

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

2.1.1 Algoritma Genetika

Algoritma genetika merupakan salah satu metode optimasi yang terinspirasi dari proses evolusi biologis melalui mekanisme seleksi alam, crossover, dan mutasi (Sapta et al., 2025). Algoritma genetika bekerja dengan menghasilkan sejumlah populasi solusi yang kemudian di evaluasi menggunakan fungsi fitness, lalu pada tahap berikutnya akan melalui proses iterasi berulang atau generasi, solusi tersebut diperbaiki hingga ditemukan solusi paling optimal. Dalam konteks penjadwalan, algoritma genetika sangat relevan karena mampu mengakomodasi banyak kendala atau biasa disebut dengan constraint sekaligus, seperti keterbatasan waktu, ruang, serta ketersediaan dosen yang sulit diselesaikan secara manual.

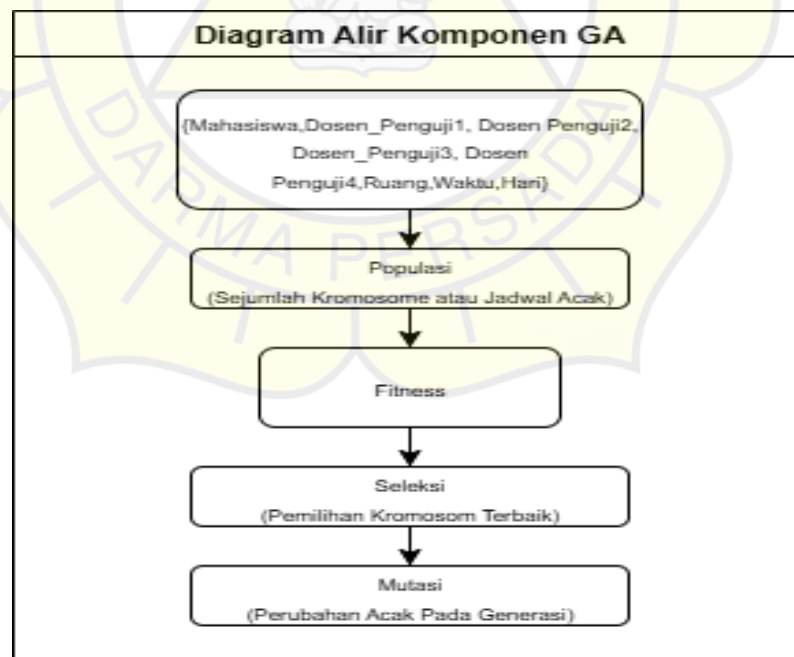
Menurut (Fuenmayor et al., 2022), penerapan algoritma genetika dalam penjadwalan ruang laboratorium dengan penyesuaian operator mutasi terbukti mampu menghasilkan jadwal yang lebih cepat, efisien dan stabil dibandingkan pendekatan konvensional. Selain itu, mengembangkan modifikasi algoritma genetika pada proses encoding variable dan operator lokal membuat kecepatan konfigurasi dalam pendekatan ini meningkat secara signifikan dan permasalahan pada lokal optimal dapat dihindari.

Hal ini membuktikan bahwa algoritma genetika sangat fleksibel dalam mengakomodasi berbagai parameter penjadwalan yang dinamis, sehingga cocok untuk permasalahan penjadwalan kuliah maupun sidang yang berubah sewaktu-waktu.

Berdasarkan dari penelitian tersebut, dapat disimpulkan bahwa algoritma genetika memiliki tingkat efektifitas tinggi dalam menyelesaikan masalah penjadwalan, terutama dalam mengoptimalkan alokasi sumber daya yang terbatas, minimalisir konflik, serta menghasilkan solusi yang efisien.

2.1.2 Komponen Algoritma Genetika

Algoritma genetika atau GA memiliki komponen parameter utama yang menjadi dasar dalam proses pencarian solusi. Setiap komponen saling terkait untuk menghasilkan jadwal yang optimal.



