

**OPTIMASI PENJADWALAN RUANGAN DI GEDUNG
DAKWAH MUHAMMADIYAH CABANG DUKUN
MENGUNAKAN ALGORITMA GENETIKA**

SKRIPSI



Disusun oleh:

Herlando Prayitno

210602011

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS MUHAMMADIYAH GRESIK
2024**

LEMBAR PERSETUJUAN

SKRIPSI

**OPTIMASI PENJADWALAN RUANGAN DI GEDUNG DAKWAH
MUHAMMADIYAH CABANG DUKUN MENGGUNAKAN ALGORITMA
GENETIKA**

Disusun Oleh :

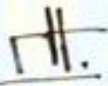
Herlando Pravitno

210602011

Telah disetujui untuk dipresentasikan dalam sidang Skripsi

Susunan Tim Pembimbing

Pembimbing 1



Harunur Rosvid, S.T., M.Kom, Ph.D.

NIP. 0621 0408 106

Pembimbing 2



Henny Dwi Bhakti, S.Si., M.Si.

NIP. 0621 1709 200

Diterima pada tanggal 08 Januari 2025 dan dinyatakan telah memenuhi
persyaratan pelaksanaan sidang Skripsi

Mengetahui,

Ketua Program Studi Teknik Informatika



Henny Dwi Bhakti, S.Si., M.Si.

NIP. 0621 1709 200

**LEMBAR PENGESAHAN
SKRIPSI**

**OPTIMASI PENJADWALAN RUANGAN DI GEDUNG DAKWAH
MUHAMMADIYAH CABANG DUKUN MENGGUNAKAN ALGORITMA
GENETIKA**


Disusun Oleh :

HERLANDO PRAYITNO
210602011

Telah dipertahankan di depan Tim Penguji pada tanggal 13 Januari 2025

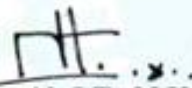
Susunan Tim Pembimbing

Penguji 1 (Ketua)



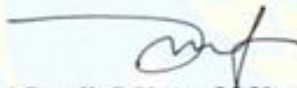
DR. Soffiana Agustin, S.Kom, M.Kom
NIP. 197 711 292 005 012 001

Penguji 2 (Sekretaris)



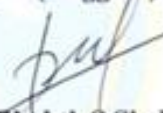
Harunur Rosvid, S.T., M.Kom., Ph.D.
NIP. 0621 0408 106

Penguji 3 (Anggota)



Deni Sutaji, S.Kom., M.Kom
NIP. 06 211 211 153

Penguji 4 (Anggota)



Henny Dwi Bhakti, S.Si., M.Si.
NIP. 06 211 709 200

Mengetahui,

Dekan Fakultas Teknik
Universitas Muhammadiyah Gresik



Harunur Rosvid, S.T., M.Kom., Ph.D.
NIP. 0621 0408 106

Ketua Program Studi Teknik Informatika
Universitas Muhammadiyah Gresik



Henny Dwi Bhakti, S.Si., M.Si.
NIP. 06 211 709 200

PERNYATAAN KEASLIAN SKRIPSI

Yang bertanda tangan dibawah ini:

Nama : Herlando Prayitno
NIM : 210602011
Fakultas/Program Studi : Teknik/Teknik Informatika

Menyatakan dengan sebenarnya bahwa skripsi yang berjudul “ **OPTIMASI PENJADWALAN RUANGAN DI GEDUNG DAKWAH MUHAMMADIYAH CABANG DUKUN MENGGUNAKAN ALGORITMA GENETIKA**” yang saya buat untuk melengkapi persyaratan menjadi Sarjana Komputer pada Program Studi Teknik Informatika Fakultas Teknik Universitas Muhammadiyah Gresik, bukan merupakan tiruan atau duplikasi dari skripsi yang sudah dipublikasikan atau pernah dipakai untuk untuk mendapatkan gelar kesarjanaan di lingkungan Universitas Muhammadiyah Gresik maupun di perguruan tinggi atau instansi manapunkecuali bagian yang secara tertulis dikutip dalam naskah ini dan disebutkan dalam sumber kutipan atau daftar pustaka.

