomatis Berbasis Web Di Jurusan Ilmu Komputer Universitas Lampung”. Pada kesempatan ini penulis mengucapkan Terima kasih kepada semua pihak yang telah membantu dalam penyusunan skripsi ini, yaitu:

1. Tuhan Yesus Kristus yang menjadi sumber kekuatan, sukacita dan pengharapan yang selalu memberikan penyertaan dan pemeliharaan selama penulis menyelesaikan skripsi.
2. Kedua orang tua, bapak Yanto Hendro Lukito dan ibu Lianty serta adik-adik, Michelle Natalie Loisa dan Caroline Christiani Lorista serta keluarga besar yang telah memberikan dukungan dan motivasi kepada penulis untuk menyelesaikan skripsi dengan baik.
3. Bapak Dr. Eng. Heri Satria, S.Si., M.Si. selaku Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam.
4. Bapak Dwi Sakethi,S.Si., M.Kom selaku Ketua Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung.
5. Bapak Tristiyanto, S.Kom., M.I.S., Ph.D. selaku Ketua Program Studi Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung.
6. Ibu Yunda Heningtyas, M.Kom selaku Sekretaris Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung.
7. Bapak M. Iqbal Parabi, S.Si., M.T selaku Dosen Pembimbing Akademik.

8. Bapak Febi Eka Febriansyah, S.T., M.T. selaku Dosen Pembimbing utama atas ketersediannya dalam memberikan bimbingan, motivasi, kritik serta saran dalam proses penyelesaian skripsi.
9. Bapak Didik Kurniawan, S.Si., M.T. selaku Dosen Pembahas 1 yang telah memberikan kritik dan saran hingga masukan kepada penulis selama proses penyelesaian skripsi.
10. Bapak Wartariyus, S.Kom., M.T.I selaku Dosen Pembahas 2 yang telah memberikan kritik dan saran hingga masukan kepada penulis selama proses penyelesaian skripsi.
11. Ibu Anie Rose Irawati, ST., M.Cs yang telah memberikan bimbingan dan petunjuk berharga selama penulis menjalani masa perkuliahan.
12. Bapak dan Ibu Dosen Jurusan Ilmu Komputer Universitas Lampung yang telah memberikan ilmu, pengetahuan serta pengalamannya kepada penulis selama penulis menjadi mahasiswa.
13. Seluruh staf Jurusan Ilmu Komputer yang selalu membantu penulis dari awal hingga akhir masa perkuliahan.
14. Jihan Haya Mufialdo dan Iqbal Al Hafidzu Rahman selaku rekan skripsi yang selalu mendukung dan memberikan semangat dalam proses pembuatan skripsi.
15. Teman-teman “Harta Tahta Nilai A” dan “temennya Cahya”, Fathimah Abiyyi Khairunnisa, Siti Ayuni, Shafa Auliya, Jihan Haya Mufialdo, Vidya Sinta Bilkis, Nabillah Aisyah, Roy Rafles Matorang Pasaribu, dan Ikhsan Saputra selaku teman seperjuangan yang selalu membantu dari awal masa perkuliahan, yang memberikan motivasi dan semangat kepada penulis.
16. Yohanes Tegas Prasetyo yang setia menemani dalam suka dan duka selama proses ini terima kasih atas kesabaran, motivasi, semangat dan kepercayaannya.
17. Teman-teman KKN dan Teman-teman MSIB di Diskominfotiksan Pesawaran yang selalu memberikan semangat dan motivasi kepada penulis.

18. Keluarga besar GBI A.H Nasution Metro yang telah memberikan doa, semangat, motivasi dan dukungannya kepada penulis.
19. Teman-teman Ilmu Komputer yang menjadi keluarga besar Jurusan Ilmu Komputer selama menjalankan masa studi di Jurusan Ilmu Komputer Universitas Lampung.

Bandar Lampung, 29 Juli 2025



Cindy Loria
NPM. 2117051004

DAFTAR ISI

DAFTAR TABEL.....	vii
DAFTAR GAMBAR.....	viii
DAFTAR KODE.....	ix
I. PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	3
1.3. Batasan Masalah.....	3
1.4. Tujuan Penelitian.....	4
1.5. Manfaat Penelitian.....	4
II. TINJAUAN PUSTAKA	5
2.1. Penelitian Terdahulu	5
2.1.1. Application of Genetic Algorithm Concept on Course Scheduling.....	6
2.1.2. Genetic Algorithm for Optimization of Lecturer Schedule Preparation	6
2.1.3. Genetic Algorithm for Solving University Timetabling Problem	7
2.2. Uraian Landasan Teori	7
2.2.1. Penjadwalan.....	8
2.2.2. Algoritma Genetika	8
2.2.3. <i>Library PyGAD pada Python</i>	19
2.2.4. Sistem Informasi Akademik (SIAKAD).....	23
2.2.5. <i>Web Service</i>	23
2.2.6. FastAPI	24
III. METODOLOGI PENELITIAN.....	25
3.1. Waktu dan Tempat Penelitian.....	25
3.2. Perangkat Penelitian	25
3.2.1. Perangkat Keras	25

3.2.2. Perangkat Lunak	26
3.3. Tahapan Penelitian	26
3.3.1. Pengumpulan Data.....	27
3.3.2. Pemodelan Data.....	28
3.3.3. Implementasi Algoritma Genetika.....	38
3.3.4. Pengembangan <i>Web Service</i>	44
3.3.5. Pengujian <i>Web Service</i>	44
IV. HASIL DAN PEMBAHASAN	46
4.1. Pemodelan	46
4.1.1. Struktur Data.....	46
4.1.2. Kendala (Constraints) dalam Penjadwalan	46
4.2. Implementasi PyGAD	60
4.2.1. Inisialisasi Algoritma Genetika.....	60
4.2.2. Pemilihan Solusi Terbaik	62
4.3. Pengembangan <i>Web Service</i>	63
4.4. Pengujian.....	64
4.4.1. Pengujian Model.....	64
4.4.2. Pengujian Web service.....	66
V. KESIMPULAN DAN SARAN	68
5.1. Kesimpulan.....	68
5.2. Saran.....	68
DAFTAR PUSTAKA.....	70
LAMPIRAN.....	74

DAFTAR TABEL

Table	Halaman
1. Penelitian Terdahulu.....	5
2. Jumlah Mata Kuliah Semester Ganjil	29
3. Jumlah Mata Kuliah Semester Genap	30
4. Jumlah Dosen	30
5. Jumlah Ruangan	30
6. Jumlah Kelas Prodi	30
7. Jumlah Kelas Kuliah Semester Ganjil.....	31
8. Jumlah Kelas Kuliah Semester Genap	31
9. Tabel Sesi	31
10. Skenario Pengujian Web Service	45
11. Pengujian Constraint	64
12. Hasil Pengujian Waktu Komputasi	65
13. Hasil Pengujian Web Service	66

DAFTAR GAMBAR

Gambar	Halaman
1. Komponen Algoritma Genetik	9
2. Contoh Data yang akan Diolah	10
3. Plotting Jadwal	12
4. Perhitungan Fitness	14
5. Probabilitas Individu	14
6. Interval Roulette	16
7. Hasil Offspring 1	17
8. Hasil Offspring 2	18
9. Source Code Fungsi Fitness	20
10. Source Code Instance Kelas PyGAD	20
11. Source Code Method run()	21
12. Siklus PyGAD	22
13. Tahapan Penelitian	26
14. Roulette Wheel	41

DAFTAR KODE

Kode	Halaman
1. Constraint 1	47
2. Constraint 2	48
3. Constraint 3	49
4. Constraint 4	51
5. Constraint 5	52
6. Constraint 6	53
7. Constraint 7	53
8. Constraint 8	54
9. Constraint 9	56
10. Constraint 10	57
11. Constraint 11	58
12. Constraint 12	59
13. Parameter Algoritma Genetik.....	60
14. Pencarian Solusi Terbaik.....	62

I. PENDAHULUAN

1.1. Latar Belakang

Penjadwalan perkuliahan merupakan salah satu kegiatan administratif yang kompleks di perguruan tinggi, terutama dalam mengatur alokasi waktu, ruang, dan dosen. Setiap semester, perguruan tinggi dihadapkan pada tantangan untuk memastikan bahwa jadwal perkuliahan yang dibuat tidak hanya memenuhi kebutuhan akademik mahasiswa, tetapi juga memperhitungkan keterbatasan sumber daya yang ada, seperti jumlah ruang kuliah yang terbatas, dosen yang memiliki jadwal padat, serta distribusi mata kuliah yang bervariasi. Ketika jumlah mahasiswa dan program studi semakin bertambah, kompleksitas dalam penjadwalan semakin meningkat. Jika penjadwalan dilakukan secara manual, akan sangat memakan waktu dan rawan terjadi kesalahan seperti bentroknya jadwal antara kelas, ruang, atau dosen (Prambudi et al., 2021).

Jurusan Ilmu Komputer Universitas Lampung merupakan salah satu jurusan yang setiap tahunnya menerima banyak mahasiswa baru, sehingga setiap semester, ada puluhan mata kuliah yang harus dijadwalkan. Dengan sistem penjadwalan manual yang selama ini diterapkan, seringkali terjadi konflik penjadwalan yang mengganggu kelancaran perkuliahan. Misalnya, bentroknya jadwal antar mata kuliah di ruangan yang sama, ketidaksesuaian jadwal dosen, atau jadwal yang terlalu padat bagi mahasiswa. Hal ini tidak hanya mengganggu proses belajar mengajar, tetapi juga mempengaruhi efisiensi operasional jurusan secara keseluruhan.

Berbagai metode optimasi telah diusulkan untuk menyelesaikan permasalahan penjadwalan, salah satunya adalah Algoritma Genetika. Algoritma Genetika adalah algoritma pencarian berbasis evolusi yang meniru proses seleksi alam, di mana solusi terbaik dipilih melalui proses seleksi, *crossover*, dan mutasi (Ansari & Saubari, 2020). Algoritma ini memiliki kemampuan untuk menyelesaikan masalah optimasi yang kompleks, seperti penjadwalan, dengan menghasilkan solusi optimal melalui pendekatan iteratif. Penelitian telah membuktikan bahwa Algoritma Genetika mampu menghasilkan jadwal yang optimal dalam menyelesaikan masalah penjadwalan perkuliahan (Fachrie & Fira Waluyo, 2020).

Keunggulan Algoritma Genetika dibandingkan dengan metode optimasi lainnya adalah kemampuannya untuk menangani masalah dengan banyak variabel dan pembatas (*constraints*) yang sering kali tidak dapat diselesaikan dengan metode konvensional (Alnowaini & Aljomai, 2021). Dalam konteks penjadwalan perkuliahan, Algoritma Genetika dapat digunakan untuk mencari solusi optimal dengan memperhatikan berbagai kendala seperti ketersediaan dosen, kapasitas ruangan, dan preferensi waktu kuliah (Urva & Sellyana, 2019). Selain itu, Algoritma Genetika juga mampu menghasilkan solusi yang mendekati optimal dalam waktu yang relatif singkat, bahkan untuk masalah penjadwalan yang kompleks.

Sistem penjadwalan otomatis berbasis web dengan implementasi Algoritma Genetika diharapkan dapat menjadi solusi efektif untuk mengatasi masalah penjadwalan di Jurusan Ilmu Komputer Universitas Lampung. Sistem ini tidak hanya mempermudah proses penjadwalan, tetapi juga mampu mengoptimalkan penggunaan sumber daya yang tersedia seperti ruangan dan dosen. Dengan demikian, sistem ini akan memberikan dampak positif bagi kelancaran operasional akademik di jurusan ini.

Melalui penelitian ini, diharapkan dapat dikembangkan sebuah layanan web (*web service*) dengan FastAPI untuk penjadwalan perkuliahan otomatis

yang mampu menyelesaikan masalah penjadwalan dengan menggunakan Algoritma Genetika. Layanan ini diharapkan dapat memberikan solusi yang optimal, efisien, dan fleksibel sehingga mampu mengurangi konflik penjadwalan yang sering terjadi.