Gambar 2.1 Diagram Alir Komponen GA

Seperti Gambar 2.1, pada kasus penjadwalan sidang skripsi, komponen dapat dijelaskan sebagai berikut:

a. Kromosom

Kromosom merupakan representasi dari sebuah solusi yang menggambarkan susunan jadwal sidang secara lengkap dengan susunan sebagai berikut:

{mahasiswa,Dosen_Penguji1,Dosen_Penguji2,Dosen_Penguji3,Dosen_Peng
uji4,Ruang, Waktu, Hari}

Dengan rangkaian sebagai berikut maka mahasiswa akan mendapatkan 4 dosen pengji, 1 ruangan dengan waktu dan hari yang sudah menjadi pelengkap formasi.

b. Populasi

Populasi merupakan sejumlah jadwal atau kromosom yang di bangkitkan secara acak pada awal proses. Dengan terdapat formasi kromosom maka populasi awal bisa berdiri dari 30-50 jadwal acak. Tujuan nya menyediakan keberagaman solusi agar algoritma dapat mengeksplorasi berbagai kemungkinan kombinasi jadwal.

c. Fitness

Fungsi fitness digunakan untuk menentukan sebuah kelayakan pada kualitas kromosom, semakin tinggi nilai semakin baik untuk sebuah solusi dengan kriteria sebagai berikut:

1. Tidak ada mahasiswa waktu bersamaan diruang yang sama.
2. Dosen penguji tidak ada bentrok pada jadwal yang sama.
3. Distribusi jadwal terkoordinasi.
4. Meminimalisir total bentrok pada sebuah wadah.

Dengan komposisi $\text{Fitness} = \text{Skor Maksimum} - (\text{Jumlah Bentrok Ruang} + \text{Jumlah Bentrok Dosen})$, maka nilai fitness dapat teruji Ketika algoritma genetika dijalankan.

d. Seleksi

Proses pemilihan kromosom terbaik yang akan digunakan sebagai induk untuk membuat generasi baru. Dengan kasus sebagai berikut:

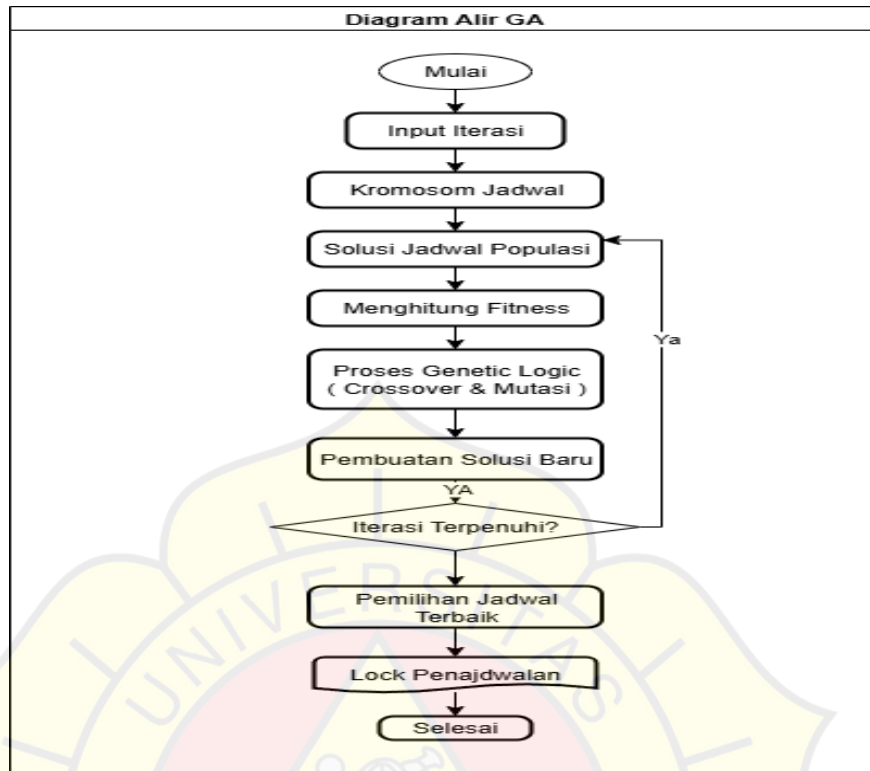
1. Induk 1: [M01, DosenA, DosenB, DosenC, Ruang1, Hari1]
2. Induk 2: [M02, DosenX, DosenY, DosenZ, Ruang2, Hari2]
3. Hasil Crossover (anak): [M01, DosenA, DosenB, DosenC, Ruang2, Hari2]

e. Mutasi

Perubahan acak pada generasi dalam kromosom untuk menjaga keberagaman populasi.