Apabila dikemudian hari saya terbukti melanggar pernyataan saya tersebut diatas, maka saya bersedia untuk mempertanggung jawabkan sesuai peraturan yang berlaku.

Gresik, 08 Januari 2025


Herlando Prayitno
NIM.210602011

KATA PENGANTAR

Puji syukur kehadiran Tuhan Yang Maha Esa, berkat rahmat dan nikmat yang telah Dia berikan kepada penulis sehingga dapat menyelesaikan proposal skripsi dengan judul “OPTIMASI PENJADWALAN RUANGAN DI GEDUNG DAKWAH MUHAMMADIYAH CABANG DUKUN MENGGUNAKAN ALGORITMA GENETIKA” dengan lancar. Penulisan skripsi ini disusun sebagai syarat untuk melaksanakan sidang skripsi di jurusan Teknik Informatika, Fakultas Teknik di Universitas Muhammadiyah Gresik.

Pada proses penyusunan skripsi ini, penulis tidak dapat menyelesaikannya tanpa bantuan dari berbagai pihak. Oleh karena itu pada kesempatan ini, penulis juga mengucapkan terima kasih kepada:

1. Allah SWT, yang senantiasa memberikan kesehatan, kesabaran, dan ketabahan pada penulis.
2. Keluarga besar penulis, khususnya Bapak Suyitno dan Ibu Endang Nurhayati sebagai orang tua, serta saudara, ponakan, dan seluruh anggota keluarga lainnya yang selalu memberikan dukungan, semangat, dan doa dalam proses penyelesaian penelitian ini.
3. Ibu Nadhirotul Laily, S.Psi., M.Psi., Ph.D, Psikolog, selaku rektor Universitas Muhammadiyah Gresik.
4. Bapak Harunur Rosyid, S.T., M.Kom, Ph.D, selaku Dekan Fakultas Teknik.
5. Ibu Henny Dwi Bhakti, S.Si., M.Si, selaku Kepala Program Studi Teknik Informatika.
6. Bapak Harunur Rosyid, S.T., M.Kom, Ph.D dan Ibu Henny Dwi Bhakti, S.Si., M.Si., selaku pembimbing yang telah memberikan bimbingan dan arahan dalam penyelesaian skripsi ini.
7. Seluruh dosen Teknik Informatika Universitas Muhammadiyah Gresik yang telah memberikan ilmu melalui mata kuliah yang diajarkan.
8. Rekan-rekan seperjuangan mahasiswa teknik informatika angkatan 2021 dan Himpunan Mahasiswa Teknik Informatika Universitas Muhammadiyah Gresik yang selalu bersemangat, saling mendukung,

dan berkontribusi bersama dalam mewujudkan visi dan misi selama menempuh pendidikan di Teknik Informatika.

9. Semua pihak yang tidak dapat penulis sebutkan satu persatu yang telah memberikan bantuan dan dukungan serta do'a selama menjalani perkuliahan.

Penulis menyadari bahwa laporan proposal skripsi ini masih jauh dari kata kesempurnaan. Oleh karena itu penulis mengharapkan kritik dan saran yang sifatnya membangun bagi penulis, pembaca, dan masyarakat sehingga laporan ini dapat bermanfaat.

Gresik, 08 Januari 2025

Herlando Prayitno

DAFTAR ISI

HALAMAN JUDUL.....	i
LEMBAR PERSETUJUAN SKRIPSI.....	ii
LEMBAR PENGESAHAN SKRIPSI.....	iii
LEMBAR KEASLIAN SKRIPSI	iv
KATA PENGANTAR	v
DAFTAR ISI.....	vii
DAFTAR GAMBAR	ix
DAFTAR TABEL.....	xi
DAFTAR PERSAMAAN	xiii
DAFTAR LAMPIRAN.....	xiv
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian.....	3
1.6 Metodologi Penelitian	3
1.7 Sistematika Penulisan.....	5
BAB 2 TINJAUAN PUSTAKA	6
2.1 Optimasi	6
2.2 Penjadwalan (<i>schedulling</i>).....	7
2.3 Algoritma Genetika	7
2.4 Review Artikel.....	11
BAB 3 ANALISIS DAN PERANCANGAN SISTEM	17
3.1 ANALISIS SISTEM.....	17
3.2 HASIL ANALISIS SISTEM.....	17
3.3 REPRESENTASI MODEL	26
3.4 Perancangan Sistem.....	39
3.4.1 Diagram Konteks	39
3.4.2 Diagram Jenjang Sistem	40
3.4.3 Data Flow Diagram (DFD) Sistem	41

