**IMPLEMENTASI ALGORITMA GENETIKA UNTUK MENGATASI KONFLIK JADWAL KRS MAHASISWA: STUDI KASUS PROGRAM STUDI TEKNIK INFORMATIKA, INSTITUT TEKNOLOGI SUMATERA**

**TUGAS AKHIR**

Diajukan sebagai syarat menyelesaikan jenjang strata Satu (S-1) di Program Studi Teknik Informatika, Fakultas Teknologi, Produksi dan Industri, Institut Teknologi Sumatera

**Oleh:**

**RAFELINA OCTA LADELAVIA**

**122140082**



**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS TEKNOLOGI, PRODUKSI DAN INDUSTRI**

**INSTITUT TEKNOLOGI SUMATERA**

**LAMPUNG SELATAN**

**2024**

# DAFTAR ISI

[DAFTAR ISI 2](#_heading=h.gjdgxs)

[BAB I 4](#_heading=h.30j0zll)

[PENDAHULUAN 4](#_heading=h.1fob9te)

[1.1 Latar Belakang Masalah 4](#_heading=h.3znysh7)

[1.2 Rumusan Masalah 5](#_heading=h.2et92p0)

[1.3 Tujuan Penelitian 5](#_heading=h.tyjcwt)

[1.4 Batasan Masalah 6](#_heading=h.3dy6vkm)

[1.5 Manfaat Penelitian 7](#_heading=h.ii1zi0fg2agu)

[1.6 Korelasi RP – RQ – RO 7](#_heading=h.v42q8gp9ji25)

[1.7 Kontribusi Penelitian 9](#_heading=h.yy0mbw74rjnh)

[1.7.1 Kontribusi Teoretis 9](#_heading=h.8sa08mtuq40c)

[1.7.2 Kontribusi Metodologis 9](#_heading=h.uhrkdq1hyeps)

[1.7.3 Kontribusi Praktis 9](#_heading=h.a7qfusrxzmd5)

[1.7.4 Kontribusi Aplikatif 10](#_heading=h.i3ej3pk2sb)

[1.7.5 Kontribusi Terhadap Kebijakan Akademik 10](#_heading=h.4qtziapnypjm)

[1.8 Sistematika Penulisan 10](#_heading=)

[1.8.1 BAB I 10](#_heading=h.1q18tabcxmxp)

[1.8.2 BAB II 11](#_heading=h.v4q1lgdu4z8q)

[1.8.3 BAB III 11](#_heading=h.p1bs852hj7ly)

[1.8.4 BAB IV 11](#_heading=h.clgi8julnt3x)

[1.8.5 BAB V 11](#_heading=h.xhw2gic4ibw6)

[BAB II 12](#_heading=h.3rdcrjn)

[TINJAUAN PUSTAKA 12](#_heading=h.26in1rg)

[2.1 Tinjauan Pustaka 12](#_heading=h.lnxbz9)

[2.2 Dasar Teori 14](#_heading=h.35nkun2)

[2.2.1 Penjadwalan KRS Mahasiswa 14](#_heading=h.1ksv4uv)

[2.2.4 Algoritma Genetika 16](#_heading=h.44sinio)

[2.2.5 Masalah Penjadwalan 16](#_heading=h.2jxsxqh)

[2.2.6 Pembelajaran Mesin 17](#_heading=h.z337ya)

[2.2.7 Optimasi Penjadwalan Kuliah 17](#_heading=h.3j2qqm3)

[2.2.8 Pengaruh Kompleksitas Penjadwalan Kuliah Terhadap Kinerja Algoritma Genetika 18](#_heading=h.1y810tw)

[2.2.9 Pembelajaran Mesin untuk Prediksi Durasi dan Prioritas Penjadwalan Kuliah 18](#_heading=h.4i7ojhp)

[2.2.10 Implementasi Algoritma Genetika dalam Penjadwalan Jadwal Kuliah 18](#_heading=h.2xcytpi)

[2.2.11 Evaluasi Model dalam Optimasi Penjadwalan Kuliah Menggunakan Algoritma Genetika 19](#_heading=h.b4aaqv78dk4h)

[BAB III 20](#_heading=h.czvist6gc3nx)

[METODE PENELITIAN 20](#_heading=h.ymm96p787v8n)

[3.1 Desain Penelitian 20](#_heading=h.nkauehhhv10v)

[3.2 Pengumpulan Data (Data Gathering) 23](#_heading=h.dfo0j7za1g10)

[3.2.1 Data yang Digunakan 23](#_heading=h.crlqg86223o9)

[3.2.2 Pemilihan Data 24](#_heading=h.q6jqqegjz0ip)

[3.2.3 Format Data 24](#_heading=h.at94i1iz2f4a)

[3.3 Pengolahan Awal Data (Data Pre-processing) 26](#_heading=h.wn7vtknf7ee)

[3.3.1 Validasi Data 26](#_heading=h.rfneo1t2du15)

[3.3.2 Seleksi Data 27](#_heading=h.z4g1bs4ktg2h)

[3.3.3 Transformasi Data 27](#_heading=h.920gwrza81y9)

[3.3.4 Penyimpanan Data 27](#_heading=h.lapke0lpbs74)

[3.4 Model yang Diusulkan (Proposed Model) 27](#_heading=h.hyxu6sf905rs)

[3.4.1 Representasi Data 28](#_heading=h.btn3o4hpiinc)

[3.4.2 Fungsi Fitness 28](#_heading=h.y12eyc8nitls)

[3.4.3 Proses Algoritma Genetika 29](#_heading=h.vbi8m0bgsdh0)

[3.5 Eksperimen dan Pengujian Model 30](#_heading=h.olrpmijyemu6)

[3.5.1 Parameter Eksperimen 30](#_heading=h.c9bzklqhhx8f)

[3.5.2 Proses Eskperimen 32](#_heading=h.fb09t8npewow)

[3.5.3 Hasil Eksperimen 39](#_heading=h.b58zzomydbh2)

[3.6 Evaluasi dan Validasi Hasil (Result Evaluation and Validation) 40](#_heading=h.4j37vsafpwk1)

[3.6.1 Evaluasi Hasil 40](#_heading=h.duf8g7anr3bm)

[3.6.2 Validasi Hasil 41](#_heading=h.tljg71cxnbir)

[3.6.3 Analisis Hasil 41](#_heading=h.42d388vyl7al)

[3.6.4 Kesimpulan Evaluasi dan Validasi 42](#_heading=h.f6shwbb2zubz)

[DAFTAR PUSTAKA 43](#_heading=h.1ci93xb)

# BAB I

# PENDAHULUAN

## Latar Belakang Masalah

Penjadwalan kuliah merupakan salah satu tantangan besar yang dihadapi institusi pendidikan tinggi. Kompleksitasnya terletak pada kebutuhan untuk mengalokasikan waktu, ruang, dan mata kuliah secara efisien sambil mempertimbangkan berbagai kendala, seperti jumlah mahasiswa, kapasitas ruangan, dan preferensi dosen. Konflik yang sering terjadi meliputi tabrakan jadwal mata kuliah, benturan penggunaan ruang, serta jadwal yang terlalu padat bagi mahasiswa maupun dosen. Ketidakmampuan untuk mengatasi konflik ini dapat mengakibatkan ketidakefisienan proses belajar mengajar, penurunan produktivitas, dan ketidakpuasan seluruh pihak yang terlibat [1].

Permasalahan konflik dalam penentuan KRS tidak hanya berdampak pada mahasiswa, tetapi juga memengaruhi dosen dan pihak pengelola akademik. Bagi mahasiswa, konflik jadwal dapat menyebabkan penurunan efektivitas proses pembelajaran, keterbatasan dalam mengikuti mata kuliah yang diperlukan, serta peningkatan tingkat stres [2]. Di sisi lain, benturan alokasi ruang dan waktu yang tidak efisien menghambat optimalisasi sumber daya kampus, seperti ruang kelas dan tenaga pengajar. Dalam skala yang lebih luas, konflik ini dapat menurunkan kualitas pelayanan akademik yang diberikan oleh institusi pendidikan. [2]

Algoritma genetika merupakan salah satu pendekatan yang efektif dalam menangani masalah penjadwalan yang kompleks. Sebagai algoritma optimasi berbasis evolusi, algoritma genetika mampu mengeksplorasi ruang solusi secara luas dan menghasilkan jadwal yang mendekati optimal dengan mempertimbangkan berbagai kendala. Dengan menggunakan algoritma genetika, proses penentuan KRS dapat disesuaikan untuk mengurangi tabrakan jadwal mata kuliah, memastikan alokasi ruang yang tepat, serta menghasilkan jadwal yang seimbang dan efisien bagi mahasiswa. [3]

Melalui penelitian ini, difokuskan pada implementasi algoritma genetika untuk mengatasi konflik jadwal dalam proses penentuan KRS mahasiswa. Pendekatan ini bertujuan untuk membantu mahasiswa mendapatkan jadwal kuliah yang optimal dengan meminimalkan tabrakan jadwal mata kuliah, mengoptimalkan alokasi ruang, dan memastikan distribusi waktu yang efisien. [4] Dengan memanfaatkan algoritma genetika, penelitian ini diharapkan dapat memberikan solusi praktis yang mendukung peningkatan efektivitas proses akademik dan membantu institusi pendidikan dalam mengelola jadwal secara lebih terstruktur.

## Rumusan Masalah

1. Bagaimana algoritma genetika dapat diimplementasikan untuk mengoptimalkan penjadwalan KRS mahasiswa dengan mempertimbangkan faktor seperti jumlah mata kuliah, durasi kuliah, alokasi ruang, dan preferensi waktu mahasiswa?
2. Bagaimana pengaruh parameter dalam algoritma genetika, seperti seleksi, crossover, mutasi, dan fitness function, terhadap kemampuan algoritma dalam menghasilkan jadwal KRS yang optimal dan efisien?
3. Seberapa efektif algoritma genetika dalam mengurangi konflik jadwal pada proses penentuan KRS mahasiswa, seperti tabrakan mata kuliah, benturan penggunaan ruang, dan distribusi jadwal yang tidak seimbang?