2.1.3 Tahapan Algoritma Genetika

Dalam penerapan pada penjadwalan skripsi, algoritma genetika memiliki serangkaian tahapan sistematis yang dimulai dari populasi hingga memperoleh jadwal optimal. Setiap tahap dirancang untuk meniru mekanisme evolusi ilmiah bertahap. Dengan pendekatan ini sistem diharapkan mampu menghasilkan jadwal terhindar dari bentrokan, tetapi terdistribusi secara adil dan terstruktur.



Gambar 2.2 Diagram Alir Algoritma Genetika

Sumber: Jurnal Teknologi Terapan 2021

Pada gambar 2.2 merupakan diagram alir algoritma genetika, sebagai berikut:

a. Inisialisasi Populasi

Pada tahap ini, sistem membangkitkan populasi awal yang terdiri dari kromosom. Kromosom merepresentasikan jadwal skripsi lengkap dengan kombinasi struktur gen dalam kromosom.

Contoh:

1. 15 mahasiswa harus dijadwalkan. Dengan memiliki 4 dosen tersedia untuk pengujian.
2. Jadwal dibatasi oleh 3 ruangan (T-301,T-302,T-303)

Dengan kondisi ini, sebuah kromosom dapat direpresentasikan sebagai array berdimensi 15 generasi dengan berisikan:

[ID_Mahasiswa, Hari, Ruangan, Pembimbing, Penguji1, Penguji2]

b. Perhitungan Fitness

$$Fitness = 1000 - (w1 \times Cruangan + w2 \times C dosen + w3 \times Cjadwal) \quad (2.1)$$

Setiap kromosom dievaluasi dengan fungsi fitness untuk mengukur kualitas jadwal dengan memounyai beberapa faktor antara lain:

1. Tidak ada bentrok antara ruangan dan dosen
2. Mahasiswa terdistribusi merata.

Semakin sedikit konflik nilai fitness mendekati aturan maksimum.

c. Seleksi

Berdasarkan dengan metode Route Wheel Selection maka probabilitas terpilih kromosom berbanding lurus dengan fitness. Misalkan dengan jumlah 50 kromosom maka 20 di pilih terbaik untuk dilakukan keproses selanjutnya.

d. Crossover

Pada tahap ini crossover adalah penggabungan dua jadwal untuk membentuk jadwal baru. One-Point-Crossover menjadi titik potong tertentu, lalu bagian depan kromosom pertama digabungkan dengan bagian belakang kromosom ke dua, contoh:

1. Jadwal A mengatur sekitar 1-7 mahasiswa dengan adanya konflik pada mahasiswa 8-15.
2. Jadwal B bagus untuk mahasiswa tersisa, dengan terbentuknya kembali jadwal C karena lebih baik menggabungkan keunggulan jadwal A dan B.

e. Mutasi

Dengan swap mutasi yaitu menukar ruangan atau hari sidang antara dua mahasiswa maka terjadilah Reassignment mutasi yaitu mengubah hari sidang secara acak untuk satu mahasiswa. Contoh:

Jika mahasiswa A dan B awalnya sidang dihari yang sama dan bentrok, maka proses mutasi dapat memindahkan A ke hari lain yang masih tersedia.

f. Populasi Terbaru

Setelah proses crossover dan mutasi sudah berhasil dilakukan.

Maka generasi ini kembali dihitung fitnessnya, dengan cara nilai tertinggi akan disimpan sebagai kandidat terbaik pada satu jadwal dan jika masih ada konflik untuk ke 15 mahasiswa tersebut, akan diproses ke generasi berikutnya.

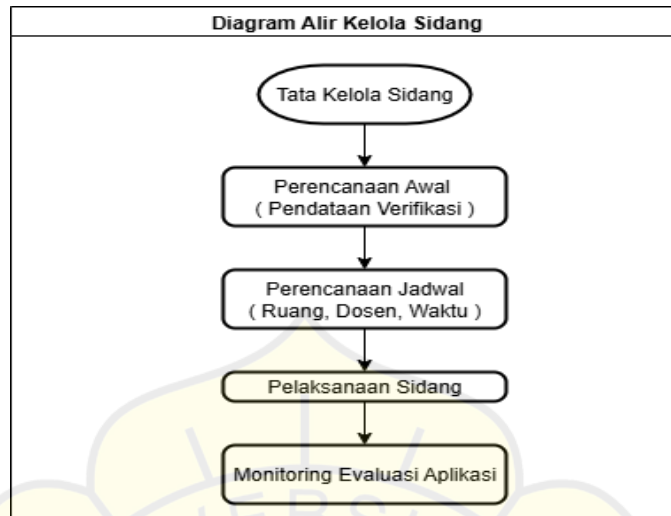
g. Hasil Kondisi Terminasi

Jadwal sidang secara optimal, melainkan dengan hasil output berikut:

1. Hari 1 adalah 5 mahasiswa
2. Hari 2 adalah 5 mahasiswa
3. Hari 3 adalah 5 mahasiswa tersisa.

Maka tidak ada dosen atau jadwal bentrok dengan distribusi seimbang, dan semua mahasiswa terjadwal untuk persidangan.

2.1.4 Tata Kelola Penjadwalan



Gambar 2.3 Diagram Tata Kelola

Sumber: Diadaptasi dari (Guntara et al., 2023) dan dimodifikasi oleh peneliti (2025)

Berdasarkan dari gambar 2.3 merupakan Tata kelola sidang diawali dengan perancangan awal, yaitu pendataan mahasiswa, dengan verifikasi berkas, dan penentuan kebutuhan sidang.

Tahap berikutnya adalah perancangan jadwal, dimana alokasi ruangan, waktu serta pemerataan dosen yang tersedia diatur secara sistematis, setelah jadwal ditetapkan dilaksanakan pelaksanaan sidang sesuai dengan ketentuan dan terakhir adalah monitoring dan evaluasi untuk merekap hasil jadwal yang telah dilakukan dengan proses Algoritma Genetika.

2.1.5 Metodologi DevOps

Metode DevOps merupakan pendekatan modern dalam pengembangan perangkat lunak yang menggabungkan proses development dan operations secara terpadu.

Dengan menekan kolaborasi antar tim dengan pengembang untuk meningkatkan efisiensi perangkat lunak. Pendekatan ini berfokus terhadap otomatisasi, integrasi berkelanjutan, pengiriman berkelanjutan dan monitoring sistem secara real time.

Fleksibilitas, skalabilitas dan peningkatan berkelanjutan seperti sistem penjadwalan skripsi berbasis web deplatform mobile. Dengan penerapan DevOps, proses pengembangan menjadi lebih adaptif terhadap perubahan kebutuhan, serta meminimalisir risiko kegagalan sistem.

Menurut Daives, C metodologi DevOps mampu mengurangi hambatan komunikasi antar tim dan meningkatkan kecepatan pengiriman secara signifikan, tetapi juga menjelaskan bahwa penerapan DevOps memerlukan kesiapan infrastruktur, budaya kerja, kolaboratif serta tools otomatisasi yang tepat untuk mencapai hasil optimal (Ruseno et al., 2025).

Terdapat peran dari DevOps antara lain sebagai berikut:

1. Proses pengembangan lebih cepat dan adaptif terhadap perubahan kebutuhan.
2. Otomatisasi pengujian dan deployment untuk meningkatkan efisiensi.
3. Monitoring berkelanjutan terhadap potensi kegagalan.
4. Meningkatkan komunikasi antar tim pengembang dan operasional.

Dalam konteks penelitian ini, metode DevOps sangat cocok digunakan dalam penerapan sistem penjadwalan sidang skripsi yang dapat dikembangkan sesuai kebutuhan. DevOps juga mendukung efisiensi dengan cara melakukan perbaikan secara terus menerus tanpa perlu memulai proses pengembangan dari awal lagi. (Bayu Aji et al., 2021).

2.1.6 UML (Unified Modifed Language)

Unified Modifed Language merupakan bahwa permodelan visual yang digunakan dalam rekayasa perangkat lunak untuk menggambarkan desain sistem. Uml memberikan kerangka kerja komprehensif untuk modelkan baik secara struktur maupun perilaku sistem antar bagian (Guntara et al., 2023).