1.2. Rumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan, maka rumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Bagaimana penerapan Algoritma Genetik dalam penjadwalan perkuliahan secara otomatis menggunakan bahasa pemrograman Python.
2. Bagaimana pengembangan *web service* dengan *framework* FastAPI untuk mendukung implementasi Algoritma Genetika dalam sistem penjadwalan perkuliahan otomatis.

1.3. Batasan Masalah

Untuk menjaga fokus penelitian, beberapa batasan yang diterapkan dalam penelitian ini adalah sebagai berikut:

1. Algoritma yang digunakan dalam penelitian ini adalah Algoritma Genetika tanpa melibatkan algoritma lain sebagai pembanding.
2. Hasil penelitian berupa *web service* yang dibangun menggunakan *framework* FastAPI.
3. Constraint yang digunakan berdasarkan pada aturan penjadwalan di jurusan Ilmu Komputer Universitas Lampung.
4. Penelitian ini difokuskan pada penjadwalan perkuliahan di Jurusan Ilmu Komputer Universitas Lampung T.A. 2023/2024.

1.4. Tujuan Penelitian

Penelitian ini memiliki beberapa tujuan yang hendak dicapai, yaitu:

1. Menerapkan Algoritma Genetik dalam penjadwalan perkuliahan secara otomatis menggunakan bahasa pemrograman Python.
2. Mengembangkan *web service* berbasis FastAPI untuk mendukung proses implementasi Algoritma Genetik dalam sistem penjadwalan perkuliahan otomatis, sehingga sistem dapat diintegrasikan dengan antarmuka pengguna berbasis web secara efisien dan responsif.

1.5. Manfaat Penelitian

Adapun manfaat yang diharapkan dari penelitian ini adalah sebagai berikut:

1. Membantu Jurusan Ilmu Komputer Universitas Lampung dalam menyusun jadwal perkuliahan secara lebih efisien, mengurangi beban administrasi, dan meminimalkan konflik jadwal.
2. Dosen akan mendapatkan jadwal yang lebih terstruktur, yang dapat mengurangi kemungkinan bentroknya jadwal mengajar.
3. Mahasiswa akan mendapatkan jadwal yang lebih optimal, dengan minimnya bentrokan antar mata kuliah, sehingga proses perkuliahan dapat berjalan lebih lancar.
4. Penelitian ini dapat menjadi referensi bagi penelitian lebih lanjut dalam pengembangan sistem penjadwalan otomatis yang lebih kompleks menggunakan metode optimasi seperti Algoritma Genetika.

II. TINJAUAN PUSTAKA

2.1. Penelitian Terdahulu

Beberapa penelitian terdahulu yang digunakan sebagai referensi dalam penelitian ini sebagai berikut.

Tabel 1. Penelitian Terdahulu

Penulis	Judul	Hasil
(Ansari & Saubari, 2020)	<i>Application of Genetic Algorithm Concept on Course Scheduling</i>	Aplikasi penjadwalan di Program Studi Informatika, Universitas Muhammadiyah Banjarmasin.
(Urva & Sellyana, 2019)	<i>Genetic Algorithm for Optimization of Lecturer Schedule Preparation</i>	Sistem penjadwalan dosen di STT Dumai menggunakan algoritma Genetika
(Modibbo et al., 2019)	<i>Genetic Algorithm for Solving University Timetabling Problem</i>	Aplikasi penjadwalan di Departemen Statistik dan Riset Operasi Universitas Teknologi Modibbo Adama Yola

2.1.1. Application of Genetic Algorithm Concept on Course Scheduling

Penelitian yang dilakukan oleh Ansari dan Saubari (2020) berjudul *Application of Genetic Algorithm Concept on Course Scheduling* fokus pada penerapan Algoritma Genetika untuk menyelesaikan masalah penjadwalan kursus di perguruan tinggi. Masalah penjadwalan perkuliahan sering kali sangat kompleks karena melibatkan banyak variabel dan pembatas (*constraints*) seperti jumlah ruangan, ketersediaan dosen, dan waktu perkuliahan. Penjadwalan manual sering kali memakan waktu lama dan rawan terjadinya kesalahan seperti bentrokan jadwal antar mata kuliah, ruang, atau dosen. Oleh karena itu, penelitian ini bertujuan untuk mengatasi permasalahan tersebut dengan menerapkan konsep Algoritma Genetika.

Penelitian ini menunjukkan bahwa Algoritma Genetika dapat diterapkan secara efektif dalam penjadwalan perkuliahan. Dalam eksperimen yang dilakukan, Algoritma Genetika berhasil menghasilkan jadwal yang optimal dengan waktu pemrosesan yang lebih singkat dibandingkan dengan metode tradisional. Hasil penelitian ini juga menunjukkan bahwa Algoritma Genetika dapat beradaptasi dengan mudah terhadap perubahan-perubahan pada pembatas, seperti perubahan ketersediaan ruang atau waktu dosen. Dengan menerapkan seleksi, *crossover*, dan mutasi secara tepat, Algoritma Genetika mampu menghasilkan solusi penjadwalan yang minim bentrokan dan sesuai dengan kebutuhan akademik.

2.1.2. Genetic Algorithm for Optimization of Lecturer Schedule Preparation

Penelitian yang dilakukan oleh Urva dan Sellyana (2021) berjudul *Genetic Algorithm for Optimization of Lecturer Schedule Preparation* berfokus pada penerapan Algoritma Genetika untuk optimasi penjadwalan dosen di institusi pendidikan tinggi. Penelitian ini bertujuan untuk mengembangkan sistem penjadwalan dosen yang optimal menggunakan Algoritma Genetika, yang mampu menghasilkan jadwal tanpa bentrokan serta sesuai dengan

preferensi dosen dan kebutuhan institusi. Hasil penelitian Urva dan Sellyana (2021) menunjukkan bahwa Algoritma Genetika mampu menghasilkan jadwal dosen yang optimal dengan sedikit bentrokan dan sesuai dengan preferensi waktu dosen. Sistem penjadwalan yang dikembangkan dalam penelitian ini juga mampu mendistribusikan beban mengajar dosen secara lebih merata, sehingga tidak ada dosen yang mendapatkan beban mengajar yang berlebihan.

2.1.3. Genetic Algorithm for Solving University Timetabling Problem

Penelitian yang dilakukan oleh Modibbo, Umar, Mijinyawa, dan Hafisu (2020) berjudul *Genetic Algorithm for Solving University Timetabling Problem* membahas penerapan Algoritma Genetika untuk menyelesaikan masalah penjadwalan di universitas. Penelitian ini bertujuan untuk memanfaatkan Algoritma Genetika dalam membangun solusi penjadwalan perkuliahan yang bebas bentrok, sekaligus memaksimalkan penggunaan sumber daya seperti ruangan dan waktu. Algoritma Genetika dipilih karena kemampuannya dalam mencari solusi optimal melalui proses iteratif yang didasarkan pada seleksi alam.

Penelitian ini menyoroti pentingnya penjadwalan dalam berbagai aspek, terutama di bidang pendidikan, sebagai salah satu masalah optimasi yang kompleks. Masalah penjadwalan pendidikan melibatkan banyak batasan dan variabel yang saling bersaing, seperti ketersediaan waktu, ruangan, dan sumber daya manusia. Algoritma optimasi seperti Algoritma Genetik menjadi pilihan populer dalam menyelesaiannya.

2.2. Uraian Landasan Teori

Berikut ini beberapa teori yang berkaitan dengan penelitian yang dilakukan.

2.2.1. Penjadwalan

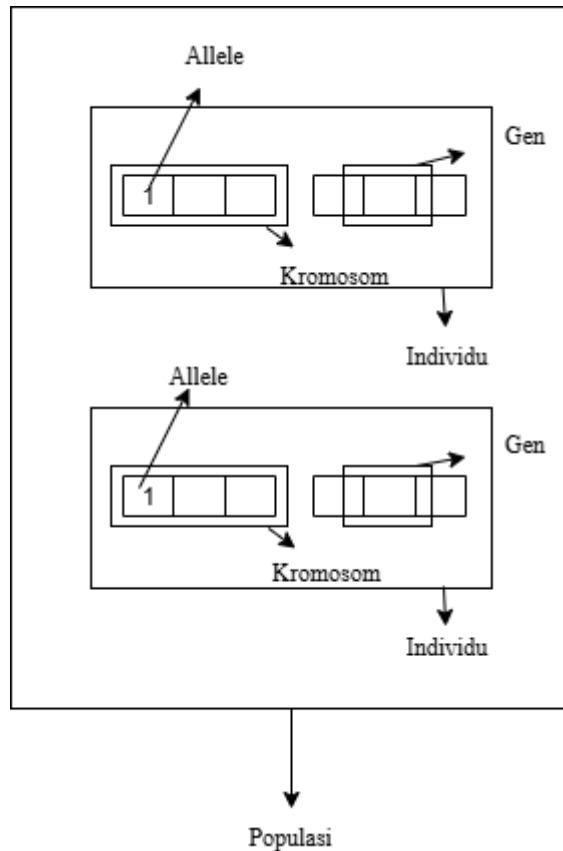
Penjadwalan adalah proses pengaturan waktu untuk berbagai aktivitas agar dapat dilakukan dengan efektif, sesuai urutan, serta mengoptimalkan penggunaan sumber daya yang ada. Dalam konteks pendidikan, penjadwalan bertujuan untuk mengorganisasi jadwal kelas, tenaga pengajar, dan ruang agar proses pembelajaran berjalan lancar dan efisien (M. C. Chen et al., 2021). Penjadwalan melibatkan pertimbangan berbagai faktor seperti ketersediaan ruangan, beban kerja pengajar, serta kebutuhan siswa dalam mengikuti berbagai mata pelajaran yang ditawarkan pada waktu yang tidak bertabrakan (Boctuan et al., 2019).

2.2.2. Algoritma Genetika

Algoritma genetika adalah metode optimasi yang meniru proses evolusi alam untuk menyelesaikan masalah kompleks. Algoritma ini bekerja dengan menciptakan populasi awal kandidat solusi yang berevolusi melalui operasi seleksi, persilangan, dan mutasi, sehingga menghasilkan solusi optimal dengan pendekatan iteratif. (Lambora et al., 2019). Algoritma ini fleksibel, efektif dalam penjadwalan, dan dapat disesuaikan untuk kebutuhan spesifik, seperti mengatur penjadwalan kelas atau tugas komputasi (R. Chen et al., 2024). Kelebihan algoritma genetika adalah fleksibilitas dalam menangani berbagai kendala dan tujuannya untuk menghasilkan jadwal tanpa bentrok dan mengoptimalkan sumber daya yang ada (Amindoust et al., 2021).

2.2.2.1. Komponen Algoritma Genetik

Algoritma Genetik terdiri dari beberapa komponen utama yang berperan dalam proses evolusi buatan untuk menemukan solusi optimal. Setiap komponen memiliki fungsi spesifik yang meniru mekanisme seleksi alam dalam biologi. Komponen-komponen ini bekerja secara iteratif untuk memperbaiki kualitas solusi hingga memenuhi kriteria yang diinginkan. Berikut adalah komponen utama dalam Algoritma Genetik.