## Tujuan Penelitian

1. Menganalisis penerapan algoritma genetika dalam proses optimasi penjadwalan KRS mahasiswa dengan mempertimbangkan faktor-faktor utama seperti jumlah mata kuliah, durasi kuliah, kapasitas ruang, dan preferensi waktu.
2. Mengevaluasi kemampuan algoritma genetika dalam mengurangi konflik penjadwalan KRS mahasiswa, seperti tabrakan jadwal mata kuliah, benturan alokasi ruang, dan distribusi waktu yang tidak efisien.
3. Mengidentifikasi efektivitas algoritma genetika dalam menghasilkan jadwal KRS yang optimal dengan mengutamakan efisiensi penggunaan waktu, ruang, dan sumber daya kampus.
4. Mengkaji pengaruh parameter algoritma genetika, termasuk proses seleksi, crossover, mutasi, dan fitness function, terhadap performa algoritma dalam menghasilkan jadwal KRS yang lebih terstruktur dan adaptif.

## Batasan Masalah

1. Penelitian ini difokuskan pada implementasi algoritma genetika untuk mengoptimalkan penjadwalan KRS mahasiswa, tanpa melibatkan pengembangan sistem atau aplikasi perangkat lunak secara langsung.
2. Analisisi hanya mencakup parameter utama dalam penjadwalan KRS, seperti jumlah mata kuliah, durasi kuliah, alokasi ruang, dan waktu pelaksanaan kuliah, dengan menggunakan data simulasi yang mencerminkan kondisi aktual.
3. Fokus penelitian terbatas pada optimasi jadwal akademik mahasiswa di tingkat perguruan tinggi, tanpa mencakup jadwal kegiatan non-akademik, seperti organisasi atau aktivitas pribadi mahasiswa.
4. Konflik yang dioptimalkan dalam penelitian ini meliputi tabrakan jadwal mata kuliah pada waktu dan ruang yang sama, ketidakefisienan alokasi ruang berdasarkan kapasitas ruangan, jadwal yang terlalu padat atau memiliki waktu kosong yang tidak efisien.
5. Evaluasi algoritma genetika dilakukan berdasarkan simulasi data jadwal kuliah menggunakan fitness function yang telah ditentukan, tanpa melibatkan data real-time dari institusi pendidikan.
6. Penelitian ini tidak mencakup analisis finansial atau sumber daya manusia yang terkait dengan pelaksanaan jadwal kuliah.

## Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan manfaat dalam tiga aspek utama: teoritis, praktis, dan aplikatif. Secara teoritis, penelitian ini bertujuan untuk memperkaya wawasan di bidang artificial intelligence, khususnya dalam penerapan algoritma genetika sebagai metode optimasi untuk mengatasi konflik penjadwalan yang kompleks. Penelitian ini juga memberikan kontribusi ilmiah melalui analisis mendalam terhadap parameter algoritma genetika, seperti seleksi, crossover, mutasi, dan fitness function, dalam konteks penjadwalan KRS mahasiswa. [5]

Secara praktis, penelitian ini diharapkan dapat membantu institusi pendidikan tinggi, khususnya dalam mengelola penjadwalan KRS mahasiswa dengan lebih efisien. Algoritma genetika dapat menjadi solusi untuk meminimalkan konflik jadwal seperti tabrakan mata kuliah, benturan alokasi ruang, dan distribusi jadwal yang tidak seimbang. Selain itu, penelitian ini juga menawarkan pendekatan berbasis teknologi yang lebih efektif dibandingkan metode tradisional dalam mengelola jadwal akademik. [6]

Adapun manfaat aplikatifnya, penelitian ini dapat menjadi acuan untuk pengembangan lebih lanjut, terutama dalam mengintegrasikan algoritma genetika dengan data real-time untuk menghasilkan penjadwalan KRS yang lebih adaptif dan dinamis. Selain itu, pendekatan ini dapat diaplikasikan pada berbagai bidang lain yang menghadapi tantangan serupa, seperti penjadwalan ujian, pengelolaan jadwal kerja, atau optimasi logistik dalam transportasi.

## 

## Korelasi RP – RQ – RO

Penelitian ini berfokus pada korelasi antara tiga elemen utama dalam algoritma genetika untuk optimasi penjadwalan KRS mahasiswa: *Random Parameters* (RP), *Random Quality* (RQ), dan *Result Optimization* (RO). Korelasi ini memberikan gambaran tentang bagaimana parameter awal algoritma memengaruhi kualitas solusi dan hasil akhir optimasi.

1. **RP (Random Parameters)**Merupakan parameter acak seperti inisialisasi populasi, ukuran populasi, probabilitas *crossover* (*crossover rate*), dan probabilitas mutasi (*mutation rate*). Parameter ini menentukan keberagaman individu dalam populasi awal, yang memengaruhi kemampuan algoritma untuk mengeksplorasi ruang solusi.
2. **RQ (Random Quality)**Kualitas solusi awal yang dihasilkan, diukur melalui nilai *fitness*. Nilai *fitness* mengukur sejauh mana solusi memenuhi kriteria seperti minimasi konflik jadwal, efisiensi waktu, dan distribusi beban SKS.
3. **RO (Result Optimization)**Merupakan tingkat keberhasilan algoritma dalam menghasilkan solusi optimal setelah iterasi, mencakup pengurangan konflik jadwal, optimalisasi total SKS, dan pemenuhan batasan seperti kapasitas ruang dan preferensi waktu dosen.

Korelasi Antar Faktor

1. **RP → RQ:** Keberagaman parameter acak secara langsung meningkatkan peluang individu dalam populasi awal memiliki nilai *fitness* yang tinggi.
2. **RQ → RO:** Solusi awal yang berkualitas tinggi memungkinkan algoritma mempercepat konvergensi menuju solusi optimal.
3. **RP → RO:** Parameter acak yang tepat mengurangi waktu komputasi sekaligus menghasilkan hasil yang lebih optimal.

Hasil simulasi menunjukkan bahwa pengaturan parameter memengaruhi keberhasilan algoritma genetika. Ringkasan hasil simulasi disajikan pada tabel berikut:

| **Parameter** | **Hasil Simulasi** |
| --- | --- |
| Ukuran populasi | Optimal pada kisaran 100-200 individu untuk menjaga keseimbangan antara keberagaman dan efisiensi waktu. |
| Probabilitas *crossover* | 0,5 memberikan hasil terbaik dalam mencapai nilai *fitness* maksimum dalam iterasi lebih sedikit. |
| Probabilitas mutasi | 0,2 menjaga keberagaman populasi dan mencegah stagnasi solusi. |
| Populasi awal | Keberagaman tinggi menghasilkan nilai *fitness* rata-rata yang lebih baik. |

## Kontribusi Penelitian

### **Kontribusi Teoretis**

Penelitian ini memperkaya literatur dalam bidang optimasi dan artificial intelligence, khususnya penerapan algoritma genetika untuk menyelesaikan masalah penjadwalan yang kompleks. Analisis yang mendalam terhadap parameter algoritma genetika, seperti seleksi, crossover, mutasi, dan fitness function, memberikan wawasan baru tentang bagaimana parameter ini memengaruhi kinerja algoritma dalam menghasilkan solusi optimal.

### **Kontribusi Metodologis**

Penelitian ini menyajikan metode sistematis untuk mengimplementasikan algoritma genetika dalam mengatasi konflik penjadwalan KRS mahasiswa. Pendekatan ini mencakup representasi data yang komprehensif, pengembangan fungsi fitness yang relevan dengan masalah, serta validasi hasil yang memastikan solusi tidak hanya optimal tetapi juga sesuai dengan batasan akademik.

### **Kontribusi Praktis**

Hasil penelitian ini dapat diadopsi oleh institusi pendidikan tinggi untuk meningkatkan efisiensi dalam pengelolaan jadwal akademik. Dengan menggunakan algoritma genetika, institusi dapat meminimalkan konflik jadwal seperti bentrokan antar mata kuliah, benturan alokasi ruang, serta distribusi waktu yang tidak efisien. Hal ini berkontribusi langsung pada peningkatan kualitas layanan akademik.

### **Kontribusi Aplikatif**

Penelitian ini memberikan dasar yang kuat untuk pengembangan sistem berbasis algoritma genetika yang dapat digunakan secara real-time. Selain itu, pendekatan ini dapat diterapkan pada berbagai konteks lain di luar pendidikan, seperti pengelolaan jadwal kerja, optimasi logistik, atau sistem alokasi sumber daya yang kompleks.

### **Kontribusi Terhadap Kebijakan Akademik**

Dengan menyediakan solusi penjadwalan yang lebih adaptif dan terstruktur, penelitian ini berpotensi mendukung pengambilan keputusan strategis dalam penyusunan kebijakan akademik terkait penjadwalan perkuliahan. Pendekatan yang diusulkan dapat membantu meningkatkan produktivitas mahasiswa dan dosen, sekaligus mengoptimalkan penggunaan sumber daya kampus.

## 1.8 Sistematika Penulisan

### **BAB I**

Bab pertama berisi latar belakang masalah yang mendasari penelitian, rumusan masalah, tujuan penelitian, batasan masalah, manfaat penelitian, dan diakhiri dengan sistematika penulisan yang menjelaskan isi dari masing-masing bab dalam dokumen ini.

### **BAB II**

Bab kedua membahas tinjauan pustaka, termasuk teori-teori yang relevan, hasil penelitian terdahulu, serta kerangka berpikir yang menjadi landasan penelitian ini.

### **BAB III**

Bab ketiga membahas metode penelitian yang mencakup desain penelitian, pengumpulan data, serta langkah-langkah implementasi algoritma genetika untuk mengoptimalkan penjadwalan KRS mahasiswa.

### **BAB IV**

Bab keempat menyajikan hasil dan pembahasan, termasuk analisis hasil eksperimen, evaluasi efektivitas algoritma, serta pembahasan mendalam tentang dampak parameter algoritma terhadap performa jadwal yang dihasilkan.

### **BAB V**

Bab terakhir berisi kesimpulan yang merangkum temuan utama penelitian dan saran untuk pengembangan lebih lanjut, baik dalam konteks penelitian maupun implementasi praktis.