3.5 Perancangan Basis Data	44
3.5.1 Tabel User.....	44
3.5.2 Tabel Pengguna (GEN 1).....	44
3.5.3 Tabel Ruang (GEN 2)	45
3.5.4 Tabel Waktu (GEN 3).....	45
3.5.5 Tabel Kromosom Populasi Awal.....	46
3.5.6 Tabel Kromosom Baru.....	46
3.5.7 Tabel Parameter	47
3.6 Perancangan Antarmuka Sistem.....	48
3.6.1 Halaman Login	48
3.6.2 Halaman Dashboard.....	49
3.6.3 Halaman Input Pengguna.....	49
3.6.4 Halaman Proses Algoritma Genetika.....	49
3.7 Spesifikasi Pengembangan Sistem	50
3.8 Skenario Pengujian Sistem	51
BAB 4 IMPLEMENTASI DAN PENGUJIAN SISTEM	53
4.1 Implementasi Sistem	53
4.1.1 Implementasi Metode Algoritma Genetika.....	53
4.2 Pengujian Sistem	59
4.2.1 Antarmuka Halaman Login	59
4.2.2 Halaman Dashboard.....	60
4.2.3 Halaman Data Pengguna.....	60
4.2.4 Halaman <i>Generate</i> Jadwal Algoritma Genetika	61
4.2.5 Halaman Jadwal.....	62
4.3 Hasil Analisis Sistem.....	62
BAB 5 PENUTUP.....	66
5.1 Kesimpulan.....	66
5.2 Saran.....	66
DAFTAR PUSTAKA	68
LAMPIRAN	71

DAFTAR GAMBAR

Gambar 2.1 Diagram Alir Algoritma Genetika	8
Gambar 3.1 Diagram Alir Algoritma Genetika	18
Gambar 3.2 Diagram Populasi Awal	19
Gambar 3.3 Diagram Alir <i>Fitness</i>	19
Gambar 3.4 Diagram Alir Seleksi	20
Gambar 3.5 Diagram Alir <i>Crossover</i>	21
Gambar 3.6 Diagram Alir Mutasi	21
Gambar 3.7 <i>Probabilitas</i> Kumulatif dalam <i>Roulette wheel</i>	32
Gambar 3.8 Diagram Konteks Sistem Penjadwalan Ruangan menggunakan Algoritma Genetika	39
Gambar 3.9 Diagram Berjenjang Sistem Penjadwalan Ruangan menggunakan Algoritma Genetika	40
Gambar 3.10 DFD Level 1 Sistem Penjadwalan Ruangan di Gedung Dakwah Muhammadiyah Cabang Dukun	41
Gambar 3.11 DFD Level 2 Sistem Penjadwalan Ruangan menggunakan Algoritma Genetika	43
Gambar 3.12 ERD Sistem Penjadwalan Ruangan menggunakan Algoritma Genetika	48
Gambar 3.13 Halaman Login	48
Gambar 3.14 Halaman Dashboard	49
Gambar 3.15 Halaman Input Pengguna	49
Gambar 3.16 Halaman Proses Algoritma Genetika Admin	50
Gambar 3.17 Halaman Hasil Penjadwalan	50
Gambar 4.1 Halaman Login	59

Gambar 4.2 Halaman Dashboard.....	60
Gambar 4.3 Halaman Pengguna	61
Gambar 4.4 Halaman <i>Generate</i> Jadwal.....	61
Gambar 4.5 Halaman Jadwal.....	62