Gambar 1. Komponen Algoritma Genetik

Populasi adalah kumpulan solusi kandidat atau individu yang dihasilkan dalam iterasi awal. Populasi awal ini menjadi dasar untuk iterasi selanjutnya, dimana individu-individu yang lebih sesuai atau "fit" dipilih untuk dikombinasikan dan dimodifikasi guna menghasilkan generasi solusi baru. Teknik inisialisasi populasi yang baik dapat mempercepat konvergensi terhadap solusi optimal dengan mengurangi individu berfitness rendah, yang akhirnya meningkatkan efisiensi algoritma secara keseluruhan (Alkafaween et al., 2024). Contoh populasi dapat dilihat pada gambar berikut:

Tabel Ruang

id	Nama Ruang	Kapasitas
1	GIK L1 A	40
2	GIK L1 C	120
3	LAB R1	25
4	LAB R2	25
5	MIPA T L1 A	80
6	MIPA T L1 B	80

Tabel Mata Kuliah

id	Kode Mata Kuliah	Mata Kuliah	SKS
1	COM620101	Aljabar Linear	2
2	COM620108	Pemrograman Terstruktur	3
3	COM620214	Kecerdasan Buatan	3
4	COM620310	Pengenalan Pola	3
5	COM620401	Kapita Selektiva	2
6	COM620406	Komputasi Paralel	3

Tabel Hari

id	Hari
1	Senin
2	Selasa
3	Rabu
4	Kamis
5	Jumat

Tabel Sesi

id	mulai	selesai
1	07:30	09:10
2	09:20	11:00
3	11:10	12:50
4	13:30	15:10
5	15:30	17:00

Tabel Dosen

id	Dosen	idMatkul
1	Dr. Aristoteles, S.Si., M.Si.	3
2	Anie Rose Irawati, ST., M.Cs.	5
3	Bambang Hermanto, S.Kom., M.Cs	4
4	Didik Kurniawan, S. Si, M.T	2
5	Prof. Admi Syarif, Ph.D	1
6	Dwi Sakethi, S.Si, M.Kom	6

Gambar 2. Contoh Data yang akan Diolah

Dalam Algoritma Genetika, individu merujuk pada satu solusi potensial dalam populasi yang sedang dievaluasi. Setiap individu biasanya direpresentasikan sebagai kromosom, yang terdiri dari serangkaian gen yang menyimpan informasi spesifik terkait solusi tersebut. Proses evolusi dalam Algoritma Genetika melibatkan seleksi individu-individu dengan nilai fitness tertinggi untuk menghasilkan generasi berikutnya melalui operasi seperti crossover dan mutasi. Tujuan utamanya adalah mengoptimalkan solusi hingga mencapai hasil yang diinginkan (Cahya Yustina et al., 2024). Selain itu, individu-individu dalam Algoritma Genetika dievaluasi berdasarkan nilai fitness mereka. Jika individu melanggar batasan tertentu, konsep penalti diterapkan dengan mengurangi nilai fitness individu tersebut. Teknik ini memastikan bahwa solusi yang dihasilkan memenuhi kriteria yang telah ditetapkan dan mendorong pencarian solusi yang lebih optimal (Rohmad & Akbar, 2024).

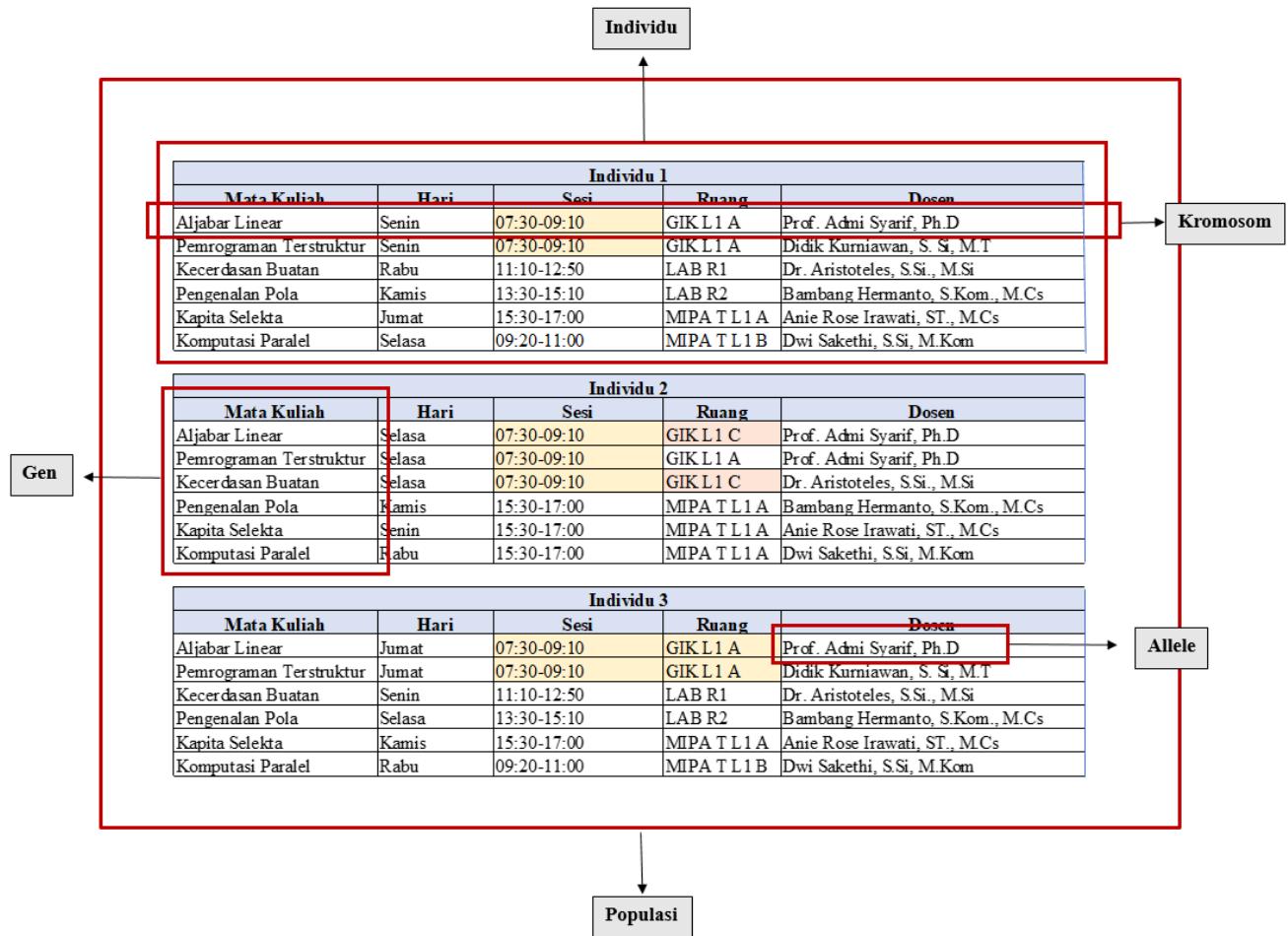
Kromosom adalah representasi dari solusi potensial dalam bentuk struktur data tertentu, seperti bit atau urutan angka, yang berfungsi sebagai unit dasar dalam proses optimasi. Kromosom ini mengkodekan informasi yang akan dievaluasi dan dimodifikasi melalui proses evolusi, seperti seleksi,

persilangan, dan mutasi, untuk mendekati solusi optimal. Tiap kromosom terdiri dari "gen" yang menyimpan nilai parameter yang berbeda (Singh et al., 2018).

Gen merujuk pada unit terkecil dalam kromosom yang mewakili satu atribut atau nilai spesifik dari solusi potensial. Setiap gen dapat dikodekan untuk merepresentasikan variabel tertentu yang relevan dengan masalah optimasi, seperti waktu atau lokasi dalam permasalahan penjadwalan. Gen-gen ini bergabung membentuk kromosom, yang berfungsi sebagai kandidat solusi yang akan dievaluasi dan dioptimalkan. Dengan proses seleksi, *crossover*, dan mutasi, gen-gen dalam populasi dapat "berevolusi" menuju solusi yang lebih optimal, membuat algoritma genetika menjadi efektif dalam menemukan solusi terbaik pada ruang pencarian yang kompleks (Hassanat et al., 2019).

Allele mengacu pada nilai spesifik dari suatu gen dalam kromosom individu yang mewakili solusi potensial dalam populasi. Setiap allele mengkodekan karakteristik tertentu yang mempengaruhi bagaimana solusi tersebut bekerja dalam proses evolusi dan seleksi. Variasi nilai allele memungkinkan pencarian solusi optimal melalui proses *crossover* dan mutasi, yang secara efektif memperkaya keragaman genetik dalam populasi dan membantu menemukan solusi yang lebih baik (Noskova et al., 2020).

Berikut contoh beberapa individu yang dibentuk berdasarkan data yang ada. Disini diberikan contoh beberapa konflik yang terjadi dalam satu populasi.



Gambar 3. Plotting Jadwal

Jumlah individu dalam suatu populasi mengacu pada berapa banyak solusi potensial yang dievaluasi dalam satu generasi, sedangkan panjang kromosom dalam tiap individu mengacu pada jumlah gen dalam kromosom tersebut yang merepresentasikan parameter atau elemen dari solusi. Jumlah individu biasanya ditentukan berdasarkan kompleksitas masalah. Untuk masalah sederhana, populasi kecil dapat digunakan, sementara masalah lebih kompleks memerlukan populasi yang lebih besar untuk menjaga keragaman solusi. Panjang kromosom, di sisi lain, tergantung pada jumlah variabel yang diperlukan untuk merepresentasikan solusi (Hassanat et al., 2019). Namun, dalam kebanyakan kasus GA, panjang kromosom adalah sama untuk semua individu karena mereka merepresentasikan parameter yang sama untuk masalah yang sama (Prambudi et al., 2021).

2.2.2.2. Fungsi *Fitness*

Dalam algoritma genetika, fungsi *fitness* adalah komponen penting yang menilai kualitas solusi atau “kromosom” dalam proses evolusi. Fungsi ini memberikan skor atau nilai yang menunjukkan seberapa baik solusi tertentu dalam memenuhi tujuan atau memenuhi kriteria masalah. Dalam konteks tertentu, fungsi *fitness* digunakan untuk mengukur jarak atau kesesuaian solusi dengan hasil yang diinginkan (Yang & Zhou, 2024). Solusi yang memiliki nilai *fitness* lebih tinggi akan lebih mungkin dipilih untuk proses seleksi, yang mencakup tahap-tahap seperti *crossover* dan mutasi untuk membentuk generasi berikutnya. Dalam penjadwalan, nilai *fitness* ditentukan oleh penalti seperti bentrok dosen dan bentrok ruang, dan bentrok lainnya.

Jumlah penalti yang ada seperti tumpang tindih jadwal pada ruang kelas, waktu, atau dosen, menjadi salah satu parameter penting yang digunakan untuk menilai kualitas solusi (Pătrăușanu et al., 2024). Konflik ini mempengaruhi nilai *fitness* dalam algoritma genetika, di mana semakin sedikit konflik, semakin baik solusi yang dihasilkan. Batas maksimum jumlah konflik biasanya tidak ditentukan secara langsung, tetapi dihitung berdasarkan kombinasi parameter seperti jumlah kelas, dosen, ruang, dan sesi waktu. Dalam implementasi praktis, jika solusi memiliki konflik yang melebihi ambang tertentu, solusi tersebut biasanya dianggap tidak layak dan dapat dihilangkan selama proses seleksi (Sánchez-Chica et al., 2019). Berikut contoh perhitungan fungsi *fitness*:

$$\text{Fitness} = \text{total fitness} - \text{penalty}$$

Misal diberikan nilai awal total fitness sebesar 100 dan tiap penalti diberikan nilai 20. Yang artinya jika individu yang dihasilkan bagus atau tidak ada penalti, maka nilai fitness-nya sama dengan total fitness awal yaitu 100.

	Individu 1	Individu 2	Individu 3
Jenis Penalti	Konflik ruangan (lebih dari satu mata kuliah di ruangan yang sama pada waktu yang sama)	1. konflik dosen yang mengajar lebih dari satu mata kuliah dalam sesi yang sama 2. konflik ruangan, yaitu ruangan yang sama digunakan untuk lebih dari satu mata kuliah pada waktu yang sama.	Konflik Kelas (satu kelas memiliki lebih dari satu mata kuliah di waktu yang sama)
Nilai penalti	20	40	20
Total Fitness	100 - 20 = 80	100 - 40 = 60	100 - 20 = 80

Gambar 4. Perhitungan *Fitness*

2.2.2.3. Seleksi

Tahap seleksi dalam algoritma genetika berfungsi untuk memilih individu-individu terbaik dari populasi untuk menghasilkan generasi berikutnya. Seleksi ini dilakukan berdasarkan nilai *fitness* setiap individu, yang menunjukkan seberapa baik solusi tersebut dalam memenuhi tujuan optimasi (Aliani et al., 2024). Individu dengan nilai *fitness* lebih tinggi memiliki peluang lebih besar untuk dipilih dan bereproduksi, sehingga karakteristik mereka diwariskan ke generasi berikutnya. *Fitness* yang telah dihasilkan di tahapan sebelumnya akan dihitung probabilitas dan kumulatifnya.

$$\text{Probabilitas} = \frac{\text{Fitness}[i]}{\text{Total Fitness}}$$

$$\text{Total fitness populasi: } 80 + 60 + 80 = 220$$

$$\text{Probabilitas Individu 1: } \frac{80}{220} = 0.3636$$

$$\text{Probabilitas Individu 2: } \frac{60}{220} = 0.2727$$

$$\text{Probabilitas Individu 3: } \frac{80}{220} = 0.3636$$

Gambar 5. Probabilitas Individu

Tahap selanjutnya yaitu memilih individu berdasarkan probabilitas yang ditentukan oleh nilai *fitness* mereka dengan metode *Roulette Wheel*. *Roulette Wheel* dipilih karena sudah diakui sebagai metode seleksi yang paling umum dan sederhana, dengan setiap individu dalam populasi diberikan bagian dari roda roulette imajiner yang proporsional dengan nilai *fitness* mereka ((Lata et al., 2017)).