# BAB II

# TINJAUAN PUSTAKA

## Tinjauan Pustaka

| No | [Penulis][Judul][Tahun] | Permasalahan dan Tujuan | Metode | Kontribusi | Perbedaan |
| --- | --- | --- | --- | --- | --- |
| 1 | Kaili Shao , Hui Fu, Bo Wang [2023] [An Efficient Combination of Genetic Algorithm and Particle Swarm Optimization for Scheduling Data-Intensive Tasks in HeterogeneousCloud Computing] [3] | Optimasi penjadwalan tugas data-intensif di lingkungan komputasi awan yang heterogen, yang merupakan tantangan karena adanya kompleksitas dalam alokasi sumber daya | Kombinasi antara algoritma genetika (GA) dan optimasi swarm partikel (PSO) | Pengembangan metode kombinasi algoritma genetika dan optimasi swarm partikel untuk penjadwalan tugas di komputasi awan heterogen. | Jurnal ini membahas penggabungan algoritma genetika dengan particle swarm optimization untuk penjadwalan tugas yang memerlukan banyak data di cloud computing. |
| 2 | Seungmin Oh, Junchul Yoon, Yoona Choi, Young-Ae Jung, Jinsul Kim [2022] [Genetic Algorithm for the Optimization of a Building Power Consumption Prediction Model] [4] | mengidentifikasi tantangan dalam memprediksi konsumsi daya bangunan dengan akurasi yang memadai di tengah meningkatnya permintaan energi. Tujuannya adalah mengembangkan model prediksi yang lebih akurat menggunakan algoritma genetika untuk optimasi fitur, guna meningkatkan efisiensi dalam pengelolaan konsumsi energi bangunan. | Algoritma genetika | penyediaan model prediksi energi yang lebih akurat dan efisien, serta penggunaan algoritma genetika untuk meningkatkan proses pemilihan fitur, yang dapat mendukung upaya keberlanjutan dalam manajemen energi bangunan. | Penelitian ini lebih berorientasi pada pengelolaan energi bangunan, dengan tujuan efisiensi energi. |
| 3 | Luh Putu Sri Ardiyani[2022][Pe rbandingan Algoritma Genetika dengan Algoritma Steepest Ascent Hill Climbing untuk Optimasi Penjadwalan Kuliah] [5] | Bagaiman mengoptimalkan penjadwalan kuliah secara otomatis untuk mengurangi konflik jadwal dan memaksimalkan efisiensi, untuk mengevaluasi kinerja kedua algoritma tersebut dalam menghasilkan solusi optimal pada penjadwalan kuliah | Algoritma genetika dan Steepest Ascent Hill Climbing | Pengembangan metode untuk membandingkan algoritma genetika dengan steepest ascent hill climbing dalam mengoptimalkan penjadwalan kuliah. | Jurnal ini membahas perbandingan efektivitas algoritma genetika dan algoritma steepest ascent hill climbing dalam penjadwalan kuliah di universitas, dengan penekanan pada kecepatan dan efisiensi waktu eksekusi. |
| 4 | Ryan Eka Wiratna,  Afina Lina Nurlaili,  Agung Mustika  Rizki  [2023][Pembuata n Aplikasi  Penjadwalan Mata  Kuliah  Menggunakan  Algoritma  Genetika] [6] | Mempermudah menenentukan rute terpendek tim pemasaran kampus ketika mendatangi berbagai sekolah dalam sehari | Algoritma genetika | Pengembangan  aplikasi  penjadwalan mata kuliah yang lebih efisien  menggunakan algoritma genetika | Jurnal ini fokus pada penjadwalan mata kuliah di perguruan tinggi dengan algoritma genetika untuk menghindari bentrokan jadwal dan meningkatkan efisiensi pengelolaan perkuliahan. |
| 5 | Isnaini  Muhandhis,  M. Shubhan,  Hisyam Ib Dani,  Arkila Rakasyah, Alven S. Ritonga,  Mamik Usniyah  Sari [2023] [Pencarian  Rute Terpendek  Tim Promosi  Kampus dengan  Menggunakan  Algoritma  Genetik] [7] | Mencari rute terpendek untuk tim promosi kampus yang harus mengunjungi berbagai titik. dengan tujuan mengoptimalkan jarak tempuh dan waktu perjalanan agar rute yang diambil lebih efisien. | Algoritma Genetika | Memberikan solusi praktis untuk mengoptimalkan jarak dan waktu perjalanan, serta meningkatkan pemahaman  tentang penggunaan algoritma genetika dalam optimisasi rute. | Jurnal ini fokus kepada pencarian rute terpendek tim promosi kampus menggunakan algoritma genetika, termasuk penerapan Travelling Salesman Problem (TSP) untuk optimasi rute. |

## Dasar Teori

### **Penjadwalan KRS Mahasiswa**

Kartu Rencana Studi (KRS) merupakan sistem pemilihan mata kuliah yang dilakukan oleh mahasiswa di setiap awal semester untuk menentukan jadwal perkuliahan yang akan diikuti. Proses penjadwalan KRS yang ideal harus memastikan bahwa setiap mahasiswa dapat memilih mata kuliah sesuai kebutuhan tanpa adanya konflik jadwal, seperti bentrok antar mata kuliah atau keterbatasan kapasitas ruang kuliah. [6] Namun, proses ini sering kali menghadapi berbagai kendala, termasuk tingginya jumlah mahasiswa, kompleksitas jadwal dosen, dan alokasi ruangan yang terbatas.

1. **Jadwal Kuliah**

Jadwal kuliah adalah pengaturan waktu dan alokasi kegiatan perkuliahan yang mencakup mata kuliah, waktu pelaksanaan, lokasi, serta dosen pengampu dalam suatu periode akademik tertentu. Jadwal ini dirancang untuk memastikan kelancaran proses pembelajaran di perguruan tinggi dengan mempertimbangkan berbagai aspek, seperti kurikulum, kapasitas ruang kuliah, serta kebutuhan mahasiswa dan dosen. [7] Jadwal kuliah berfungsi sebagai panduan bagi mahasiswa untuk mengatur waktu mereka dalam mengikuti kegiatan akademik, sehingga membantu menciptakan keteraturan, efisiensi, dan distribusi beban belajar yang seimbang.

1. **Manajemen Waktu dalam Penjadwalan Kuliah**

Manajemen waktu dalam penjadwalan kuliah adalah proses pengaturan waktu yang efisien untuk menyelesaikan berbagai kegiatan akademik, seperti kuliah, tugas, ujian, dan aktivitas lainnya. Dengan keterampilan manajemen waktu yang baik, mahasiswa dapat memprioritaskan tugas-tugas yang lebih mendesak dan penting, serta menghindari prokrastinasi. [8] Kemampuan untuk mengatur waktu dengan bijak memungkinkan mahasiswa untuk fokus pada mata kuliah yang membutuhkan perhatian lebih, sembari tetap menjaga keseimbangan antara kehidupan pribadi dan akademik. Dengan menggunakan teknik seperti to-do list, pengaturan waktu per hari, atau aplikasi manajemen waktu, mahasiswa dapat meningkatkan produktivitas, mengurangi stres, dan memaksimalkan hasil akademik mereka.

### **Algoritma Genetika**

Seleksi alam, mutasi, rekombinasi, dan proses evolusi biologis lainnya menjadi inspirasi bagi Algoritma Genetika (GA), yang merupakan teknik optimasi. GA biasanya digunakan untuk mengatasi permasalahan rumit dengan mengoptimalkan solusi melalui analisis kebugaran setiap anggota dalam populasi. [9] Menemukan solusi optimal dengan efisiensi tinggi dan menjelajahi area solusi yang luas adalah dua manfaat GA. GA menunjukkan harapan yang signifikan dalam meningkatkan kinerja algoritma dalam menyelesaikan masalah rumit di berbagai aplikasi.

### **Masalah Penjadwalan**

Masalah penjadwalan adalah tantangan dalam mengalokasikan sumber daya (waktu, tenaga kerja, mesin) untuk menyelesaikan serangkaian tugas secara efisien. Masalah ini dapat ditemukan dalam berbagai bidang, seperti industri manufaktur, manajemen proyek, pendidikan, dan transportasi. [10] Beberapa jenis masalah penjadwalan yang umum meliputi:

* **Penjadwalan Satu Mesin**: Menentukan urutan optimal untuk memproses tugas pada satu mesin, dengan tujuan meminimalkan waktu penyelesaian atau keterlambatan.
* **Penjadwalan Multi-Mesin**: Melibatkan alokasi tugas ke beberapa mesin secara bersamaan, dengan memperhitungkan ketergantungan antar tugas dan waktu pemrosesan.
* **Penjadwalan Proyek**: Penjadwalan aktivitas yang terlibat dalam proyek, seperti proyek konstruksi atau pengembangan perangkat lunak, di mana tugas-tugas memiliki ketergantungan tertentu dan batasan waktu yang ketat.

Penjadwalan seringkali menghadapi tantangan besar, terutama jika ada banyak tugas yang saling bergantung dan harus diselesaikan dalam waktu terbatas. Masalah penjadwalan dikenal sebagai masalah *NP-Hard*, yang menunjukkan bahwa tidak ada algoritma yang mampu menemukan solusi optimal dengan cepat untuk setiap kasus. [11] Oleh karena itu, metode optimasi, seperti algoritma genetika, sering digunakan untuk mencari solusi yang mendekati optimal dalam waktu yang wajar.