DAFTAR TABEL

Tabel 2.1 Review Artikel	12
Tabel 3.1 Data Lembaga.....	22
Tabel 3.2 Data Kegiatan.....	22
Tabel 3.3 Data Hari	23
Tabel 3.4 Data Jam.....	23
Tabel 3.5 Data Ruangan	23
Tabel 3.6 Data Kelompok Pengguna.....	24
Tabel 3.7 Data Kelompok Ruangan	24
Tabel 3.8 Data Kelompok Waktu.....	25
Tabel 3.9 Matriks Populasi Awal	25
Tabel 3.10 Individu Ke-1 Pembentukan Populasi Awal	27
Tabel 3.11 Individu Ke-2 Pembentukan Populasi Awal	28
Tabel 3.12 Individu Ke-3 Pembentukan Populasi Awal	28
Tabel 3.13 Individu Ke-4 Pembangkitan Populasi Awal	29
Tabel 3.14 Individu Ke-5 Pembangkitan Populasi Awal	29
Tabel 3.15 Hitung Nilai Probabilitas.....	31
Tabel 3.16 Hitung Nilai Komulatif	31
Tabel 3.17 Populasi Baru Hasil Seleksi	33
Tabel 3.18 Populasi Baru dengan Induk.....	33
Tabel 3.19 Proses <i>Crossover</i> 1	34
Tabel 3.20 Proses <i>Crossover</i> 2	35
Tabel 3.21 Proses <i>Crossover</i> 3	35
Tabel 3.22 Proses <i>Crossover</i> 4	35

Tabel 3.23 Sebelum Mutasi.....	37
Tabel 3.24 Sesudah Mutasi.....	38
Tabel 3.25 Tabel User	44
Tabel 3.26 Tabel Lembaga	44
Tabel 3.27 Tabel Kegiatan	45
Tabel 3.28 Tabel Ruangan.....	45
Tabel 3.29 Tabel Hari.....	45
Tabel 3.30 Tabel Jam	46
Tabel 3.31 Tabel kromosom_populasi_awal.....	46
Tabel 3.32 Tabel kromosom_baru.....	46
Tabel 3.33 Tabel Parameter.....	47
Tabel 3.34 Tabel Skenario Pengujian.....	52
Tabel 3.35 Hasil Pengujian.....	52
Tabel 4.1 Source Code Pembentukan Populasi Awal	53
Tabel 4.2 Source Code Mencari Nilai Fitness.....	55
Tabel 4.3 Source Code Proses Seleksi	56
Tabel 4.4 Source Code Proses <i>Crossover</i>	57
Tabel 4.5 Source Code Mutasi	58
Tabel 4.6 Skenario Pengujian.....	63
Tabel 4.7 Hasil Pengujian.....	63

DAFTAR PERSAMAAN

(2.1)	9
(2.2)	10

DAFTAR LAMPIRAN

Lampiran 1 Data Penjadwalan	71
Lampiran 2 Source Code Login	71
Lampiran 3 Source Code Dashboard.....	73
Lampiran 4 Source Code Penjadwalan	73
Lampiran 5 Source Code Generate dan Jadwal.....	75
Lampiran 6 Data 5 Pengujian 1	77
Lampiran 7 Data 5 Pengujian 2	77
Lampiran 8 Data 5 Pengujian 3	78
Lampiran 9 Data 5 Pengujian 4.....	78
Lampiran 10 Data 5 Pengujian 5	78
Lampiran 11 Data 10 Pengujian 1	78
Lampiran 12 Data 10 Pengujian 2	79
Lampiran 13 Data 10 Pengujian 3	79
Lampiran 14 Data Pengujian 4	79
Lampiran 15 Data 10 Pengujian 5	80
Lampiran 16 Data 15 Pengujian 1	80
Lampiran 17 Data 15 Pengujian 2	80
Lampiran 18 Data 15 Pengujian 3	81
Lampiran 19 Data 15 Pengujian 4	81
Lampiran 20 Data 15 Pengujian 5	81