Roulette wheel selection, juga dikenal sebagai *fitness proportionate selection*, adalah metode yang digunakan dalam algoritma genetika untuk memilih individu dari populasi berdasarkan probabilitas yang sebanding dengan nilai *fitness* mereka. Prosesnya mirip dengan putaran roda roulette dimana setiap individu diberikan "slice" dari roda yang ukurannya proporsional dengan *fitness* mereka. Individu dengan *fitness* lebih tinggi memiliki peluang lebih besar untuk dipilih sebagai parent untuk generasi berikutnya. Metode ini memastikan bahwa individu dengan *fitness* lebih tinggi memiliki peluang lebih besar untuk dipilih, namun tetap memberikan kesempatan kepada individu dengan *fitness* rendah untuk terlibat dalam proses, menjaga keragaman dalam populasi (Das & Kumar Pratihar, 2019).

Berbeda dengan *Tournament Selection*, yang cenderung memilih individu dengan *fitness* tinggi secara langsung, *Roulette Wheel* tetap memberikan kesempatan bagi individu dengan *fitness* lebih rendah untuk terpilih. *Roulette Wheel* juga lebih efisien dibanding *Rank-Based Selection*, yang sering kali memperlambat evolusi karena distribusi probabilitas seleksi yang merata. Karena probabilitas seleksi proporsional terhadap *fitness*, individu yang memiliki solusi lebih baik akan lebih sering terpilih, namun masih ada kemungkinan individu lain dipilih, yang membantu eksplorasi solusi global (Isa et al., 2024). Interval *Roulette* dapat dilihat pada gambar berikut:

Individu	Probabilitas	Interval Roulette
Individu 1	0.3636	0.000 - 0.3636
Individu 2	0.2727	0.3637 - 0.6363
Individu 3	0.376	0.6364 - 1.000

Gambar 6. Interval *Roulette*

Setelah interval *roulette* didapatkan, selanjutnya adalah memilih individu berdasarkan angka acak yang dihasilkan dalam interval [0, 1]. Berikut adalah simulasi pemilihan individu:

Pada putaran pertama, dihasilkan angka acak yaitu 0.25. Karena 0.25 berada di interval *roulette* individu 1 ($0.3636 < 0.25 \leq 0.6363$), maka individu 1 terpilih. Sedangkan pada putaran kedua, dihasilkan angka acak yaitu 0.80. Karena 0.80 berada di interval *roulette* individu 3 ($0.634 < 0.80 \leq 1$), maka individu 3 terpilih.

2.2.2.4. *Crossover*

Crossover adalah salah satu operator utama yang memungkinkan pertukaran informasi antara dua solusi induk untuk menghasilkan keturunan baru dengan gabungan sifat dari keduanya. *Crossover* bertujuan untuk menjaga dan memperkuat karakteristik yang baik dari kedua induk dalam upaya mencari solusi optimal (Almanjahie & Ahmad, 2024). Uniform crossover adalah salah satu metode dalam algoritma genetika yang digunakan untuk menghasilkan individu baru (*offspring*) dengan menggabungkan informasi genetik dari dua induk (*parents*). Berbeda dengan metode *crossover* tradisional seperti *one-point* atau *two-point crossover* yang membagi kromosom pada titik tertentu, *uniform crossover* memperlakukan setiap gen secara independen. Dalam proses ini, untuk setiap posisi gen dalam kromosom, dipilih secara acak apakah gen tersebut

akan diwarisi dari induk pertama atau induk kedua, biasanya dengan probabilitas yang sama (Syswerda, 2018).

Berdasarkan hasil seleksi sebelumnya, individu yang terpilih adalah individu 1 dan individu 3. Pada tahap ini ditentukan titik *crossover* secara *uniform*. Yang berarti nantinya pada *offspring* 1 data pada baris pertama, ketiga, kelima diambil dari individu 1 sedangkan baris kedua, keempat, dan keenam diambil dari individu 3, begitu seterusnya. Sedangkan pada *offspring* 2 sebaliknya, yaitu data pada baris pertama, ketiga, kelima diambil dari individu 3 sedangkan baris kedua, keempat, dan keenam diambil dari individu 1.

Hasil *offspringnya* sebagai berikut:

- Offspring* 1 (gabungan dari Individu 1 dan Individu 3)

Mata Kuliah	Hari	Sesi	Ruang	Dosen
Aljabar Linear	Senin	07:30-09:10	GIK L1 A	Prof. Admi Syarif, Ph.D
Pemrograman Terstruktur	Jumat	07:30-09:10	GIK L1 A	Didik Kurniawan, S. Si, M.T
Kecerdasan Buatan	Rabu	11:10-12:50	LAB R1	Dr. Aristoteles, S.Si., M.Si
Pengenalan Pola	Selasa	13:30-15:10	LAB R2	Bambang Hermanto, S.Kom., M.Cs
Kapita Selekta	Jumat	15:30-17:00	MIPA T L1 A	Anie Rose Irawati, ST., M.Cs
Komputasi Paralel	Rabu	09:20-11:00	MIPA T L1 B	Dwi Sakethi, S.Si, M.Kom

Gambar 7. Hasil *Offspring* 1

Gambar tersebut menunjukkan hasil dari *offspring* 1 atau gabungan antara individu 1 dengan individu 3 dimana baris 1, 3, 5 pada tabel adalah milik individu 1, sedangkan baris 2, 4, dan 6 adalah milik individu 3.

b. *Offspring 2* (gabungan dari Individu 3 dan Individu 1)

Mata Kuliah	Hari	Sesi	Ruang	Dosen
Ajabar Linear	Jumat	07:30-09:10	GIK L1 A	Prof. Admi Syarif, Ph.D
Pemrograman Terstruktur	Senin	07:30-09:10	GIK L1 A	Didik Kurniawan, S. Si, M.T
Kecerdasan Buatan	Senin	11:10-12:50	LAB R1	Dr. Aristoteles, S.Si., M.Si
Pengenalan Pola	Kamis	13:30-15:10	LAB R2	Bambang Hermanto, S.Kom., M.Cs
Kapita Selekta	Kamis	15:30-17:00	MIPA T L1 A	Anie Rose Irawati, ST., M.Cs
Komputasi Paralel	Selasa	09:20-11:00	MIPA T L1 B	Dwi Sakethi, S.Si, M.Kom

Gambar 8. Hasil *Offspring 2*

Gambar tersebut menunjukkan hasil dari *offspring 2* atau gabungan antara individu 3 dengan individu 1 dimana baris 1, 3, 5 pada tabel adalah milik individu 3, sedangkan baris 2, 4, dan 6 adalah milik individu 1.

2.2.2.5. Mutasi

Mutasi adalah proses yang mengubah nilai gen dalam individu secara acak. Mutasi biasanya terjadi dengan probabilitas tertentu dan bekerja dengan mengubah nilai gen pada kromosom untuk menciptakan individu yang lebih bervariasi, yang dapat memperbaiki kemampuan algoritma untuk menemukan solusi optimal (Hassanat et al., 2019). Berikut tahap Mutasi:

a. Memilih Individu yang Akan Dimutasi

Individu yang dipilih biasanya adalah individu yang nilai *fitnessnya* lebih rendah, karena individu ini dianggap memiliki kualitas yang kurang baik untuk solusi optimal. Ini dilakukan untuk memberi peluang individu "kurang baik" agar berkembang lebih baik melalui mutasi. Berdasarkan perhitungan *fitness* masing-masing individu pada hasil *offspring*, nilai *fitness* pada kedua individu adalah sama karena sudah tidak ada bentrok yang dihasilkan.

b. Tentukan Probabilitas Mutasi

Probabilitas mutasi adalah nilai yang menentukan seberapa sering mutasi akan terjadi pada populasi atau individu dalam algoritma genetika. Probabilitas mutasi dihitung berdasarkan sejumlah faktor yang bisa disesuaikan. Nilai ini biasanya berkisar antara 0.01 (1%) hingga 0.1 (10%) dalam banyak kasus, meskipun nilai yang lebih kecil atau lebih besar bisa digunakan tergantung pada kebutuhan spesifik masalah yang dihadapi (Vie et al., 2020).

c. Pilih Gen yang Dimutasi

Untuk memutuskan apakah sebuah gen akan dimutasi, kita menghasilkan angka acak antara 0 dan 1 untuk setiap gen. Angka ini disimulasikan dengan generator angka acak dalam algoritma genetika. Jika angka acak $<$ Probabilitas, maka gen tersebut dimutasi, jika \geq Probabilitas, maka gen tidak dimutasi.

Misal dari hasil *crossover* sebelumnya, karena kedua hasil offspring sudah tidak ada bentrok lagi, maka proses mutasi bisa dilewati, jadi jika tidak ada mutasi dan kedua *offspring* sudah optimal, maka kedua offspring akan menjadi solusi. Namun jika hanya satu solusi yang dibutuhkan, maka salah satu *offspring* bisa dipilih secara *random*.

2.2.3. Library PyGAD pada Python

PyGAD adalah sebuah pustaka Python yang digunakan untuk mengimplementasikan Algoritma Genetika. *Library* ini memungkinkan pengguna untuk memecahkan berbagai masalah optimasi dengan cara yang meniru proses evolusi alamiah, seperti seleksi, *crossover*, dan mutasi (Gad, 2021). PyGAD dirancang agar fleksibel dan mudah digunakan,

memungkinkan pengguna untuk mendefinisikan fungsi *fitness* mereka sendiri serta menyesuaikan proses evolusi sesuai dengan kebutuhan masalah yang dihadapi. Untuk menggunakan library PyGAD, dilakukan 3 langkah berikut.

2.2.3.1. Membangun fungsi fitness

```
import pygad
import numpy

equation_inputs = [4, -2, 3.5]
Y = 44

def fitness_func(solution, solution_idx):
    out = numpy.sum(solution * equation_inputs)
    fitness = 1.0/(numpy.abs(out - Y)+0.000001)
    return fitness
```

Gambar 9. *Source Code Fungsi Fitness*

Pada gambar 2 ditunjukkan pembuatan *fitness function*, yaitu fungsi yang akan mengukur seberapa baik solusi (individu) dalam populasi memecahkan masalah. Dalam kasus ini, karena masalahnya adalah *maximization problem*, fungsi ini harus mengembalikan nilai yang lebih tinggi untuk solusi yang lebih baik.

2.2.3.2. Buat *instance* kelas `pygad.GA`.

```
ga_instance = pygad.GA(num_generations=100,
                      sol_per_pop=10,
                      num_parents_mating=5,
                      num_genes=3,
                      fitness_func=fitness_func)
```

Gambar 10. *Source Code Instance Kelas PyGAD*

Kelas ini digunakan untuk mengatur berbagai parameter dari algoritma genetika, seperti ukuran populasi, jumlah generasi, probabilitas mutasi dan *crossover*, serta fitness function yang akan digunakan.