UML (Unified Modeling Language) membantu para pengembang perangkat lunak untuk menggambarkan, merancang, menentukan dan mendokumentasi sistem secara visual melalui symbol grafis yang mudah di mengerti. (Handayani et al., 2020)

Dengan cara menampilkan sisi statis maupun dinamis dari sebuah sistem, sehingga sangat berguna untuk memahami struktur dan fungsi sistem secara menyeluruh . Adapun tujuan dan fungsi dari UML sebagai berikut:

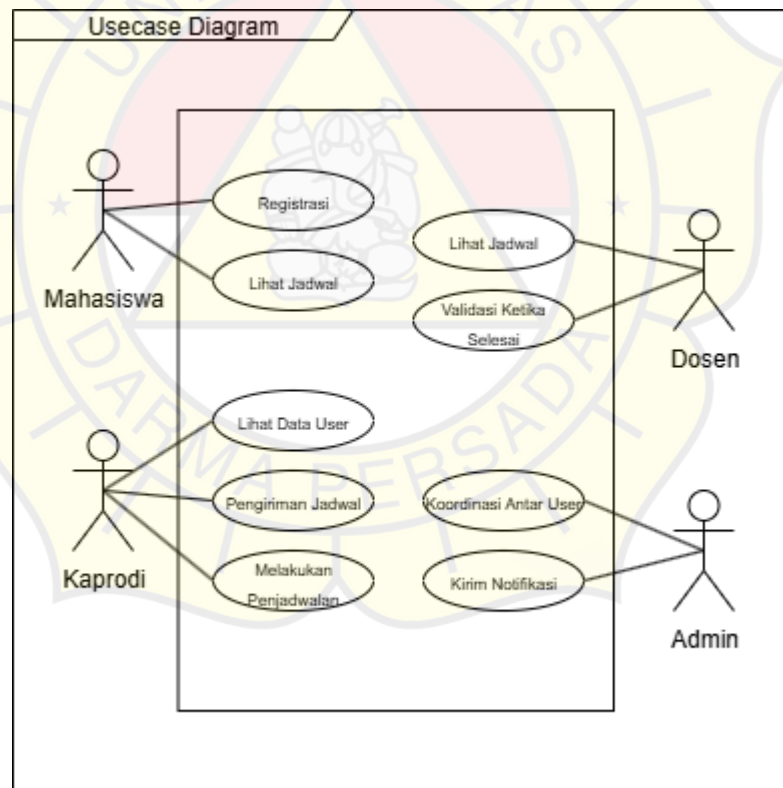
- a. Meningkatkan pemahaman sistem dengan memungkinkan semua pihak yang terlibat dalam proyeksi.
- b. Fasilitasi komunikasi berdasarkan gambar visual yang memudahkan komunikasi antar anggota tim (Soraya & Wahyudi, 2021)
- c. Dokumentasi yang konsisten: UML menyediakan standar yang digunakan untuk dokumentasikan desain sistem secara konsisten, yang mempermudah pemeliharaan sistem di masa depan (Maulana et al., 2025).
- d. Analisis dan perencanaan yang lebih baik: UML memungkinkan analisis yang lebih tepat terhadap kebutuhan sistem dan perencanaan yang lebih matang dalam tahap desain (Santoso & Migunani, 2021).

- e. Fleksibilitas dalam berbagai jenis sistem: UML dapat digunakan untuk model berbagai jenis sistem, baik perangkat lunak, sistem bisnis, maupun alur kerja organisasi (Soraya & Wahyudi, 2021).

1. Model Diagram

a. Usecase Diagram

Pengguna focus pada kebutuhan fungsional sistem, menunjukkan interaksi antara pengguna dan sistem. Diagram ini sangat penting untuk menggambarkan aktivitas yang dilakukan oleh actor berikut adalah Gambar 2.5 usecase diagram sebagai berikut:








Gambar 2.4 Usecase Diagram

Gambar 2.4 menunjukkan Use Case Diagram Sistem Penjadwalan Sidang Skripsi, yang menggambarkan interaksi antara empat aktor utama, yaitu Mahasiswa, Dosen, Kaprodi, dan Admin.

Mahasiswa dapat melakukan perlihatkan sidang. Dosen memiliki akses memvalidasi bahwa sidang sudah tertera atau terlaksana. Kaprodi membuat jadwal berdasarkan algoritma genetika dan admin berfungsi untuk mengkoordinasi antar user yang terlibat.

b. Activity Diagram

Tabel 2.1 Activity Diagram Diagram

No	Gambar	Nama	Keterangan
		Action	State dari sistem yang mencerminkan eksekusi dari suatu aksi.
		Initial Node	Bagaimana bentuk diawali
		Activity Final Node	Bagaimana object diakhiri
		Decision	Pilihan untuk mengambil keputusan dan diakhiri kondisi.
		Transition	Sebuah kejadian yang memicu sebuah state objek dengan cara memperbarui nilai sistem.