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Jadwal adalah susunan rencana kegiatan yang disusun berdasarkan pembagian waktu tertentu dan dilengkapi dengan rincian pelaksanaan yang detail. Jadwal berfungsi sebagai panduan untuk mengatur dan memastikan setiap kegiatan dapat terlaksana sesuai dengan waktu yang telah ditentukan (Informatics and Info 2024). Gedung Dakwah Muhammadiyah Cabang Dukun (GDMCD) merupakan fasilitas yang dikelola oleh Pimpinan Cabang Muhammadiyah Dukun untuk mendukung kegiatan dakwah, pendidikan, dan sosial. Proses penjadwalan ruangan di GDMCD dikelola oleh staf administrasi melalui dua cara, yaitu pengajuan langsung ke kantor administrasi atau melalui pesan WhatsApp resmi. Setiap pengajuan penggunaan ruangan harus menyertakan informasi berupa waktu pelaksanaan, jenis kegiatan, jumlah peserta, dan identitas lembaga pengguna. Staf administrasi akan memproses pengajuan tersebut dengan memeriksa ketersediaan ruangan sesuai dengan waktu yang diajukan. Beberapa kegiatan seperti diskusi kelompok, rapat internal, dan acara individu tidak memerlukan surat resmi, sedangkan untuk kegiatan formal seperti pengajian, pelatihan, dan acara kemuhammadiyah diwajibkan melampirkan surat permohonan resmi sebagai syarat tambahan. Setelah permohonan diterima, staf administrasi mengevaluasi ketersediaan ruangan dengan memeriksa jadwal yang dicatat menggunakan buku catatan dan papan pengumuman.

Permasalahan yang terjadi diawali dengan faktor yakni pencatatan yang tidak konsisten dalam penjadwalan, seperti adanya permohonan yang telah disetujui melalui WhatsApp, Namun belum diperbarui dalam catatan utama. Setiap pengajuan penggunaan ruangan harus mempertimbangkan berbagai faktor seperti waktu, jenis kegiatan, jumlah peserta, dan identitas lembaga pengguna. Proses ini sering menimbulkan bentrokan jadwal akibat ketidakkonsistenan prosedur administratif dan kendala komunikasi. Selain itu, penggunaan buku catatan dan papan pengumuman untuk mencatat jadwal menyulitkan pemantauan ketersediaan

ruangan. Hal ini menunjukkan perlunya sistem yang mendukung untuk mengoptimalkan proses penggunaan ruangan dan menghindari konflik jadwal.

Algoritma Genetika (GA) adalah metode *heuristik* yang meniru mekanisme seleksi alam dan evolusi, di mana individu dengan nilai *fitness* tinggi memiliki peluang lebih besar untuk bertahan melalui *reproduksi*, *crossover*, dan mutasi (Andriyadi and Halimah 2022). Penelitian oleh (Pangestu, Suryawan, and Latipah 2023) menunjukkan bahwa GA efektif dalam menghasilkan jadwal yang memenuhi batasan tertentu, seperti larangan bagi guru untuk mengajar lebih dari satu kelas pada waktu yang sama dan larangan bagi kelas untuk berbagi waktu pelajaran yang sama. Penelitian oleh (Kristanti et al. 2022) juga mengeksplorasi variasi parameter dalam algoritma genetika dan dampaknya terhadap optimasi jadwal. Hasilnya menunjukkan bahwa aplikasi penjadwalan berbasis GA dapat meminimalkan bentrokan jadwal dan meningkatkan pengelolaan waktu di kampus, seperti yang dibuktikan di IBI Kosgoro 1957 Jakarta. Namun, beberapa aturan masih memerlukan optimasi, terutama terkait efisiensi waktu dalam menghasilkan jadwal. Beberapa pengujian aturan diterima, tetapi penggabungan semua aturan ditolak karena mencapai batas maksimum generasi (Syawal, Lestari, and Rachman 2021).

Penyelesaian masalah penjadwalan di GDMCD dapat dioptimalkan menggunakan Algoritma Genetika (GA) yang berfungsi untuk mencari kandidat urutan kegiatan dan menghindari bentrokan jadwal (Rinaldi and Rismayadi 2022). Metode ini dirancang untuk menangani jadwal yang kompleks dengan berbagai variabel dan batasan. Dalam penerapannya, jadwal direpresentasikan sebagai kromosom yang diolah melalui proses seleksi, *crossover*, dan mutasi untuk menemukan solusi yang optimal. Pendekatan ini memungkinkan GA menghindari bentrokan jadwal dan meningkatkan efisiensi penggunaan ruangan, sehingga dapat mengatasi kendala administratif dan komunikasi yang sering terjadi.

1.2 Rumusan Masalah

Rumusan masalah pada penelitian ini adalah Bagaimana menentukan optimasi penjadwalan ruangan di Gedung Dakwah Muhammadiyah Cabang Dukun?