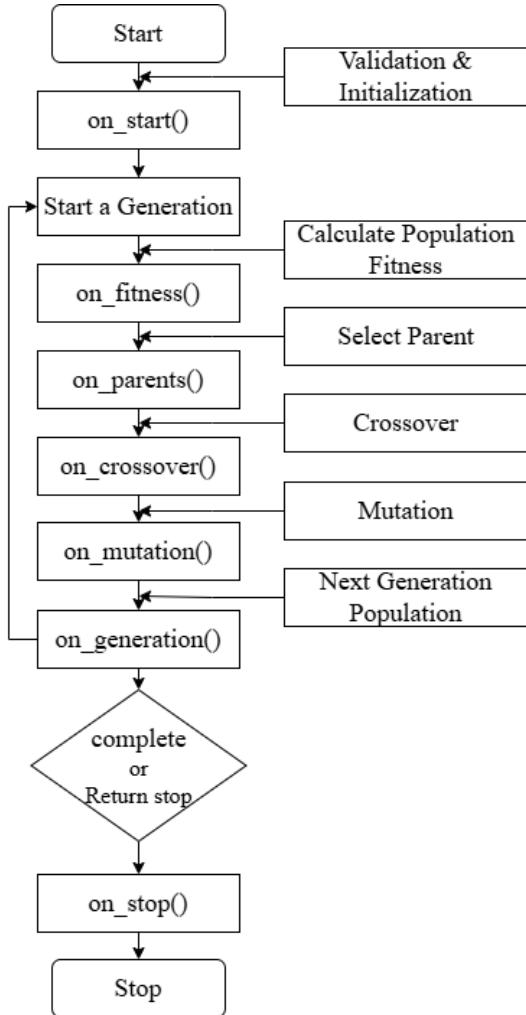
2.2.3.3. Panggil *Method run()*

g a _ i n s t a n c e . r u n ()

Gambar 11. Source Code *Method run()*

Langkah ini memanggil metode *run()* dari *instance* algoritma genetika yang telah dibuat. Metode ini menjalankan algoritma genetika sesuai dengan parameter yang telah diatur pada *instance* algoritma genetika. Setelah *run()* dipanggil, proses evolusi berjalan, menghasilkan solusi yang lebih baik seiring waktu.

Dalam contoh ini, algoritma genetika di PyGAD diatur untuk menjalankan 100 generasi, dengan 5 solusi dipilih untuk proses perkawinan, dan populasi memiliki 10 solusi yang masing-masing terdiri dari 3 gen. Siklus dari PyGAD dapat dilihat pada gambar 12.



Gambar 12. Siklus PyGAD

Siklus PyGAD dimulai setelah memanggil *method run()*. Ada 7 *function* dalam siklus pyGAD yang fungsinya sebagai berikut:

1. `On_start()` : Dipanggil sekali setelah metode `run()` dipanggil.
2. `on_fitness()` : Dipanggil setelah menghitung fitness populasi di setiap generasi.
3. `on_parents()` : Dipanggil di setiap generasi setelah orang tua dipilih.
4. `on_crossover()` : Dipanggil di setiap generasi setelah operasi crossover diterapkan.

5. `on_mutation()` : Dipanggil di setiap generasi setelah operasi mutasi diterapkan.
6. `on_generation()` : Dipanggil di akhir setiap generasi.

2.2.4. Sistem Informasi Akademik (SIAKAD)

SIAKAD (Sistem Informasi Akademik) adalah sistem informasi berbasis teknologi yang digunakan oleh perguruan tinggi untuk mengelola data dan aktivitas akademik. Sistem ini mencakup berbagai fungsi, seperti data mahasiswa, pengelolaan jadwal perkuliahan, pengisian KRS (Kartu Rencana Studi), pengolahan nilai, hingga pelaporan akademik (Hikmawati et al., 2023). Tujuan utama SIAKAD adalah mempermudah administrasi akademik dan meningkatkan efisiensi layanan kepada mahasiswa, dosen, dan pihak administrasi kampus (Sofiyah et al., 2019). SIAKAD Universitas Lampung memuat seluruh informasi seperti data mata kuliah, dosen, kelas, kapasitas kelas, sesi, dan masih banyak lagi yang mendukung proses perkuliahan.

2.2.5. Web Service

Web service adalah teknologi yang memungkinkan aplikasi berbeda untuk berkomunikasi melalui jaringan, seperti internet, menggunakan protokol standar seperti HTTP dan format data seperti XML atau JSON (Neumann et al., 2021). Kegunaannya terletak dalam memfasilitasi sistem yang beragam, memungkinkan pengembangan perangkat lunak yang lebih fleksibel dan terintegrasi. *Web service* menjadi pondasi utama dalam pengembangan arsitektur berbasis layanan, seperti RESTful APIs, yang banyak digunakan dalam aplikasi modern dan platform berbasis *cloud* (Taherdoost, 2023).

2.2.6. FastAPI

FastAPI adalah kerangka kerja web berbasis *Python* yang dirancang untuk membangun API RESTful dengan performa tinggi. FastAPI juga memungkinkan pengembangan *web services* Python yang aman, skalabel, dan terstruktur. Kerangka ini menonjol karena mendukung pemrograman asinkron yang mempercepat respons API dan mampu menangani banyak permintaan secara bersamaan, menjadikannya ideal untuk aplikasi dengan lalu lintas tinggi. FastAPI memanfaatkan tipe data *Python* untuk validasi otomatis, serta menyertakan dokumentasi API secara otomatis melalui OpenAPI, sehingga mengurangi kesalahan dan meningkatkan produktivitas pengembangan (Tragura, 2022).

III. METODOLOGI PENELITIAN

3.1. Waktu dan Tempat Penelitian

Penelitian dilakukan pada Semester Ganjil Tahun Ajaran 2024/2025 di Jurusan Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Lampung.

3.2. Perangkat Penelitian

Perangkat yang digunakan dalam penelitian ini adalah:

3.2.1. Perangkat Keras

Perangkat keras yang digunakan dalam penelitian ini adalah sebuah laptop dengan spesifikasi sebagai berikut.

- a. *System Manufacturer* : ASUS
- b. *System Model* : ASUS VivoBook
- c. *Processor* : 11th Gen Intel® Core™ i5-1135G7 CPU
@ 2.40GHz (4 cores, 8 threads)
- d. *Graphic* : Graphics: Intel® Iris® Xe Graphics
- e. *Memory* : 8.00 GB RAM

f. *System type* : 64-bit operating system, x64-based processor

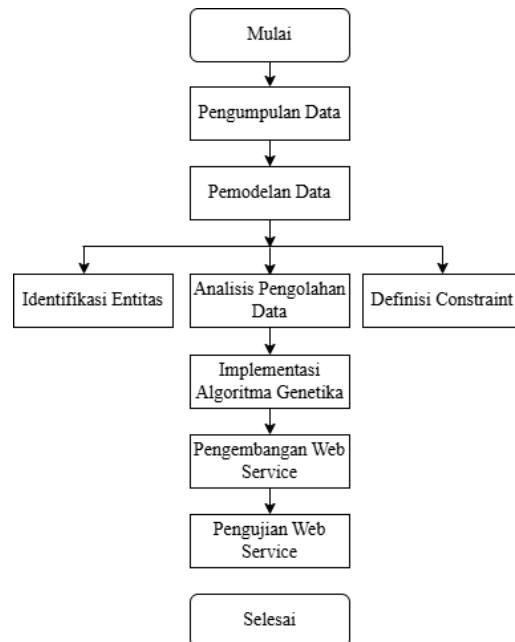
3.2.2. Perangkat Lunak

Perangkat lunak yang digunakan dalam penelitian ini adalah sebagai berikut.

- a. Sistem Operasi Windows 11 Pro 64-bit
- b. Web Browser Google Chrome dan Microsoft Edge
- c. Visual Studio Code
- d. Python
- e. FastAPI
- f. Postman Agent

3.3. Tahapan Penelitian

Tahapan yang akan dilakukan pada penelitian ini adalah sebagai berikut.



Gambar 13. Tahapan Penelitian

3.3.1. Pengumpulan Data

Data yang diperlukan dalam penelitian ini meliputi data jadwal perkuliahan, data dosen, data mata kuliah, dan data ruang kelas. Pengumpulan data ini bisa dilakukan melalui wawancara, Sistem Informasi Akademik (SIAKAD) Universitas Lampung, dan observasi. Data ini menjadi dasar untuk model penjadwalan yang akan dioptimalkan menggunakan algoritma genetika.

3.3.1.1. Wawancara

Wawancara dalam penelitian ini melibatkan pihak-pihak yang secara langsung terlibat dalam penyusunan jadwal perkuliahan, yaitu Sekretaris Jurusan dan staf Badan Khusus yang bertugas menyusun jadwal untuk para asisten dosen yang bertanggung jawab atas penjadwalan di Jurusan Ilmu Komputer Universitas Lampung. Melalui wawancara ini, akan dikumpulkan daftar *constraint* (batasan) yang digunakan dalam proses penyusunan jadwal, yang mencakup aturan umum hingga aturan internal khusus yang harus dipatuhi dalam penjadwalan perkuliahan. Berikut informasi yang didapat dari hasil wawancara:

1. Jumlah ruangan yang ada belum cukup untuk menampung seluruh kelas, terutama ketika jumlah mahasiswa mengalami peningkatan. Hal ini mengakibatkan beberapa kelas terpaksa dilaksanakan secara daring (online) untuk mengakomodasi keterbatasan kapasitas ruangan.
2. Penjadwalan semakin kompleks ketika mencakup dosen penanggung jawab (PJ) dan dosen anggota. Keduanya tidak diperbolehkan memiliki jadwal yang bertabrakan atau mengajar mata kuliah berbeda dalam sesi yang sama. Oleh karena itu, diperlukan penyesuaian tambahan dalam proses penjadwalan untuk memenuhi aturan internal jurusan.

3.3.1.2. Sistem Informasi Akademik (SIAKAD) dan Observasi

Pada tahap ini dikumpulkan data kurikulum, periode akademik, mata kuliah, program studi, nama kelas, dosen pengajar, sesi perkuliahan, serta kapasitas kelas. Data ini diambil dari menu kelas kuliah di SIAKAD yang mencakup Periode Akademik, Program Studi, Kurikulum, mata Kuliah, Prodi pengampu, Nama Kelas, Pengajar, Jadwal Mingguan, dan Kapasitas. Untuk periode akademik akan digunakan periode 2023 Ganjil dan 2023 Genap. Kurikulum akan digunakan kurikulum 2020.

Observasi langsung dilakukan untuk mendapatkan informasi tentang data ruangan dan kapasitas ruangan yang tersedia di Jurusan Ilmu Komputer Universitas Lampung. Informasi ini akan digunakan untuk memastikan alokasi ruangan supaya optimal dalam penjadwalan perkuliahan.

3.3.2. Pemodelan Data

Pemodelan data melibatkan strukturisasi data yang telah dikumpulkan agar dapat diolah oleh algoritma genetika. Tahap ini mencakup representasi data dalam bentuk yang sesuai dengan persyaratan pemrograman, seperti pemetaan mata kuliah, dosen, ruang kelas, dan waktu ke dalam parameter algoritma. Model data juga memperhatikan batasan-batasan seperti jumlah maksimum kelas yang bisa dijadwalkan dan preferensi dosen.

3.3.2.1. Identifikasi Entitas

Langkah pertama dalam pemodelan data adalah mengidentifikasi entitas utama yang akan digunakan dalam sistem. Entitas adalah objek-objek yang memiliki relevansi dan memerlukan representasi dalam basis data. Dalam kasus penjadwalan perkuliahan, entitas utama meliputi:

- a. Dosen : Merepresentasikan pengajar yang memiliki jadwal ketersediaan tertentu.
- b. Mata Kuliah : Representasi dari subjek yang dijadwalkan.
- c. Ruangan : Tempat berlangsungnya kegiatan perkuliahan.
- d. Kelas Prodi : Grup mahasiswa dalam program studi tertentu.
- e. Kelas Kuliah : Kelompok kelas untuk mata kuliah tertentu.
- f. Hari : Hari dalam seminggu di mana perkuliahan dijadwalkan.
- g. Sesi : Waktu dalam hari untuk jadwal perkuliahan.