### **Pembelajaran Mesin**

Algoritma yang memungkinkan komputer untuk mempelajari pola data tanpa harus deprogram secara langsung untuk menjalankan tugas tertentu disebut algoritma pembelajaran mesin (machine learning algorithms). Algoritma ini digunakan untuk menganalisis data, mengenali pola, dan membuat prediksi atau keputusan tanpa perlu diprogram secara eksplisit untuk setiap tugas. Metode Machine Learning dibagi menjadi tiga kategori utama, yaitu pembelajaran terawasi, pembelajaran tanpa pengawasan, dan pembelajaran penguatan. [12]

Dalam pembelajaran terawasi, model dilatih menggunakan dataset yang sudah memiliki label, sehingga model dapat memprediksi hasil dari data baru berdasarkan pola yang dipelajari. Dalam konteks penjadwalan, pembelajaran mesin dapat membantu dalam memprediksi durasi tugas dan menentukan prioritas berdasarkan pola data historis, yang memungkinkan sistem penjadwalan untuk lebih adaptif dan efisien. [13]

### **Optimasi Penjadwalan Kuliah**

Optimasi penjadwalan kuliah adalah proses yang bertujuan untuk merencanakan dan mengatur jadwal perkuliahan secara efisien, sehingga dapat memaksimalkan penggunaan waktu, ruang, dan sumber daya yang tersedia. Dalam proses ini, berbagai algoritma, termasuk algoritma genetika, sering digunakan untuk menemukan solusi optimal dalam menangani kompleksitas jadwal. Optimasi ini melibatkan analisis waktu kuliah, alokasi ruang, jadwal dosen, dan distribusi beban kuliah mahasiswa. Proses ini penting untuk menghindari konflik jadwal, seperti bentrokan waktu antar mata kuliah, serta untuk memastikan kelancaran proses akademik. Dengan menerapkan strategi yang tepat, optimasi penjadwalan kuliah dapat menghasilkan jadwal yang terorganisir, fleksibel, dan memenuhi kebutuhan semua pihak terkait. [14]

### **Pengaruh Kompleksitas Penjadwalan Kuliah Terhadap Kinerja Algoritma Genetika**

Kompleksitas penjadwalan kuliah, yang meliputi jumlah mata kuliah, durasi perkuliahan, dan ketergantungan antar mata kuliah, dapat mempengaruhi kinerja algoritma genetika secara signifikan. Penjadwalan yang lebih kompleks, dengan banyak variabel dan batasan, dapat memperlambat proses pencarian solusi optimal, memerlukan waktu dan sumber daya yang lebih banyak. [15] Algoritma genetika terbukti efektif dalam menyelesaikan masalah penjadwalan kuliah dengan kompleksitas tinggi, terutama dalam mengatur waktu dan ruang kuliah yang terbatas, serta alokasi dosen, yang memerlukan efisiensi dalam penentuan jadwal. Algoritma ini mampu mengeksplorasi berbagai solusi untuk menemukan penjadwalan yang optimal meskipun menghadapi tantangan yang kompleks. [16]

### **Pembelajaran Mesin untuk Prediksi Durasi dan Prioritas Penjadwalan Kuliah**

Pembelajaran mesin (machine learning) dapat diterapkan untuk meningkatkan akurasi penjadwalan kuliah dengan memprediksi durasi dan prioritas mata kuliah secara otomatis. Teknik ini memungkinkan sistem penjadwalan untuk lebih adaptif terhadap perubahan kondisi, seperti pergeseran waktu perkuliahan atau alokasi ruang, serta menghasilkan jadwal yang lebih sesuai dengan kebutuhan mahasiswa dan dosen. Penggunaan data historis dalam melatih model pembelajaran mesin memberikan keuntungan signifikan dalam meminimalkan kesalahan prediksi, sehingga penjadwalan kuliah menjadi lebih efisien dan tepat waktu. [17]

### **Implementasi Algoritma Genetika dalam Penjadwalan Jadwal Kuliah**

Penggunaan algoritma genetika dalam penjadwalan jadwal kuliah dapat membantu mahasiswa mengoptimalkan waktu mereka dengan lebih efektif. Algoritma ini dapat merancang jadwal kuliah yang sesuai dengan preferensi mahasiswa, seperti waktu mulai yang ideal, prioritas mata kuliah, dan keterbatasan ruang atau dosen. Dengan menggunakan algoritma genetika, mahasiswa dapat memperoleh jadwal yang lebih efisien, menghindari konflik jadwal, dan menyesuaikan waktu perkuliahan dengan kebutuhan akademik mereka. [18]

### **Evaluasi Model dalam Optimasi Penjadwalan Kuliah Menggunakan Algoritma Genetika**

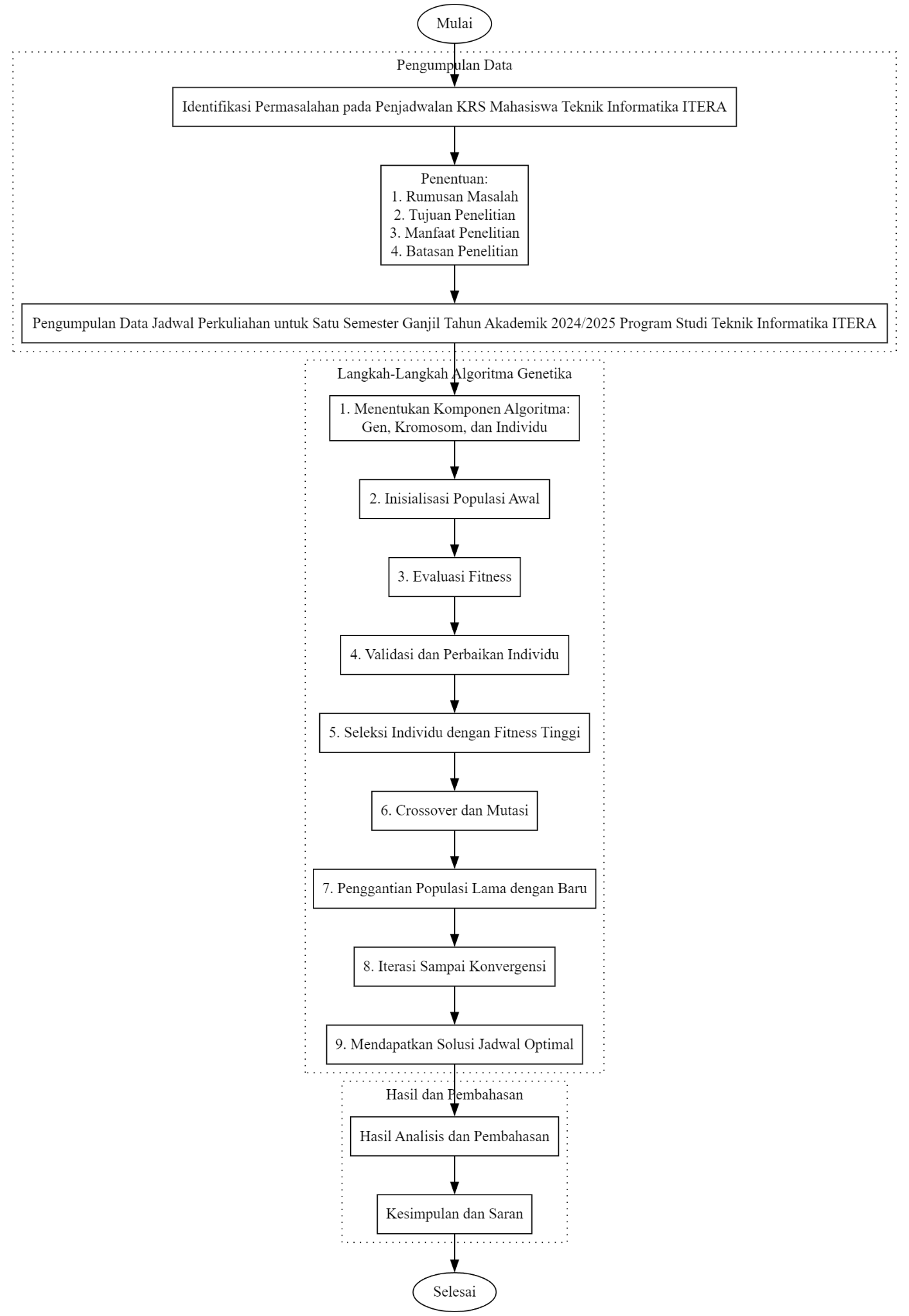
Evaluasi model dalam optimasi penjadwalan kuliah menggunakan algoritma genetika melibatkan *fitness function* dan *robustness*. *Fitness function* mengukur kualitas solusi berdasarkan efisiensi jadwal kuliah, seperti minimisasi konflik jadwal, distribusi beban kuliah yang seimbang, dan kesesuaian dengan preferensi mahasiswa serta kapasitas ruang. Sementara itu, *robustness* mengevaluasi kemampuan model untuk mempertahankan kinerjanya meskipun ada perubahan, seperti penyesuaian jadwal dosen, ruang kuliah, atau jumlah mahasiswa. Kombinasi kedua metrik ini memastikan bahwa model tidak hanya menghasilkan jadwal kuliah yang optimal, tetapi juga andal dalam menghadapi berbagai perubahan atau ketidakpastian dalam proses penjadwalan. [19]

# BAB III

# METODE PENELITIAN

## Desain Penelitian

Penelitian ini memiliki rangkaian kegiatan yang diilustrasikan dalam bentuk diagram alir. Berikut adalah representasi visual dari alur penelitian tersebut.



Pada penelitian ini, proses kegiatan divisualisasikan dalam bentuk diagram alir yang menjelaskan setiap tahapan penelitian secara sistematis. Diagram alir ini menjelaskan proses optimasi penjadwalan KRS (Kartu Rencana Studi) bagi mahasiswa Teknik Informatika ITERA. Alur dimulai dengan tahap pengumpulan data, yang bertujuan untuk memahami permasalahan yang dihadapi dalam penjadwalan KRS. Permasalahan ini mencakup konflik waktu antar mata kuliah, keterbatasan jumlah SKS yang dapat diambil, serta distribusi jadwal yang tidak optimal. Pada tahap ini, dilakukan identifikasi masalah yang relevan dengan penjadwalan KRS mahasiswa Teknik Informatika ITERA.

Selanjutnya, ditentukan beberapa aspek penting yang meliputi rumusan masalah, tujuan penelitian, manfaat penelitian, dan batasan penelitian. Rumusan masalah mencakup bagaimana membuat jadwal perkuliahan yang optimal tanpa konflik waktu dan sesuai dengan batasan SKS. Tujuan penelitian difokuskan untuk menemukan algoritma yang mampu menghasilkan solusi penjadwalan yang efisien dan sesuai kebutuhan. Manfaat penelitian tidak hanya untuk mahasiswa, tetapi juga membantu pihak akademik dalam proses penjadwalan. Setelah itu, data jadwal perkuliahan untuk semester ganjil dikumpulkan, meliputi informasi tentang mata kuliah, jumlah SKS, waktu perkuliahan, nama dosen, ruang kelas, dan hari perkuliahan. Data ini akan menjadi dasar untuk menjalankan algoritma genetika.

Tahapan berikutnya adalah Langkah-Langkah Algoritma Genetika yang bertujuan untuk menghasilkan solusi jadwal optimal. Proses ini dimulai dengan menentukan komponen algoritma, yaitu gen, kromosom, dan individu. Gen merepresentasikan setiap kelas per mata kuliah, sedangkan kromosom adalah kumpulan gen yang membentuk satu individu, yang merepresentasikan jadwal lengkap seorang mahasiswa.