Menggambarkan alur kontrol atau urutan aktivitas dalam sistem, mulai dari aktivitas pertama hingga terakhir. Diagram ini sering digunakan untuk model proses atau alur kerja dalam sistem secara rinci, membantu memahami bagaimana tugas-tugas dijalankan (Maulana et al., 2025).

c. Sequence Diagram

Menampilkan urutan interaksi antar komponen sistem. Diagram ini sangat berguna untuk menjelaskan bagaimana komponen-komponen saling berinteraksi, memberikan pemahaman yang lebih jelas tentang proses yang terjadi dalam sistem (Rohmah & Voutama, 2024).

2. Next.js

Next.js adalah framework yang dibangun di atas React.js dan menawarkan fitur untuk pengembangan aplikasi web yang kompleks, seperti Server-Side Rendering (SSR) dan Static Site Generation (SSG).

Fitur SSR memungkinkan halaman untuk dirender di server terlebih dahulu, kemudian dikirim ke klien, yang meningkatkan performa dan SEO karena konten halaman sudah siap ditampilkan ketika sampai di browser. (Al Fikri, 2024)

3. React.js

React.js merupakan JavaScript yang dikembangkan oleh facebook untuk membangun antarmuka pengguna (UI) berbasis komponen.

React memungkinkan pengembang untuk membuat aplikasi single- page application (SPA) yang responsive dan efisien dengan menggunakan client-side rendering (CSR). (Wicaksono & Putra, 2021)

Melalui pendekatan ini, react memanfaatkan virtual Dom untuk memperbarui UI dengan cara yang efisien.

React.js berfokus terhadap pembuatan komponen yang dapat digunakan kembali dan dapat dikelola secara independen. Perihal ini sangat berguna untuk menciptakan aplikasi besar dengan banyak interaksi. (Subang et al., 2024)

2.2 Kajian Literatur

Berdasarkan dari penelitian adapun kajian literatur yang tersedia, antara lain sebagai tabel berikut:

Tabel 2.2 Kajian Literatur Penelitian

No	Peneliti & Tahun	Judul	Metodologi Penelitian	Temuan Utama	Keterkaitan Penelitian Ini
1	Fauzi, Akbar & Kurniawan syah (2024)	Penjadwalan Sidang Skripsi di Universitas Muhammadiyah Jambi	Waterfall, UML, PHP, MySQL, Algoritma Genetika	Sistem mampu menghasilkan jadwal optimal berdasarkan ketersediaan mahasiswa, dosen, dan ruang	Menunjukkan efektivitas GA dalam penjadwalan kompleks; pendekatan penelitian ini
2	Lim Ying Ying & Kutty Mammi (2021)	Timetable Scheduling System using Genetic Algorithm for School of Computing (tsuGA)	Algoritma Genetika	GA mampu menghasilkan jadwal kuliah kompleks secara efisien	Menunjukkan efektivitas GA dalam mengatasi constraint multi-dosen & ruang
3	Berlian Bayu Aji, Susetyo Bagas Bhaskoro, Siti Aminah (2021)	Penjadwalan Tugas Akhir di Polman Bandung	Algoritma Genetika	Tingkat keberhasilan jadwal bebas bentrok >90%, khususnya saat rasio penguji $\geq 2:1$	Membuktikan GA efektif dalam menghindari konflik jadwal pada skenario multi-pengguna

4.	Dewi Nasien & Andi (2022)	Optimization of Genetic Algorithm in Courses Scheduling	GA Optimization	GA dioptimasi untuk mempercepat proses dan menghindari bentrok jadwal	Menerapkan GA dalam konteks universitas dengan variabel mirip sidang skripsi
5.	Benny Wijaya, Syafrul Irawadi, Lili Indah Sari & Wishnu Aribowo (2023)	Subject Scheduling Application Using Genetic Algorithm	GA berbasis aplikasi	Sistem GA menghasilkan jadwal otomatis tanpa konflik di SMK	Mendukung penerapan GA untuk sistem penjadwalan akademik berbasis aplikasi di Indonesia