3.3.2.2. Analisis Pengolahan Data

Dalam tahap ini akan dihitung jumlah data dari setiap entitas, yang nantinya akan digunakan untuk membuat populasi awal dan perhitungan fitness untuk tahapan dalam implementasi algoritma genetika. Data yang akan dihitung jumlahnya yaitu total jumlah Dosen, Mata Kuliah, Ruangan, Kelas Prodi, Kelas Kuliah, Hari, dan Sesi. Berikut jumlah data untuk tiap entitas:

a. Mata kuliah

Tabel 2. Jumlah Mata Kuliah Semester Ganjil

Prodi	Semester	Status	Jumlah
S1 Ilmu Komputer	1	Wajib	13
		Peminatan	0
	3	Wajib	8
		Peminatan	4
	5	Wajib	6
		Peminatan	5
	7	Wajib	1
		Peminatan	7
	1	Wajib	12
		Peminatan	0
D3 Manajemen Informatika	3	Wajib	9
		Peminatan	0
	5	Wajib	6
		Peminatan	0
	Total		71

Tabel 3. Jumlah Mata Kuliah Semester Genap

Prodi	Semester	Status	Jumlah
S1 Ilmu Komputer	2	Wajib	9
		Peminatan	0
	4	Wajib	6
		Peminatan	4
	6	Wajib	1
		Peminatan	7
	2	Wajib	9
		Peminatan	0
	4	Wajib	8
		Peminatan	0
Total			44

b. Dosen

Tabel 4. Jumlah Dosen

Jenis Dosen	Jumlah
Dosen Jurusan Ilmu Komputer	22
Dosen Luar Ilmu Komputer	16
Total	38

c. Ruangan

Tabel 5. Jumlah Ruangan

Ruangan	Jumlah
Kelas	6
Lab	5
Total	11

d. Kelas Program Studi

Tabel 6. Jumlah Kelas Prodi

Program Studi	Jumlah
S1 Ilmu Komputer	17
D3 Manajemen Informatika	1
Total	18

e. Kelas Kuliah

Tabel 7. Jumlah Kelas Kuliah Semester Ganjil

Program Studi	Jumlah
S1 Ilmu Komputer	113
D3 Manajemen Informatika	28
Total	141

Tabel 8. Jumlah Kelas Kuliah Semester Genap

Program Studi	Jumlah
S1 Ilmu Komputer	82
D3 Manajemen Informatika	18
Total	100

f. Hari

Proses perkuliahan di Jurusan Ilmu Komputer Universitas Lampung dilakukan selama total 5 hari yaitu hari Senin, Selasa, Rabu, Kamis, dan Jumat.

g. Sesi

Dalam satu hari, ada total 5 sesi perkuliahan.

Tabel 9. Tabel Sesi

Sesi	Mulai	Selesai
1	7:30	9:10
2	9:20	11:00
3	11:10	12:50
4	13:30	15:10
5	15:30	17:00

3.3.2.3. Definisi Batasan (*Constraint*)

Constraint yang digunakan pada penelitian ini mengacu pada hasil wawancara dengan Sekretaris Jurusan Ilmu Komputer Universitas Lampung. Berikut daftar *Constraint* yang harus dipenuhi:

- a. **Setiap kelas teori atau responsi hanya dijadwalkan satu kali pada satu ruangan yang memenuhi kriteria kapasitas atau ruang online.**

Constraint ini berarti setiap mata kuliah teori atau responsi hanya boleh berlangsung dalam satu ruangan yang kapasitasnya mencukupi atau dilakukan secara online. Tidak boleh ada duplikasi jadwal untuk kelas yang sama pada waktu berbeda. Bentuk representasi matematisnya adalah:

$$P = w_1 \sum_{i=1}^n \max \left(0, \sum_{j=1}^n 1(C_i = C_j) - 1 \right) + w_2 \sum_{i=1}^n 1(cap(R_i) < jumlahMahasiswa(C_i) \text{ dan } R_i \neq Online)$$

C_i = Kelas yang dijadwalkan

R_i = Ruangan yang digunakan

Artinya, untuk setiap kelas teori atau responsi, hanya boleh ada satu jadwal di ruangan yang kapasitasnya mencukupi atau dilakukan secara online.

- b. **Setiap kelas praktikum hanya boleh dijadwalkan satu kali pada satu ruangan laboratorium.**

Kelas praktikum harus berlangsung dalam satu sesi, hanya boleh dijadwalkan tepat satu kali dalam seminggu, tidak boleh

ada kelas praktikum yang terjadwal lebih dari sekali dan ruangan yang digunakan harus bertipe laboratorium. Representasi matematisnya adalah:

$$\forall C_i \in Praktikum, \sum_{j=1}^n 1(C_i = C_j) = 1$$

$$\forall s_i \text{ dengan } C_i \text{ Praktikum, } R_i \in Laboratorium$$

Artinya, setiap kelas praktikum harus dijadwalkan di satu ruangan laboratorium dan hanya sekali.

- c. **Dua kelas praktikum yang sama harus dijadwalkan pada satu waktu yang sama dengan dua ruangan laboratorium berbeda.**

Ini artinya jika ada dua kelas yang merupakan kelas praktikum yang sama harus diberikan dua ruang laboratorium dalam satu sesi.

$$\forall (s_i, s_j), C_i = C_j \text{ dan } C_i \in Praktikum \rightarrow R_i \neq R_j$$

$S = \{s_1, s_2, \dots, s_n\}$ adalah himpunan jadwal yang dihasilkan

C_i = Kelas yang dijadwalkan

R_i = Ruangan yang digunakan

- d. **Setiap ruangan baik kelas maupun lab hanya boleh dijadwalkan untuk satu kelas pada satu sesi**

Constraint ini berarti setiap ruang hanya boleh ditempati oleh satu kelas pada sesi waktu yang sama.

$$\forall(s_i, s_j), R_i = R_j \text{ dan } T_i = T_j \rightarrow C_i = C_j$$

$S = \{s_1, s_2, \dots, s_n\}$ adalah himpunan jadwal yang dihasilkan

C_i = Kelas yang dijadwalkan

R_i = Ruangan yang digunakan

T_i = Waktu (hari dan sesi)

e. Tidak boleh ada perkuliahan di hari jumat sesi 3

Artinya pada hari Jumat, hanya sesi 1, 2, 4, dan 5 yang diperbolehkan.

$$\forall s_i, \quad 1(D_i = 5 \text{ dan } S_i = 3) = 0$$

$S = \{s_1, s_2, \dots, s_n\}$ adalah himpunan jadwal yang dihasilkan

C_i = Kelas yang dijadwalkan

R_i = Ruangan yang digunakan

T_i = Waktu (hari dan sesi)

D_i = Hari

S_i = Sesi

f. Setiap dosen hanya boleh mengajar satu kelas teori atau responsi di satu sesi yang sama

Constraint ini berarti seorang dosen tidak boleh memiliki dua jadwal mengajar di waktu yang sama untuk menghindari bentrok jadwal pengajaran. Representasi matematisnya adalah:

$$\forall(s_i, s_j), L_i = L_j \text{ dan } T_i = T_j \rightarrow C_i = C_j$$

$S = \{s_1, s_2, \dots, s_n\}$ adalah himpunan jadwal yang dihasilkan

C_i = Kelas yang dijadwalkan

R_i = Ruangan yang digunakan

$T_i = \text{Waktu (hari dan sesi)}$

$L_i = \text{Dosen}$

- g. Setiap dosen hanya boleh mengajar dua kelas praktikum di satu sesi yang sama.**

Dosen hanya boleh mengajar maksimal dua kelas praktikum yang sama dalam satu sesi.

$$\forall(L_k, T_l), \sum_{i=1}^n 1(L_i = L_k \text{ dan } T_i = T_l \text{ dan } P_i = \text{Praktikum}) \leq 2$$

$S = \{s_1, s_2, \dots, s_n\}$ adalah himpunan jadwal yang dihasilkan

$C_i = \text{Kelas yang dijadwalkan}$

$R_i = \text{Ruang yang digunakan}$

$T_i = \text{Waktu (hari dan sesi)}$

$L_i = \text{Dosen}$

$P_i = \text{Tipe Kelas}$

- h. Setiap dosen tidak boleh mengajar kelas responsi atau teori dan kelas praktikum secara bersamaan di satu sesi yang sama.**

Dosen tidak boleh mengajar teori/responsi dan praktikum dalam sesi yang sama. Dengan asumsi bahwa setiap kelas praktikum memiliki dua kelas paralel yang harus dijadwalkan bersamaan, maka jika seorang dosen mengajar kelas praktikum, dosen itu tidak bisa lagi mengajar kelas teori/responsi pada sesi yang sama.

$$\begin{aligned}
 & \forall(L_k, T_l), \sum_{i=1}^n 1(L_i = L_k \text{ dan } T_i = T_l \text{ dan } P_i \\
 & = \text{Teori/Responsi}) \cdot \sum_{j=1}^n 1(L_j = L_k \text{ dan } T_j \\
 & = T_l \text{ dan } P_j = \text{Praktikum}) = 0
 \end{aligned}$$

$S = \{s_1, s_2, \dots, s_n\}$ adalah himpunan jadwal yang dihasilkan

C_i = Kelas yang dijadwalkan

R_i = Ruangan yang digunakan

T_i = Waktu (hari dan sesi)

L_i = Dosen

P_i = Tipe Kelas (Teori/Responsi/Praktikum)

- i. **Setiap kelas hanya boleh memiliki satu jadwal kelas teori atau responsi di satu sesi yang sama.**

Artinya setiap kelas tidak boleh memiliki lebih dari satu kelas teori atau responsi dalam satu sesi.

$$\forall(C_k, T_l), \sum_{i=1}^n 1(C_i = C_k \text{ dan } T_i = T_l \text{ dan } P_i = \text{Teori/Responsi}) \leq 1$$

$S = \{s_1, s_2, \dots, s_n\}$ adalah himpunan jadwal yang dihasilkan

C_i = Kelas yang dijadwalkan

R_i = Ruangan yang digunakan

T_i = Waktu (hari dan sesi)

P_i = Tipe Kelas

- j. **Setiap kelas hanya boleh memiliki dua jadwal kelas praktikum di satu sesi yang sama.**

Artinya, jika suatu kelas memiliki lebih dari dua mata kuliah praktikum dalam satu sesi, maka terjadi pelanggaran.

$$\forall(C_k, T_l), \sum_{i=1}^n 1(C_i = C_k \text{ dan } T_i = T_l \text{ dan } P_i = \text{Praktikum}) \leq 2$$

$S = \{s_1, s_2, \dots, s_n\}$ adalah himpunan jadwal yang dihasilkan

C_i = Kelas yang dijadwalkan

R_i = Ruangan yang digunakan

T_i = Waktu (hari dan sesi)

P_i = Tipe Kelas

- k. Setiap kelas tidak boleh memiliki jadwal kelas responsi atau teori dan kelas praktikum secara bersamaan di satu sesi yang sama.**

Mahasiswa tidak boleh memiliki kelas teori/responsi dan praktikum pada waktu yang sama.

$$\begin{aligned} \forall(C_k, T_l), \sum_{i=1}^n 1(C_i = C_k \text{ dan } T_i = T_l \text{ dan } P_i \\ = \text{Teori/Responsi}) \cdot \sum_{j=1}^n 1(C_j = C_k \text{ dan } T_j \\ = T_l \text{ dan } P_j = \text{Praktikum}) = 0 \end{aligned}$$

$S = \{s_1, s_2, \dots, s_n\}$ adalah himpunan jadwal yang dihasilkan

C_i = Kelas yang dijadwalkan

R_i = Ruangan yang digunakan

T_i = Waktu (hari dan sesi)

P_i = Tipe Kelas (Teori/Responsi/Praktikum)

- I. Satu dosen hanya dapat mengajar maksimal 3 sesi dalam sehari.**

Seorang dosen tidak boleh dijadwalkan untuk mengajar lebih dari tiga sesi dalam satu hari. Ini berlaku untuk semua jenis kelas (teori, responsi, dan praktikum).

$$\forall(L_k, D_m), \sum_{i=1}^n 1(L_i = L_k \text{ dan } D_i = D_m) \leq 3$$

$S = \{s_1, s_2, \dots, s_n\}$ adalah himpunan jadwal yang dihasilkan
 $D_i = \text{Hari}$
 $L_i = \text{Dosen}$
 $S_i = \text{Sesi}$