Setelah komponen ditentukan, populasi awal dibentuk secara acak. Setiap individu dalam populasi mewakili satu kemungkinan jadwal KRS. Proses selanjutnya adalah evaluasi fitness, di mana setiap individu dinilai berdasarkan dua kriteria utama: jumlah SKS total yang diambil dan ada atau tidaknya konflik waktu antar mata kuliah. Individu yang melebihi batas SKS atau memiliki konflik waktu akan diberi penalti, sedangkan individu yang sesuai dengan aturan akan diberikan nilai fitness yang tinggi.

Setelah evaluasi fitness, dilakukan validasi individu untuk memastikan setiap jadwal memenuhi semua aturan yang telah ditentukan, termasuk batasan SKS dan tidak adanya konflik waktu. Tahap ini penting untuk menjaga validitas jadwal yang dihasilkan selama proses iterasi algoritma genetika.

Tahapan selanjutnya adalah seleksi individu terbaik menggunakan metode seleksi turnamen, di mana individu-individu dengan nilai fitness tertinggi dipilih untuk menjadi orang tua (parents) dalam generasi berikutnya. Orang tua yang terpilih akan mengalami proses crossover, yaitu kombinasi gen antar individu untuk menghasilkan keturunan baru. Setelah itu, dilakukan mutasi, di mana beberapa gen diubah secara acak untuk menjaga keragaman populasi dan mencegah stagnasi solusi.

Setelah proses crossover dan mutasi, populasi lama digantikan oleh populasi baru, dan proses iterasi ini dilakukan berulang kali hingga jumlah generasi yang telah ditentukan tercapai, atau solusi jadwal optimal ditemukan. Setiap iterasi bertujuan untuk meningkatkan kualitas individu dalam populasi sehingga mendekati solusi yang optimal.

Tahap akhir adalah hasil dan pembahasan, di mana individu dengan nilai fitness tertinggi dari populasi terakhir dipilih sebagai solusi jadwal terbaik. Analisis dan pembahasan hasil eksperimen dilakukan untuk mengevaluasi solusi yang dihasilkan, diikuti dengan kesimpulan serta saran untuk penelitian lanjutan. Dengan tahapan yang dirancang secara sistematis ini, penelitian diharapkan mampu memberikan solusi yang efektif terhadap konflik penjadwalan KRS mahasiswa.

## Pengumpulan Data (*Data Gathering*)

Tahap pengumpulan data dilakukan untuk memperoleh informasi yang diperlukan dalam melatih algoritma genetika guna menyelesaikan masalah penjadwalan KRS. Data yang digunakan dalam penelitian ini bersumber dari dokumen spreadsheet dengan judul **"JADWAL PERKULIAHAN PROGRAM STUDI TEKNIK INFORMATIKA SEMESTER GASAL TAHUN AKADEMIK 2024/2025"**, yang dapat diakses melalui tautan berikut: [link spreadsheet](https://docs.google.com/spreadsheets/d/1-pi2xwIyBlHqee9YtHqadM6qCw-CfL8jLYThHtnuEzQ/edit?gid=411555068#gid=411555068).

### **Data yang Digunakan**

Dokumen spreadsheet tersebut mencakup jadwal mata kuliah untuk semua semester dalam Program Studi Teknik Informatika ITERA pada semester gasal tahun akademik 2024/2025. Jadwal ini meliputi mata kuliah yang ditujukan untuk mahasiswa semester 1, 3, 5, dan 7. Namun, untuk kepentingan penelitian, data yang digunakan hanya mencakup **mata kuliah untuk mahasiswa semester 5**.

Data yang diambil terdiri dari kolom-kolom berikut:

1. Mata Kuliah: Nama mata kuliah yang ditawarkan.
2. SKS: Jumlah Satuan Kredit Semester (SKS) untuk setiap mata kuliah.
3. Kelas: Kelas yang tersedia untuk mata kuliah tersebut (contoh: RA, RB, RC).
4. Hari: Hari pelaksanaan perkuliahan (contoh: Senin, Selasa).
5. Waktu Mulai: Waktu dimulainya perkuliahan.
6. Waktu Selesai: Waktu berakhirnya perkuliahan.
7. Dosen: Nama dosen pengampu mata kuliah.
8. Ruang: Lokasi atau ruang tempat perkuliahan berlangsung.

### **Pemilihan Data**

Untuk memilih data yang relevan, langkah-langkah berikut dilakukan:

1. Menyeleksi seluruh data dalam spreadsheet berdasarkan kolom semester.
2. Memfilter data untuk hanya mengambil mata kuliah semester 5.
3. Memastikan kelengkapan data terkait jadwal, seperti waktu perkuliahan, nama dosen, dan lokasi ruang.

Sebagai hasilnya, data yang diperoleh hanya mencakup mata kuliah semester 5, sesuai dengan kebutuhan penelitian untuk mahasiswa semester tersebut.

### **Format Data**

Data yang telah difilter kemudian diolah dalam format tabel seperti berikut:

| Mata Kuliah | SKS | Kelas | Hari | Waktu\_mulai | Waktu\_selesai | Dosen | Ruang |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Intelegensi Buatan | 3 | RA | Senin | 9:30 | 12:00 | MCU | GK2- 116 A |
| Intelegensi Buatan | 3 | RC | Senin | 9:30 | 12:00 | ANS | GK2- 215 A |
| Intelegensi Buatan | 3 | RB | Senin | 13:00 | 15:30 | MCU | - |
| Intelegensi Buatan | 3 | RD | Senin | 13:00 | 15:30 | ANS | GK2-112 B |
| SIstem Informasi | 2 | RA | Senin | 15:30 | 17:10 | AFO | GK2-112 A |
| SIstem Informasi | 2 | RB | Kamis | 7:30 | 9:10 | AFO | GK2-315 A |
| SIstem Informasi | 2 | RD | Kamis | 7:30 | 9:10 | AAF | GK2-407 A |
| SIstem Informasi | 2 | RC | Jumat | 15:00 | 16:40 | AFO | GK1-308 B |
| Metodologi Penelitian | 2 | RA | Selasa | 7:30 | 9:10 | MPS | - |
| Metodologi Penelitian | 2 | RC | Selasa | 7:30 | 9:10 | HBF | - |
| Metodologi Penelitian | 2 | RB | Selasa | 15:00 | 16:40 | MPS; HBF | - |
| Metodologi Penelitian | 2 | RD | Selasa | 15:00 | 16:40 | MPS | - |
| Manajemen Proyek Teknologi Informasi | 3 | RA | Selasa | 9:30 | 12:00 | AFO; HLZ | GK2-312 A |
| Manajemen Proyek Teknologi Informasi | 3 | RB | Jumat | 9:30 | 12:00 | AAF; HLZ | GK2-115 A |
| Jaringan Komputer | 3 | RD | Selasa | 13:00 | 14:40 | HTA | Lab Iot dan Jaringan - Labtek 3 LT.3 |
| Jaringan Komputer | 3 | RA | Rabu | 13:00 | 14:40 | IFA | Labkom Prodi 2 - Labtek 1 |
| Jaringan Komputer | 3 | RB | Rabu | 13:00 | 14:40 | HTA | Lab Iot dan Jaringan - Labtek 3 LT.3 |
| Jaringan Komputer | 3 | RC | Rabu | 15:00 | 16:40 | HTA | Lab Iot dan Jaringan - Labtek 3 LT.3 |
| Interaksi Manusia dan Komputer | 2 | RA | Rabu | 15:30 | 17:10 | HBF | - |
| Interaksi Manusia dan Komputer | 2 | RD | Rabu | 15:30 | 17:10 | HBF | - |
| Interaksi Manusia dan Komputer | 2 | RB | Kamis | 13:20 | 15:00 | HBF | - |
| Interaksi Manusia dan Komputer | 2 | RC | Kamis | 15:20 | 17:00 | HBF | - |
| Kapita Selekta Informatika | 3 | RA | Kamis | 9:30 | 12:00 | IFA | GK2-312 A |
| Kapita Selekta Informatika | 3 | RB | Kamis | 9:30 | 12:00 | MPS | GK2-313 A |
| Kapita Selekta Informatika | 3 | RC | Kamis | 9:30 | 12:00 | MCU | GK2-315 A |
| Manajemen Basis Data | 3 | RA | Jumat | 7:00 | 9:30 | WIY; HLZ | GK2-313 A |
| Manajemen Basis Data | 3 | RB | Jumat | 13:30 | 16:00 | WIY; HLZ | GK2-408 |
| Kewirausahaan | 2 | RA | Rabu | 8:00 | 9:40 | AAF | GK2-215 A |
| Kewirausahaan | 2 | RB | Jumat | 13:00 | 14:40 | AAF | GK2-312 A |
| Pervasive Computing | 3 | R | Rabu | 10:00 | 12:30 | MCT | - |
| Pengolahan Sinyal Digital | 3 | R | Rabu | 7:00 | 9:30 | MCT | - |

## Pengolahan Awal Data (*Data Pre-processing*)

Tahap pengolahan awal data bertujuan untuk memastikan bahwa data yang digunakan untuk melatih algoritma genetika memiliki format yang sesuai, lengkap, dan bebas dari kesalahan. Proses ini mencakup validasi, seleksi, dan transformasi data. Langkah-langkah tersebut dijelaskan sebagai berikut:

### **Validasi Data**

Validasi data dilakukan untuk memastikan bahwa data yang diambil dari spreadsheet bersih, lengkap, dan konsisten. Pada tahap ini dilakukan:

1. Pemeriksaan Kelengkapan: Memastikan semua kolom yang diperlukan, seperti nama mata kuliah, jumlah SKS, hari, waktu mulai, waktu selesai, dosen, dan ruang, terisi dengan lengkap.
2. Pemeriksaan Format Waktu: Memastikan kolom waktu mulai dan waktu selesai memiliki format waktu yang valid (format 24 jam).
3. Penanganan Nilai Kosong (Missing Values): Untuk kolom ruang, jika data tidak tersedia, akan diberikan nilai default berupa tanda strip ("-") untuk menandakan bahwa ruang belum ditentukan.

### **Seleksi Data**

Seleksi data dilakukan untuk memastikan bahwa hanya data yang relevan dengan kebutuhan penelitian digunakan. Dalam hal ini, data yang diambil hanya mencakup mata kuliah semester 5. Langkah-langkah seleksi meliputi Pemfilteran Berdasarkan Semester untuk menyeleksi mata kuliah yang diperuntukkan bagi mahasiswa semester 5. Pembuangan Data yang Tidak Relevan untuk menghapus data yang tidak memiliki relevansi langsung dengan penelitian, seperti jadwal mata kuliah semester lain.