3.3.3. Implementasi Algoritma Genetika

3.3.3.1. Inisialisasi Populasi

Tahap awal dalam algoritma genetika adalah inisialisasi populasi. Dalam konteks penjadwalan perkuliahan, populasi awal terdiri dari sejumlah solusi penjadwalan yang dibangkitkan secara acak. Setiap individu dalam populasi mewakili satu solusi penjadwalan, di mana gen-gen pada individu tersebut adalah komponen-komponen penjadwalan seperti waktu perkuliahan, dosen, mata kuliah, dan ruang.

a. Representasi kromosom

Setiap individu atau solusi (disebut kromosom) direpresentasikan dalam bentuk struktur data, seperti array atau list. Pada sistem ini, satu kromosom terdiri dari serangkaian gen yang merepresentasikan jadwal kuliah. Setiap kelas dan dosen (*classLecturer*) harus memiliki penjadwalan, yang terdiri dari Hari (*day_id*), Sesi (*session_id*), dan Ruangan (*room_id*). Sehingga

setiap kromosom direpresentasikan sebagai daftar dengan format [*day_1*, *session_1*, *room_1*, *day_2*, *session_2*, *room_2*, ..., *day_n*, *session_n*, *room_n*].

b. Ukuran Individu

Jumlah individu dalam satu populasi harus ditentukan pada tahap ini. Ukuran populasi yang besar bisa membantu mempertahankan keragaman solusi, tetapi akan memakan lebih banyak waktu komputasi. Pada sistem ini nantinya akan dibuat beberapa solusi individu dalam generasi awal menyesuaikan banyaknya data yang ada. Tiap individu dalam populasi dihasilkan dengan aturan berikut:

1. Menentukan hari (*day_id*)

Dimana hari akan dipilih secara acak dari daftar hari yang valid atau yang tersedia. Jika nantinya hari yang dipilih hari jumat, maka sesuai constraint yang ada, sesi ke 3 tidak boleh terpilih untuk menghasilkan jadwal.

2. Menentukan Sesi (*session_id*)

Di penentuan sesi ini, semua hari kecuali hari jumat sesinya dipilih secara acak dari semua sesi yang tersedia. Sedangkan jika memilih hari jumat, maka sesi yang dipilih hanya daftar sesi yang diperbolehkan.

3. Menentukan Ruangan (*room_id*)

Pemilihan ruangan akan tergantung pada jenis mata kuliahnya. Jika mata kuliahnya teori/responsi, maka akan dipilih ruangan teori, jika ruangannya praktikum, maka akan dicari pasangan lab yang kapasitasnya cukup atau lebih dari kapasitas kelas, dan jika tidak ada lagi ruangan yang tersedia, maka akan digunakan kelas online.

Inisialisasi populasi ini memastikan bahwa solusi awal memenuhi aturan dasar sebelum masuk ke proses selanjutnya.

3.3.3.2. Evaluasi *Fitness*

Setelah populasi awal dibentuk, setiap individu dievaluasi menggunakan fungsi *fitness*. Fungsi ini mengukur seberapa baik solusi penjadwalan memenuhi tujuan penelitian, misalnya meminimalkan konflik jadwal, memaksimalkan penggunaan sumber daya, dan memenuhi preferensi dosen.

- a. Fungsi *fitness* dapat dinyatakan sebagai fungsi yang mempertimbangkan berbagai kendala.
- b. *Fitness* yang lebih tinggi menunjukkan solusi yang lebih baik.

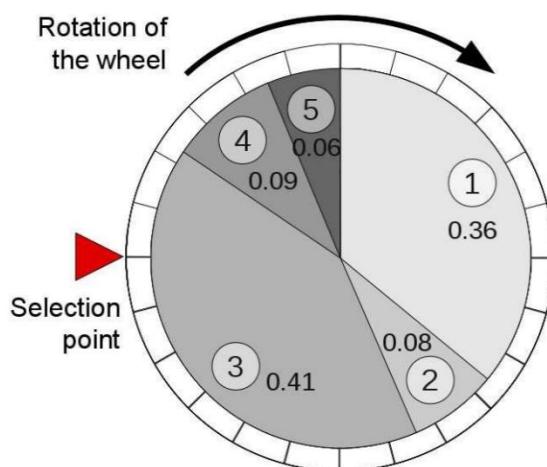
Pada awalnya setiap solusi akan diberikan nilai awal *fitness* sebesar 1000, kemudian penalti akan dikurangi dari nilai *fitness* jika terjadi pelanggaran aturan atau *constraint*. Penalti merupakan pengurangan nilai *fitness* yang diberikan jika suatu solusi (individu) melanggar aturan atau kendala dalam masalah yang sedang diselesaikan. Semakin tinggi nilai penalti, semakin rendah nilai *fitness*, karena penalti berbanding terbalik dengan *fitness*. Artinya, jika kromosom memiliki sedikit penalti, nilai *fitness* akan lebih tinggi, yang meningkatkan peluang kromosom tersebut terpilih untuk generasi berikutnya.

Kemudian kromosom diproses dalam kelipatan 3 karena tiap *entry* terdiri dari *day_id*, *session_id*, *room_id*. Tiap tiap *constraint* akan diberikan nilai penalty tergantung pada seberapa prioritas *constraint* tersebut. Kemudian nilai *fitness* akan dihitung berdasarkan penalti yang ada di tiap individu. Nilai *fitness* dihitung dalam rentang 0 hingga 1 dimana fitness yang baik

adalah *fitness* dengan nilai mendekati 1 atau sama dengan 1. Nilai *fitness* akan dihasilkan dari pengurangan *fitness* awal oleh jumlah penalti.

3.3.3.3. Seleksi

Tahap seleksi dalam algoritma genetika bertujuan untuk memilih individu dengan nilai *fitness* tinggi untuk bereproduksi dan menghasilkan generasi berikutnya. Seleksi memastikan bahwa gen-gen terbaik diwariskan, sementara individu dengan *fitness* rendah dieliminasi. Dalam penelitian ini nantinya hanya 4 individu yang terbaik yang dipilih sebagai parents yang ditandai dengan parameter *num_parents_mating*=10. PyGAD secara default menggunakan *Roulette Wheel Selection* untuk memilih individu terbaik sebagai metode seleksi. *Roulette Wheel Selection* dipilih karena proses seleksinya menggunakan probabilitas nilai *fitness* sehingga nilai *fitness* yang besar memiliki peluang lebih besar untuk terpilih (Setiawan et al., 2019).



Gambar 14. Roulette Wheel

Dalam *Roulette Wheel Selection*, probabilitas pemilihan individu sebanding dengan nilai *fitness*-nya. Metode ini menyerupai putaran roda

keberuntungan di mana setiap individu diberi bagian pada roda sesuai dengan *fitness*-nya. Proses dari *roulette wheel* ini adalah:

1. Setiap individu diberi proporsi sesuai nilai *fitness*-nya.
2. Sebuah roda roulette akan diputar, dan individu yang terpilih adalah berdasarkan probabilitas yang dihitung sebelumnya.
3. Proses ini diulang hingga jumlah individu yang dibutuhkan untuk reproduksi tercapai.

3.3.3.4. Crossover

Tahap *crossover* atau rekombinasi bertujuan untuk menghasilkan solusi baru dengan menggabungkan gen-gen dari dua orang tua (*parent*). *Crossover* adalah tahapan yang sangat penting dalam algoritma genetika karena terdapat proses persilangan gen antara 2 individu yang menghasilkan dua individu baru (*offspring*) pada generasi berikutnya. Metode *crossover* yang digunakan adalah *Uniform Crossover*.

Dari populasi yang telah diseleksi, 10 individu terbaik dipilih berdasarkan nilai *fitness*. Pasangan parents dipilih secara acak dari kelompok ini. Tahap selanjutnya adalah kombinasi gen *parents* yang dengan *uniform crossover* berarti tiap gen dalam anak memiliki probabilitas 50% untuk diwarisi dari salah satu orang tua. Contoh kombinasinya seperti ini:

Parent 1: [2, 5, 1, 3, 4, 6, 8, 9, 7]

Parent 2: [3, 4, 2, 1, 6, 7, 9, 5, 8]

Anak : [2, 4, 1, 1, 4, 7, 8, 5, 8]

Nantinya akan ada penggantian populasi dimana anak-anak yang baru dibuat akan menggantikan individu yang kurang fit dalam populasi. Akan ada parameter *keep_parents=2* yang akan mempertahankan dua orang tua terbaik yang membantu menjaga kualitas solusi dari generasi ke generasi.

3.3.3.5. Mutasi

Mutasi adalah proses pengubahan acak gen dalam individu untuk mencegah konvergensi prematur dan menjaga keragaman genetik. Mutasi dilakukan dengan probabilitas tertentu (biasanya kecil). Dalam konteks penjadwalan, mutasi dapat berarti mengganti waktu atau ruang kelas untuk satu mata kuliah secara acak. Dalam penelitian ini, mutasi terjadi berdasarkan konfigurasi *mutation_type* = "random" dan *mutation_percent_genes* = 15. 15 pada *mutation_percent_genes* didasarkan pada studi kasus penelitian ini, dimana termasuk kedalam masalah kompleks yang memungkinkan menghasilkan banyak kombinasi Solusi. Biasanya masalah ini menggunakan rentang 0% sampai 30% (Isa et al., 2024).

Penelitian ini menggunakan *random mutation*, yang berarti beberapa gen dalam kromosom dipilih secara acak dan nilainya diganti dengan nilai baru yang juga dipilih secara acak. Mutasi akan mengubah sebagian gen dalam solusi dengan nilai baru dari rentang yang telah ditentukan. Mutasi terjadi pada 15% gen, yang artinya 15% dari total gen dalam setiap individu akan mengalami perubahan. Jika dalam satu kromosom terdapat 200 gen, maka sekitar 30 gen akan mengalami mutasi.

3.3.3.6. Iterasi (Generasi)

Algoritma genetika akan mengulangi tahap seleksi, *crossover*, dan mutasi hingga tercapai kriteria terminasi. Kriteria ini berupa:

- a. Jumlah Generasi : Algoritma berhenti setelah sejumlah generasi tertentu.
- b. Konvergensi Solusi : Algoritma berhenti jika tidak ada perubahan signifikan dalam *fitness* individu terbaik setelah beberapa generasi.

Pada setiap iterasi, algoritma diharapkan menghasilkan populasi yang lebih baik dari sebelumnya. Solusi dengan *fitness* tertinggi pada akhir iterasi adalah solusi terbaik yang ditemukan.

3.3.4. Pengembangan *Web Service*

Tahapan pengembangan *web service* bertujuan untuk menciptakan layanan backend yang mendukung sistem penjadwalan perkuliahan berbasis algoritma genetika. Pengembangan *web service* menggunakan *framework* FastAPI karena mendukung pengembangan API RESTful yang cepat dan efisien. *Web service* ini menyediakan *endpoint* API yang dapat digunakan oleh aplikasi *frontend* atau sistem lain. *Web service* akan menerima request pembuatan jadwal dari aplikasi web melalui endpoint /api/generate-schedule pada port 8000. Kemudian data yang dibutuhkan akan diambil dari database melalui endpoint /api/public pada port 3000. Data tersebut akan diproses oleh model algoritma genetik dan menghasilkan jadwal dan mengirim jadwal tersebut kedalam format JSON ke database melalui endpoint /api/public/schedule pada port 3000 dan mengirim response ke aplikasi web.

3.3.5. Pengujian *Web Service*

Pengujian *web service* merupakan langkah krusial dalam proses pengembangan sistem penjadwalan perkuliahan otomatis untuk memastikan bahwa semua *endpoint* bekerja sesuai fungsinya dan dapat menangani berbagai skenario yang mungkin terjadi dalam penggunaannya. Pengujian ini dilakukan menggunakan Postman, sebuah alat yang memungkinkan pengujian langsung terhadap layanan web melalui permintaan HTTP. Pengujian mencakup verifikasi terhadap berbagai parameter yang dikirimkan melalui permintaan HTTP, seperti data kelas, jadwal, ruangan, dosen, serta parameter waktu yang menjadi bagian

penting dalam sistem penjadwalan. Tujuan utama pengujian ini adalah untuk memastikan bahwa setiap permintaan diproses dengan benar oleh *web service*, dan bahwa respon yang diberikan sesuai dengan yang diharapkan. Misalnya, untuk *endpoint* pembuatan jadwal, pengujian dilakukan untuk memastikan bahwa data yang dikirimkan disimpan dengan benar dalam basis data dan jadwal yang dihasilkan tidak bertentangan dengan jadwal yang sudah ada.

Tabel 10. Skenario Pengujian Web Service

Endpoint	Kasus Uji	Respon yang Diharapkan
/api/schedule	Mengirimkan data <i>ClassLecturer</i> , <i>ScheduleDay</i> , <i>ScheduleSession</i> , dan <i>Room</i> lengkap	Data jadwal perkuliahan yang telah diproses
/api/schedule	Mengirimkan data dengan data <i>ClassLecturer</i> kosong	Data kelas kosong; data tidak diproses menjadi jadwal
/api/schedule	Mengirimkan data dengan data <i>ScheduleDay</i> kosong	Data hari kosong; data tidak diproses menjadi jadwal
/api/schedule	Mengirimkan data dengan data <i>ScheduleSession</i> kosong	Data sesi kosong; data tidak diproses menjadi jadwal
/api/schedule	Mengirimkan data dengan data <i>Room</i> kosong	Data ruangan kosong; data tidak diproses menjadi jadwal

V. KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan hasil pengujian dan analisis yang telah dilakukan, dapat ditarik beberapa kesimpulan sebagai berikut:

1. Penelitian ini telah berhasil menerapkan Algoritma Genetika dengan library PYGAD dengan bahasa pemrograman Python untuk menyelesaikan masalah penjadwalan perkuliahan, dengan rata-rata waktu yang dibutuhkan untuk menghasilkan jadwal kuliah untuk satu jurusan di satu periode akademik kurang dari 3 menit.
2. Penelitian ini telah berhasil mengembangkan *web service* dengan *framework* FastAPI yang dapat mengintegrasikan model Algoritma Genetik yang telah dibangun.

5.2. Saran

Meskipun penelitian ini telah mencapai tujuan yang diharapkan, terdapat beberapa potensi pengembangan lebih lanjut yang dapat dipertimbangkan:

1. Mengembangkan model penjadwalan asisten untuk kelas praktikum dan responsi yang terintegrasi dengan model Algoritma Genetika yang sudah ada.
2. Mengembangkan mekanisme konfigurasi yang memungkinkan aturan dan parameter pada constraint dapat diubah secara fleksibel melalui file

3. konfigurasi, tanpa perlu melakukan perubahan langsung pada kode sumber. Hal ini akan meningkatkan fleksibilitas sistem dalam menyesuaikan kebijakan penjadwalan sesuai dengan kebutuhan yang berubah.
4. Mengembangkan sistem penjadwalan dengan antarmuka pengguna yang intuitif untuk diintegrasikan dengan model Algoritma Genetik melalui *web service* yang telah dikembangkan.
5. Menambahkan fitur tukar jadwal antara jadwal yang ada dengan jadwal kosong.

DAFTAR PUSTAKA

- Aliani, C., Rossi, E., Soliński, M., Francia, P., Lanatà, A., Buchner, T., & Bocchi, L. (2024). Genetic Algorithms for Feature Selection in the Classification of COVID-19 Patients. *Bioengineering*, 11(9). <https://doi.org/10.3390/bioengineering11090952>
- Alkafaween, E., Hassanat, A., Essa, E., & Elmougy, S. (2024). An Efficiency Boost for Genetic Algorithms: Initializing the GA with the Iterative Approximate Method for Optimizing the Traveling Salesman Problem—Experimental Insights. *Applied Sciences (Switzerland)*, 14(8). <https://doi.org/10.3390/app14083151>
- Almanjahie, I. M., & Ahmad, I. (2024). Enhancing probabilistic based real-coded crossover genetic algorithms with authentication of VIKOR multi-criteria optimization method. *AIMS Mathematics*. <https://www.aimspress.com/article/id/670e4984ba35de7eae907848>
- Amindoust, A., Asadpour, M., & ... (2021). A hybrid genetic algorithm for nurse scheduling problem considering the fatigue factor. *Journal of Healthcare* <https://doi.org/10.1155/2021/5563651>
- Ansari, R., & Saubari, N. (2020). Application of genetic algorithm concept on course scheduling. *IOP Conference Series: Materials Science and Engineering*, 821(1). <https://doi.org/10.1088/1757-899X/821/1/012043>
- Cahya Yustina, M., Nul Ichsan, I., & Muhammad Suranegara, G. (2024). Implementasi Algoritma Genetika Proses Mutasi Differential Evolution Pada Sistem Penjadwalan Mata Pelajaran. *Media Online*, 5(1), 116–130. <https://doi.org/10.30865/klik.v5i1.2109>
- Chen, M. C., Sze, S. N., Goh, S. L., Sabar, N. R., & Kendall, G. (2021). A Survey of University Course Timetabling Problem: Perspectives, Trends and Opportunities. *IEEE Access*, 9, 106515–106529. <https://doi.org/10.1109/ACCESS.2021.3100613>
- Chen, R., Su, L., & Guan, J. (2024). AI-Powered Genetic Algorithm for Optimal Scheduling in Medical Research and Educational Management. *Computer-Aided Design and Applications*, 21(S24), 17–34. <https://doi.org/10.14733/cadaps.2024.S24.17-34>

- Fachrie, M., & Fira Waluyo, A. (2020). *Guided Genetic Algorithm to Solve University Course Timetabling with Dynamic Time Slot*.
- Gad, A. F. (2021). *PyGAD: An Intuitive Genetic Algorithm Python Library*. <http://arxiv.org/abs/2106.06158>
- Hassanat, A., Almohammadi, K., Alkafaween, E., Abunawas, E., Hammouri, A., & Prasath, V. B. S. (2019a). Choosing mutation and crossover ratios for genetic algorithms-a review with a new dynamic approach. *Information (Switzerland)*, 10(12). <https://doi.org/10.3390/info10120390>
- Hassanat, A., Almohammadi, K., Alkafaween, E., Abunawas, E., Hammouri, A., & Prasath, V. B. S. (2019b). Choosing mutation and crossover ratios for genetic algorithms-a review with a new dynamic approach. *Information (Switzerland)*, 10(12). <https://doi.org/10.3390/info10120390>
- Hikmawati, F., Manajemen, J., Syariah, B., Ekonomi, F., Islam, B., & Pontianak, I. (2023). MANFAAT SISTEM INFORMASI AKADEMIK (SIAKAD) DALAM PERGURUAN TINGGI. *Jurnal Ilmiah Rekayasa Dan Manajemen Sistem Informasi*, 9(1), 45–51.
- Isa, F. M., Ariffin, W. N. M., Jusoh, M. S., & Putri, E. P. (2024). A Review of Genetic Algorithm: Operations and Applications. *Journal of Advanced Research in Applied Sciences and Engineering Technology*, 40(1), 1–34. <https://doi.org/10.37934/araset.40.1.134>
- Lambora, A., Gupta, K., & Chopra, K. (2019). Genetic algorithm-A literature review. *2019 International Conference* <https://ieeexplore.ieee.org/abstract/document/8862255/>
- Lata, S., Gobind Singh, G., Sohal, A., Lata Yadav, S., & Lata Yadav Asha Sohal Assistant Professor Associate Professor, S. (2017). Comparative Study of Different Selection Techniques in Genetic Algorithm Saneh Lata Yadav Comparative Study of Different Selection Techniques in Genetic Algorithm. In *International Journal of Engineering, Science and Mathematics* (Vol. 6). <https://www.researchgate.net/publication/325011628>
- Modibbo, U. M., Umar, I., Mijinyawa, M., Modibbo, U. M., Umar, I., Mijinyawa, M., & Hafisu, R. (2019). *Genetic Algorithm for Solving University Timetabling Problem*. www.amity.edu/ajcs
- Neumann, A., Laranjeiro, N., & Bernardino, J. (2021). An Analysis of Public REST Web Service APIs. *IEEE Transactions on Services Computing*, 14(4), 957–970. <https://doi.org/10.1109/TSC.2018.2847344>

- Noskova, E., Ulyantsev, V., Koepfli, K. P., O'brien, S. J., & Dobrynin, P. (2020). GADMA: Genetic algorithm for inferring demographic history of multiple populations from allele frequency spectrum data. *GigaScience*, 9(3). <https://doi.org/10.1093/gigascience/giaa005>
- Pătrăușanu, A., Florea, A., Neghină, M., Dicoiu, A., & Chiș, R. (2024). A Systematic Review of Multi-Objective Evolutionary Algorithms Optimization Frameworks. In *Processes* (Vol. 12, Issue 5). Multidisciplinary Digital Publishing Institute (MDPI). <https://doi.org/10.3390/pr12050869>
- Prambudi, A. P., Waluyo, A., & Fatich, E. V. L. N. (2021). Perancangan Sistem Penjadwalan Perkuliahian Berbasis Website Menggunakan Algoritma Genetika. *Teknik Informatika Dan Sistem Informasi*, 8, 1133–1146.
- Rohmad, A. N., & Akbar, M. (2024). Penerapan Algoritma Genetika Dalam Pengelompokan Mahasiswa KKN (Studi Kasus: KKN Angkatan XLII Universitas Mercu Buana Yogyakarta). *JIKO (Jurnal Informatika Dan Komputer)*, 8(1), 50. <https://doi.org/10.26798/jiko.v8i1.1073>
- Sánchez-Chica, A., Zulueta, E., Teso-Fz-Betoño, D., Martínez-Filgueira, P., & Fernandez-Gamiz, U. (2019). ANN-based stop criteria for a genetic algorithm applied to air impingement design. *Energies*, 13(1). <https://doi.org/10.3390/en13010016>
- Setiawan, J. Y., Dyah,), Herwindiati, E., & Sutrisno, T. (2019). *Jurnal Ilmu Komputer dan Sistem Informasi*.
- Singh, R. K., Panchal, V. K., & Singh, B. K. (2018). A review on genetic algorithm and its applications. *2018 Second International ...* <https://ieeexplore.ieee.org/abstract/document/8753030/>
- Sofiyah, S., Purnomo, B., Kunci, K., & Pegawai, A. (2019). ANALISIS PENERAPAN SISTEM INFORMASI AKADEMIK (SIAKAD) ONLINE DI UNIVERSITAS SULTAN AGENG TIRTAYASA. In *Business Journal: Jurnal Bisnis dan Sosial* (Vol. 05, Issue 02a). <http://journals.telkomuniversity.ac.id/business>
- Syswerda, G. (2018). *Uniform Crossover in Genetic Algorithms*.
- Taherdoost, H. (2023). Security and Internet of Things: Benefits, Challenges, and Future Perspectives. In *Electronics (Switzerland)* (Vol. 12, Issue 8). MDPI. <https://doi.org/10.3390/electronics12081901>
- Tragura, S. J. C. (2022). *Building Python Microservices with FastAPI: Build secure, scalable, and structured Python microservices from design concepts*

- to infrastructure. [ieeexplore.ieee.org.](https://ieeexplore.ieee.org/abstract/document/10163452/)
<https://ieeexplore.ieee.org/abstract/document/10163452/>
- Urva, G., & Sellyana, A. (2019). Genetic algorithm for optimization of lecturer schedule preparation. *Journal of Physics: Conference Series*, 1175(1).
<https://doi.org/10.1088/1742-6596/1175/1/012042>
- Vie, A., Kleinnijenhuis, A. M., & Farmer, D. J. (2020). *Qualities, challenges and future of genetic algorithms: a literature review*.
<http://arxiv.org/abs/2011.05277>
- Yang, T., & Zhou, Y. (2024). Analysis of Multi-Objective Evolutionary Algorithms On Fitness Function With Time-Linkage Property. *IEEE Transactions on Evolutionary* <https://ieeexplore.ieee.org/abstract/document/10454586/>