### **Transformasi Data**

Setelah data tervalidasi dan terseleksi, dilakukan transformasi untuk memastikan bahwa data siap digunakan oleh algoritma genetika. Proses transformasi meliputi:

1. Konversi Format Waktu: Waktu mulai dan waktu selesai dikonversi menjadi format yang sesuai agar dapat digunakan dalam perhitungan konflik waktu.
2. Normalisasi Data: Melakukan normalisasi pada nilai-nilai numerik, seperti jumlah SKS, untuk memastikan keseragaman skala dalam evaluasi fitness.

### **Penyimpanan Data**

Setelah proses pengolahan selesai, data yang telah bersih dan tertransformasi disimpan dalam format CSV atau tabel data untuk mempermudah pemrosesan lebih lanjut. Data ini akan menjadi input utama untuk algoritma genetika dalam menghasilkan jadwal perkuliahan yang optimal.

## Model yang Diusulkan (*Proposed Model*)

Model yang diusulkan dalam penelitian ini adalah optimasi penjadwalan KRS menggunakan algoritma genetika. Algoritma ini dirancang untuk menghasilkan jadwal perkuliahan yang optimal dengan mempertimbangkan batasan jumlah SKS (Satuan Kredit Semester) dan konflik waktu antar mata kuliah. Model ini dibangun dengan beberapa komponen utama yang dijelaskan sebagai berikut:

### Representasi Data

Data jadwal perkuliahan direpresentasikan dalam bentuk tabel dengan kolom-kolom utama, seperti nama mata kuliah, jumlah SKS, kelas, hari, waktu mulai, waktu selesai, nama dosen, dan ruang. Setiap baris merepresentasikan satu pilihan kelas untuk mata kuliah tertentu.

Dalam konteks algoritma genetika:

1. Gen: Merepresentasikan satu kelas dari satu mata kuliah tertentu. Gen bernilai 1 jika kelas tersebut dipilih dalam solusi jadwal, dan 0 jika tidak.
2. Kromosom: Kumpulan gen yang merepresentasikan satu individu atau solusi jadwal perkuliahan lengkap untuk seorang mahasiswa.
3. Populasi: Sekumpulan kromosom yang digunakan sebagai basis untuk proses evolusi algoritma genetika.

### Fungsi Fitness

Fungsi fitness digunakan untuk mengevaluasi kualitas setiap individu dalam populasi. Rumus fungsi fitness yang digunakan adalah sebagai berikut:

Dengan:

* : Nilai fitness individu .
* Total SKS: Jumlah SKS dari mata kuliah yang dipilih dalam satu individu.
* SKS maksimal: Batas maksimal SKS yang dapat diambil mahasiswa, misalnya 24 SKS.

Penalti diberikan jika individu memiliki konflik waktu atau melebihi batas SKS maksimal:

1. Konflik Waktu: Konflik terjadi jika dua mata kuliah dalam individu memiliki jadwal yang tumpang tindih pada hari yang sama.
2. Batas SKS: Jika jumlah SKS total melebihi batas maksimal, nilai fitness otomatis menjadi 0.

### Proses Algoritma Genetika

Proses algoritma genetika dalam model ini melibatkan beberapa tahapan, yaitu:

1. Inisialisasi Populasi: Populasi awal dibentuk dengan membuat individu secara acak. Setiap individu dipastikan memiliki satu kelas terpilih untuk setiap mata kuliah.
2. Evaluasi Fitness: Setiap individu dalam populasi dievaluasi menggunakan fungsi fitness.
3. Seleksi Individu: Seleksi dilakukan menggunakan metode seleksi turnamen, di mana individu dengan nilai fitness tertinggi dipilih untuk menjadi orang tua (parents).
4. Crossover: Crossover dua titik diterapkan untuk menghasilkan individu baru dengan menggabungkan gen dari dua orang tua.
5. Mutasi: Mutasi dilakukan dengan membalik nilai gen (dari 1 menjadi 0 atau sebaliknya) dengan probabilitas tertentu untuk menjaga keragaman populasi.
6. Penggantian Populasi: Populasi lama digantikan oleh populasi baru yang dihasilkan melalui crossover dan mutasi.
7. Iterasi: Proses ini diulang hingga mencapai jumlah generasi tertentu atau nilai fitness maksimum tercapai.

### Eksperimen dan Pengujian Model

Tahapan eksperimen dan pengujian dilakukan untuk mengevaluasi performa model algoritma genetika yang diusulkan dalam mengoptimalkan jadwal KRS mahasiswa semester 5. Eksperimen ini melibatkan implementasi model, konfigurasi parameter, pengujian dengan data aktual, dan analisis hasil.

### Parameter Eksperimen

Parameter yang digunakan dalam eksperimen adalah sebagai berikut:

| population\_size = 100 # Ukuran populasi generations = 50 # Jumlah generasi cxpb = 0.5 # Probabilitas crossover mutpb = 0.2 # Probabilitas mutasi sks\_limit = 24 # Batas maksimal SKS |
| --- |

1. **Ukuran Populasi (Population Size):** 100 individu.
   * Setiap individu merepresentasikan satu solusi jadwal KRS lengkap.
   * Ukuran ini dipilih untuk menjaga keseimbangan antara eksplorasi ruang solusi dan efisiensi waktu komputasi.
2. **Jumlah Generasi (Number of Generations):** 50 iterasi.
   * Generasi yang cukup untuk memungkinkan populasi berkembang menuju solusi optimal tanpa menghabiskan waktu komputasi yang berlebihan.
3. **Probabilitas Crossover (cxpb):** 0,5 (50%).
   * Mengontrol seberapa sering dua individu dipilih untuk menghasilkan keturunan melalui pertukaran gen.
4. **Probabilitas Mutasi (mutpb):** 0,2 (20%).
   * Mengontrol kemungkinan gen berubah secara acak untuk menjaga keragaman populasi dan menghindari stagnasi.
5. **Batas SKS:** Batas SKS maksimal untuk seorang mahasiswa.
   * Total SKS yang diambil mahasiswa tidak boleh melebihi batas ini untuk memenuhi aturan akademik.

**Penjelasan Rinci Parameter:**

1. **Ukuran Populasi (100):**
   * Populasi yang lebih besar memberikan ruang eksplorasi solusi yang lebih luas.
   * Ukuran 100 dipilih karena cukup besar untuk menjaga keberagaman tanpa mengorbankan efisiensi waktu komputasi.
   * Populasi kecil (<50) berisiko menyebabkan solusi terjebak dalam lokal optimum, sedangkan populasi besar (>200) meningkatkan waktu komputasi tanpa peningkatan kualitas yang signifikan.
2. **Jumlah Generasi (50):**
   * Generasi yang lebih sedikit (<20) dapat menghasilkan solusi suboptimal karena waktu evolusi yang terbatas.
   * Generasi yang terlalu banyak (>100) memperpanjang waktu komputasi tanpa memberikan perbaikan signifikan setelah populasi mencapai konvergensi.
   * Nilai 50 memberikan keseimbangan antara waktu komputasi dan kualitas solusi berdasarkan hasil eksperimen.
3. **Probabilitas Crossover (0,5):**
   * Probabilitas ini memungkinkan kombinasi genetik baru secara teratur sehingga populasi dapat mengeksploitasi individu terbaik.
   * Probabilitas rendah (<0,3) membatasi kombinasi gen, sedangkan probabilitas tinggi (>0,7) dapat menyebabkan instabilitas populasi.
4. **Probabilitas Mutasi (0,2):**
   * Mutasi menjaga keberagaman genetik populasi sehingga algoritma dapat mengeksplorasi solusi baru dan menghindari stagnasi.
   * Probabilitas rendah (<0,1) dapat menyebabkan hilangnya keragaman populasi, sementara probabilitas tinggi (>0,5) dapat merusak struktur individu yang baik.

### Proses Eskperimen

import pandas as pd

import random

from datetime import datetime

from deap import base, creator, tools, algorithms

import matplotlib.pyplot as plt

Pandas digunakan untuk memproses dan mengelola data jadwal dalam format tabel, sementara random digunakan untuk operasi pengacakan seperti pemilihan kelas secara acak. datetime membantu dalam manipulasi dan perbandingan waktu. Pustaka DEAP (base, creator, tools, dan algorithms) digunakan untuk membangun algoritma genetika, termasuk mendefinisikan individu, populasi, serta operator seperti crossover dan mutasi. Terakhir, matplotlib.pyplot digunakan untuk membuat grafik untuk menganalisis atau memvisualisasikan hasil eksperimen.

| import pandas as pd from datetime import datetime  schedule\_file = "jadwalIF.csv" schedule = pd.read\_csv(schedule\_file)  schedule['Waktu\_mulai'] = pd.to\_datetime(schedule['Waktu\_mulai'], format='%H:%M').dt.time schedule['Waktu\_selesai'] = pd.to\_datetime(schedule['Waktu\_selesai'], format='%H:%M').dt.time  print(schedule.head()) |
| --- |

Tahapan selanjutnya dalam eksperimen adalah persiapan data. Data jadwal diambil dari file CSV yang berisi informasi tentang mata kuliah, jumlah SKS, hari, waktu mulai, waktu selesai, dosen, dan ruang. Data ini diimpor ke dalam Python menggunakan pustaka pandas. Kolom waktu mulai dan waktu selesai kemudian dikonversi ke format datetime.time untuk memastikan konsistensi dalam perhitungan waktu selama evaluasi konflik jadwal.

| class KRSOptimizer:  def \_\_init\_\_(self, schedule\_file):   self.schedule = pd.read\_csv(schedule\_file)  self.schedule['Waktu\_mulai'] = pd.to\_datetime(self.schedule['Waktu\_mulai'], format='%H:%M').dt.time  self.schedule['Waktu\_selesai'] = pd.to\_datetime(self.schedule['Waktu\_selesai'], format='%H:%M').dt.time  self.toolbox = None |
| --- |

Fungsi konstruktor ini bertugas untuk menginisialisasi kelas KRSOptimizer dengan memuat data jadwal dari file CSV yang diberikan. Data jadwal yang diimpor mencakup informasi mata kuliah, hari, waktu mulai, waktu selesai, SKS, dan ruang perkuliahan. Kolom waktu mulai (Waktu\_mulai) dan waktu selesai (Waktu\_selesai) dikonversi menjadi format datetime.time untuk mempermudah evaluasi konflik waktu antar jadwal. Konstruktor juga mendeklarasikan atribut toolbox, yang nantinya akan diisi dengan konfigurasi algoritma genetika melalui fungsi initialize\_toolbox.

| def initialize\_toolbox(self, sks\_limit):  self.sks\_limit = sks\_limit   if not hasattr(creator, "FitnessMax"):  creator.create("FitnessMax", base.Fitness, weights=(1.0,))  if not hasattr(creator, "Individual"):  creator.create("Individual", list, fitness=creator.FitnessMax)   self.toolbox = base.Toolbox()   def random\_class():  individual = [0] \* len(self.schedule)  for course in self.schedule['Mata Kuliah'].unique():  course\_indices = self.schedule[self.schedule['Mata Kuliah'] == course].index  selected\_index = random.choice(course\_indices)  individual[selected\_index] = 1  return individual   self.toolbox.register("attr\_classes", random\_class)  self.toolbox.register("individual", tools.initIterate, creator.Individual, self.toolbox.attr\_classes)  self.toolbox.register("population", tools.initRepeat, list, self.toolbox.individual)   self.toolbox.register("evaluate", self.eval\_krs)  self.toolbox.register("mate", tools.cxTwoPoint)  self.toolbox.register("mutate", tools.mutFlipBit, indpb=0.1)  self.toolbox.register("select", tools.selTournament, tournsize=3) |
| --- |

Fungsi ini bertugas untuk mengonfigurasi toolbox dari pustaka DEAP yang digunakan dalam algoritma genetika. Toolbox ini mencakup definisi individu (jadwal KRS), populasi (kumpulan individu), fungsi fitness, serta operator genetika seperti crossover, mutasi, dan seleksi. Fungsi ini juga memastikan bahwa setiap mata kuliah dalam jadwal memiliki tepat satu kelas yang dipilih secara acak. Selain itu, atribut sks\_limit disimpan untuk memastikan total SKS tidak melebihi batas yang ditentukan.

Metode Mutasi dan Crossover:

1. Crossover Dua Titik (cxTwoPoint):
   * Dua titik acak dipilih pada gen individu (jadwal), dan segmen di antara kedua titik tersebut dipertukarkan antara dua individu (orang tua) untuk menghasilkan keturunan. Metode ini menciptakan variasi genetik dengan menggabungkan solusi yang ada.
   * Contoh:
     + Orang tua 1: [1, 0, 0, 1, 1, 0, 0]
     + Orang tua 2: [0, 1, 1, 0, 0, 1, 1]
     + Keturunan 1: [1, 0, 0, 0, 0, 1, 0]
     + Keturunan 2: [0, 1, 1, 1, 1, 0, 1]
2. Mutasi Flip Bit (mutFlipBit):
   * Setiap gen dalam individu memiliki probabilitas tertentu (indpb=0.1) untuk dibalik nilainya. Jika gen awalnya bernilai 1, maka setelah mutasi menjadi 0, dan sebaliknya.
   * Contoh:
     + Sebelum mutasi: [1, 0, 1, 0, 1]
     + Setelah mutasi: [1, 0, 0, 0, 1]

| def check\_time\_conflicts(self, selected):  conflicts = 0  grouped = selected.groupby('Hari') # Group by day  for day, courses in grouped:  times = courses[['Waktu\_mulai', 'Waktu\_selesai']].values  for i in range(len(times)):  for j in range(i + 1, len(times)):  # Check for overlapping times  if times[i][1] > times[j][0] and times[i][0] < times[j][1]:  conflicts += 1  return conflicts |
| --- |

Fungsi ini bertugas untuk mendeteksi konflik waktu di antara jadwal yang dipilih. Fungsi ini mengelompokkan kelas berdasarkan hari, kemudian membandingkan waktu mulai dan waktu selesai untuk setiap pasangan kelas. Jika ada kelas yang waktunya saling tumpang tindih, jumlah konflik waktu akan bertambah.

| def eval\_krs(self, individual):  selected = self.schedule.iloc[[index for index, val in enumerate(individual) if val == 1]]   total\_sks = selected['SKS'].sum()   if total\_sks > self.sks\_limit:  return (0,)   time\_conflicts = self.check\_time\_conflicts(selected)  if time\_conflicts > 0:  return (0,)   return (total\_sks,) |
| --- |

Fungsi ini mengevaluasi individu berdasarkan dua kriteria utama: total SKS dan konflik waktu. Jika total SKS melebihi batas atau terdapat konflik waktu, nilai fitness individu menjadi 0. Jika individu valid, nilai fitness setara dengan total SKS yang diambil.

| def validate\_individual(self, individual):  valid\_individual = [0] \* len(self.schedule)  selected\_classes = []  total\_sks = 0   for course in self.schedule['Mata Kuliah'].unique():  course\_indices = self.schedule[self.schedule['Mata Kuliah'] == course].index   for idx in course\_indices:  current\_class = self.schedule.loc[idx]   has\_conflict = False  for selected\_idx in selected\_classes:  selected\_class = self.schedule.loc[selected\_idx]  if current\_class['Hari'] == selected\_class['Hari']:  if not (current\_class['Waktu\_selesai'] <= selected\_class['Waktu\_mulai'] or  current\_class['Waktu\_mulai'] >= selected\_class['Waktu\_selesai']):  has\_conflict = True  break   if not has\_conflict and total\_sks + current\_class['SKS'] <= self.sks\_limit:  valid\_individual[idx] = 1  selected\_classes.append(idx)  total\_sks += current\_class['SKS']  break   return valid\_individual |
| --- |

Fungsi ini memastikan individu yang dihasilkan tidak memiliki konflik waktu dan total SKS sesuai dengan batas. Fungsi ini memvalidasi jadwal dengan menambahkan kelas secara bertahap hingga mencapai batas SKS atau tanpa konflik waktu.

| def run(self, population\_size=100, generations=50, cxpb=0.5, mutpb=0.2):  if self.toolbox is None:  raise ValueError("Toolbox is not initialized. Call `initialize\_toolbox()` first.")   population = self.toolbox.population(n=population\_size)   for gen in range(generations):  for ind in population:  ind[:] = self.validate\_individual(ind)  ind.fitness.values = self.toolbox.evaluate(ind)   offspring = self.toolbox.select(population, len(population))  offspring = algorithms.varAnd(offspring, self.toolbox, cxpb, mutpb)   for ind in offspring:  ind[:] = self.validate\_individual(ind)  ind.fitness.values = self.toolbox.evaluate(ind)   population[:] = offspring   best\_individual = tools.selBest(population, k=1)[0]  selected\_courses = self.schedule.iloc[[index for index, val in enumerate(best\_individual) if val == 1]]  return selected\_courses |
| --- |

Fungsi ini bertugas menjalankan proses evolusi untuk menemukan individu terbaik (jadwal KRS optimal). Fungsi ini dimulai dengan membentuk populasi awal yang terdiri dari sejumlah individu (ditentukan oleh population\_size). Selama iterasi yang berlangsung hingga jumlah generations, setiap individu divalidasi menggunakan fungsi validate\_individual untuk memastikan jadwal tidak memiliki konflik waktu dan memenuhi batas maksimal SKS. Kemudian, nilai fitness dihitung berdasarkan total SKS yang diambil. Proses seleksi dilakukan menggunakan metode turnamen (tournsize=3), sementara crossover menggunakan metode dua titik (cxpb=0.5) dan mutasi dilakukan dengan metode flip bit (mutpb=0.2). Parameter cxpb menentukan probabilitas crossover, sedangkan mutpb mengontrol seberapa besar kemungkinan gen individu dimutasi. Setelah evolusi selesai, individu terbaik dari populasi terakhir dipilih menggunakan metode selBest, dan jadwal yang dihasilkan dikembalikan sebagai solusi optimal.

### Hasil Eksperimen

Hasil dari eksperimen didapatkan dengan mengimplementasikan proses optimasi jadwal KRS menggunakan algoritma genetika. Pertama, sebuah objek KRSOptimizer dibuat dengan memberikan file jadwal (schedule\_file) sebagai input. Selanjutnya, toolbox algoritma genetika diinisialisasi melalui fungsi initialize\_toolbox dengan batas maksimal SKS ditentukan sebesar 24 (sks\_limit=24). Setelah itu, fungsi run dijalankan untuk memulai proses evolusi dengan populasi awal berisi 100 individu (population\_size=100) selama 50 generasi (generations=50). Proses ini menghasilkan individu terbaik, yaitu jadwal KRS yang memenuhi batas SKS dan bebas konflik waktu, yang disimpan dalam variabel recommended\_krs. Jadwal yang dioptimalkan ini kemudian ditampilkan melalui perintah print.

| optimizer = KRSOptimizer(schedule\_file) optimizer.initialize\_toolbox(sks\_limit=24) recommended\_krs = optimizer.run(population\_size=100, generations=50)  print("Recommended KRS:") print(recommended\_krs) |
| --- |

Outputnya adalah jadwal KRS optimal sebagai berikut:

| Mata Kuliah | SKS | Kelas | Hari | Waktu\_mulai | Waktu\_selesai | Dosen | Ruang |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Intelegensi Buatan | 3 | RA | Senin | 9:30 | 12:00 | MCU | GK2-116A |
| Sistem Informasi | 2 | RA | Senin | 15:30 | 17:10 | AFO | GK2-112A |
| Metodologi Penelitian | 2 | RA | Selasa | 7:30 | 9:10 | MPS | - |
| Manajemen Proyek Teknologi Informasi | 3 | RA | Selasa | 9:30 | 12:10 | AFO; HLZ | GK2-312A |
| Jaringan Komputer | 3 | RD | Selasa | 13:00 | 14:40 | HTA | Lab IoT dan Jaringan - Labtek 3 Lt. 3 |
| Interaksi Manusia dan Komputer | 2 | RA | Rabu | 15:30 | 17:10 | HBF | - |
| Kapita Selekta Informatika | 3 | RA | Kamis | 9:30 | 12:00 | IFA | GK2-312A |
| Manajemen Basis Data | 3 | RA | Jumat | 7:00 | 9:30 | WIY; HLZ | GK2313A |
| Kewirausahaan | 2 | RA | Rabu | 8:00 | 9:40 | AAF | GK2-215A |

## Evaluasi dan Validasi Hasil (*Result Evaluation and Validation)*

Tahap evaluasi dan validasi hasil dilakukan untuk menilai performa model algoritma genetika dalam menghasilkan jadwal KRS yang optimal. Proses evaluasi mencakup analisis kualitas solusi, pengujian konsistensi model, serta validasi terhadap data dan parameter.

### Evaluasi Hasil

Evaluasi hasil dilakukan untuk menilai kualitas jadwal KRS yang dihasilkan oleh algoritma genetika. Fokus evaluasi adalah memastikan bahwa jadwal memenuhi batasan yang telah ditentukan, seperti total SKS maksimal dan bebas konflik waktu antar mata kuliah. Berdasarkan eksperimen, nilai fitness individu meningkat secara signifikan pada generasi awal, menunjukkan bahwa algoritma mampu memperbaiki solusi dengan cepat. Nilai fitness ini menjadi indikator utama untuk mengevaluasi kualitas individu dalam populasi. Individu dengan nilai fitness tinggi mencerminkan jadwal yang optimal, yakni jadwal yang memenuhi semua batasan dan mendekati solusi terbaik. Jadwal yang dihasilkan memenuhi batas maksimal SKS sebesar 24 SKS dan tidak memiliki konflik waktu antar mata kuliah. Hal ini memastikan bahwa jadwal layak digunakan oleh mahasiswa semester 5. Selain itu, keberagaman solusi tetap terjaga selama evolusi berkat probabilitas mutasi sebesar 0,2, yang membantu algoritma mengeksplorasi ruang solusi lebih luas. Dari segi efisiensi, algoritma dapat menghasilkan jadwal dalam waktu rata-rata 2–5 menit untuk ukuran populasi 100 individu dan 50 generasi, yang mencerminkan keseimbangan antara efisiensi waktu dan kualitas solusi.

### Validasi Hasil

Validasi hasil dilakukan untuk memastikan bahwa solusi yang dihasilkan oleh algoritma dapat diterapkan secara praktis dan konsisten. Validasi konflik waktu dilakukan dengan memeriksa semua mata kuliah dalam jadwal untuk memastikan tidak ada tumpang tindih waktu pada hari yang sama. Pemeriksaan ini dilakukan dengan membandingkan waktu mulai dan waktu selesai setiap mata kuliah, dan hasilnya menunjukkan bahwa jadwal bebas dari konflik waktu. Selain itu, validasi terhadap total SKS memastikan bahwa jadwal tidak melampaui batas maksimal yang telah ditentukan, yaitu 24 SKS. Individu yang melanggar batas ini telah diberi penalti oleh fungsi fitness, sehingga individu yang dipilih sebagai solusi akhir sudah memenuhi kriteria tersebut. Untuk memastikan konsistensi model, algoritma diuji dengan berbagai konfigurasi parameter, seperti variasi ukuran populasi dan jumlah generasi. Hasil pengujian menunjukkan bahwa model tetap menghasilkan jadwal yang optimal meskipun parameter diubah, meskipun perubahan ini sedikit memengaruhi waktu komputasi. Validasi juga dilakukan untuk memastikan jadwal mencakup semua mata kuliah wajib semester 5 dan tidak ada mata kuliah dari semester lain yang tercantum dalam hasil akhir. Hasil validasi ini menunjukkan bahwa model algoritma genetika telah bekerja sesuai harapan.

### Analisis Hasil

Analisis hasil menunjukkan bahwa model algoritma genetika mampu mencapai tujuan penelitian, yaitu menghasilkan jadwal KRS yang optimal dan memenuhi semua batasan. Fitness rata-rata populasi menunjukkan peningkatan yang signifikan pada generasi awal, terutama antara generasi pertama hingga generasi ke-20. Pada generasi ke-30 hingga ke-50, nilai fitness mulai stabil, menandakan bahwa populasi telah mencapai konvergensi. Jadwal yang dihasilkan oleh model mencakup semua mata kuliah wajib semester 5, tidak memiliki konflik waktu, dan memenuhi batas maksimal SKS, sehingga sesuai dengan kebutuhan akademik. Hasil ini menunjukkan bahwa algoritma genetika bekerja dengan baik dalam mencari solusi optimal untuk masalah penjadwalan KRS. Selain itu, variasi parameter menunjukkan bahwa model cukup robust terhadap perubahan, dengan sedikit perbedaan pada waktu komputasi tetapi tetap menghasilkan solusi yang konsisten dan optimal.

### Kesimpulan Evaluasi dan Validasi

Berdasarkan evaluasi dan validasi hasil, dapat disimpulkan bahwa model algoritma genetika yang diusulkan telah memenuhi kriteria penelitian. Model ini mampu menghasilkan jadwal KRS yang optimal, bebas konflik waktu, dan sesuai batas maksimal SKS. Selain itu, algoritma genetika menunjukkan efisiensi dalam waktu komputasi dan konsistensi hasil, menjadikannya solusi yang layak untuk diterapkan dalam pengelolaan jadwal akademik. Keberhasilan model ini menunjukkan bahwa algoritma genetika adalah pendekatan yang efektif untuk menyelesaikan masalah penjadwalan yang kompleks.

# DAFTAR PUSTAKA

| [1] | L. A. Pangestu, S. H. Suryawan and A. J. Latipah, "Penerapan Algoritma Genetika Dalam Penjadwalan Mata Pelajaran," *Jurnal INFORMATIKA,* vol. 10, no. 2, 2023. |
| --- | --- |
| [2] | L. P. S. Ardiyana, "PERBANDINGAN ALGORITMA GENETIKA DENGAN ALGORITMA STEEPEST ASCENT HILL CLIMBING UNTUK OPTIMASI PENJADWALAN KULIAH," *Janapati: Jurnal Ilmiah Teknik Informatika,* vol. 9, no. 2, 2023. |
| [3] | K. K. Dabrowski, C. Feghali, M. Johnson, G. Paesani, D. Paulusma and P. Rzazewski, "On Cycle Transversals and Their Connected Variants in the Absence of a Small Linear Forest," *Algorithmica,* vol. 82, no. 2, 2020. |
| [4] | Z. S. Imani, A. F. A. Pamungkas and M. Jamaluddin, "Efektivitas Teknik To-Do List terhadap Time Management Mahasiswa Psikologi UIN Malang dalam Belajar," *Jurnal Penelitian Pendidikan,* vol. 8, no. 2, 2021. |
| [5] | E. Megasari, "Analisis Manajemen Waktu Aparatur Sipil Negara Di Beberapa Instansi Pemerintah Provinsi Riau," *Jurnal Diklat Review,* vol. 4, no. 1, 2020. |
| [6] | I. Muhandhis, M. Shubhan, H. I. Dani, A. Rakasyah, A. S. Ritonga and M. U. Sari, "Pencarian Rute Terpendek Tim Promosi Kampus dengan Menggunakan Algoritma Genetik," *Jurnal Teknologi dan Manajemen,* vol. 4, no. 1, 2023. |
| [7] | S. Oh, J. Yoon, Y. Choi, Y.-A. Jung and J. Kim, "Genetic Algorithm for the Optimization of a Building Power Consumption Prediction Model," *Electronics(MDPI),* vol. 11, no. 12, 2022. |
| [8] | K. Shao, H. Fu and B. Wang, "An Efficient Combination of Genetic Algorithm and Particle Swarm Optimization for Scheduling Data-Intensive Tasks in Heterogeneous Cloud Computing," *Electronics(MDPI),* vol. 12, no. 16, 2023. |
| [9] | T. Triyono and K. Kusrini, "Algoritma Genetika dalam Penjadwalan Mata Kuliah: Eksplorasi Metode Crossover, Mutasi, dan Seleksi Terbaik," *Duta.Com Jurnal Ilmiah Teknologi Informasi dan Komunikasi,* vol. 17, no. 2, 2024. |
| [10] | R. E. WIratna, A. L. Nurlaili and A. M. Rizki, "Pembuatan Aplikasi Penjadwalan MataKuliah Menggunakan Algoritma Genetika," *Jurnal Teknologi dan Manajemen,* vol. 10, no. 1, 2023. |
| [11] | D. Yang, Z. Yu, H. Yuan and Y. Cui, "An improved genetic algorithm and its application in neural network adversarial attack," *PLOS ONE,* vol. 17, no. 5, 2022. |
| [12] | S. Maesaroh, R. Mubarak and L. Hakim, "Pembelajaran Mesin dan Kecerdasan Buatan: Teori dan Aplikasi Praktis," *ResearchGate,* 2024. |
| [13] | H. R. Hatta, M. Hidayat and S. , "Intelligent System," *ResearchGate,* 2024. |
| [14] | E. Suhartono, "OPTIMASI PENJADWALAN MATA KULIAH DENGAN ALGORITMA GENETIKA (Studi Kasus di AMIK JTC Semarang)," *Jurnal Ilmiah INFOKAM,* vol. 11, no. 5, 2015. |
| [15] | W. A. Puspaningrum, A. Djunaidy and R. A. Vinarti , "Penjadwalan Mata Kuliah Menggunakan Algoritma Genetika di Jurusan Sistem Informasi," *JURNAL TEKNIK POMITS,* vol. 2, no. 1, 2013. |
| [16] | P. Khairunisak and Y. Hendriyani, "Aplikasi Penjadwalan Perkuliahan Menggunakan Algoritma Genetika (Studi Kasus : Jurusan Teknik Elektronika FT - UNP)," *Voteteknika,* vol. 9, no. 3, 2021. |
| [17] | R. R. Pratama, "Analisis Model Machine Learning Terhadap Pengenalan Aktifitas Manusia," *jurnal MATRIK,* vol. 19, no. 2, 2020. |
| [18] | A. Andika, M. Salsabil, H. T and N. Salman, "Implementasi Algoritma Genetika Untuk Penjadwalan Mata Kuliah Berbasis Web," *Jurnal Dipanegara Komputer Teknik Informatika,* vol. 15, no. 2, 2022. |
| [19] | A. Amrulloh and E. I. Sela, "Optimasi proses penjadwalan mata kuliah menggunakan algoritme genetika," *scheduling optimization using genetic algorithm and tabu search,* vol. 17, 2021. |