1.3 Batasan Masalah

Adapun batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Penelitian fokus pada optimasi penjadwalan ruangan di Gedung Dakwah Muhammadiyah Cabang Dukun.
2. Pengguna fasilitas dibatasi hanya untuk organisasi dan lembaga di daerah Kecamatan Dukun.
3. Sistem hanya menangani penjadwalan ruangan, tidak termasuk pengelolaan fasilitas lainnya.
4. *Output* yang dihasilkan adalah menemukan jadwal penggunaan ruangan yang optimal pada Gedung Dakwah Muhammadiyah Cabang Dukun.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah Menentukan jadwal penggunaan ruangan yang optimal di Gedung Dakwah Muhammadiyah Cabang Dukun menggunakan metode Algoritma Genetika.

1.5 Manfaat Penelitian

Adapun manfaat dalam penelitian ini adalah sebagai berikut:

1. Memberikan solusi optimasi dalam pengelolaan jadwal penggunaan ruangan di Gedung Dakwah Muhammadiyah Cabang Dukun.
2. Pengguna dapat mengajukan jadwal dengan mudah melalui sistem tanpa harus datang langsung atau menunggu konfirmasi dari administrasi.

1.6 Metodologi Penelitian

Metode yang digunakan dalam perancangan sistem antara lain:

1. Studi Literatur

Studi Literatur ini dilakukan untuk mendapatkan informasi dari mengumpulkan dan mempelajari literature melalui buku serta karya ilmiah dan sumber informasi lain yang terkait dengan masalah yang dibahas.

2. Tahap Pengumpulan Data

Pada tahapan pengumpulan data meliputi studi pustaka yang berhubungan dengan konsep dan teori metode Algoritma Genetika dengan menggunakan bahasa pemrograman PHP *framework Laravel* dan basis data *MySQL*, serta melakukan observasi untuk mendapatkan data – data yang diperlukan pada penelitian.

3. Tahap Preprocessing Data

Tahap ini bertujuan mempersiapkan data untuk diolah menggunakan Algoritma Genetika. Proses ini memastikan bobot setiap kriteria sesuai dan direpresentasikan sebagai kromosom, sehingga mendukung langkah seleksi, *crossover*, dan mutasi dalam mencari solusi optimal.

4. Analisis Sistem

Analisis sistem dilakukan berdasarkan hasil observasi dan pengumpulan data. Analisis kebutuhan sistem dilakukan untuk menentukan kebutuhan sistem.

5. Implementasi

Pada tahap ini merupakan proses penerjemahan dari tahap perancangan ke dalam bentuk aplikasi dengan menggunakan bahasa pemrograman PHP *Framework Laravel* dan basis data *MySQL*.

6. Pengujian

Tahapan selanjutnya program yang dibangun akan di uji untuk mengetahui kinerja sistem dan keakuratan metode yang diterapkan sehingga mampu menghasilkan informasi sesuai dengan yang diharapkan.

7. Tahap Penyusunan Laporan

Pada tahap ini melakukan penyusunan laporan dari penelitian berdasarkan sistematika penulisan.

1.7 Sistematika Penulisan

Dalam sistematika penulisan laporan skripsi ini disajikan ke dalam beberapa bab, dan setiap bab disajikan dalam beberapa sub bab. Sistematika penulisan laporan skripsi ini bertujuan untuk memudahkan dalam memahami maksud dan tujuan dari penelitian ini. Hal tersebut diantaranya sebagai berikut:

BAB I : PENDAHULUAN

Bab ini membahas tentang latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian dan sistematika penulisan laporan skripsi.

BAB II : LANDASAN TEORI

Bab ini membahas dan menjelaskan teori – teori dasar yang berhubungan dengan permasalahan yang diambil, seperti penjelasan yang berhubungan dengan metode Algoritma Genetika (GA)

BAB III : ANALISIS DAN PERANCANGAN SISTEM

Bab ini menjelaskan tentang analisa kebutuhan dan perancangan dari sistem. Meliputi analisis sistem, pembuatan Diagram Konteks, DFD (*Data Flow Diagram*), serta langkah-langkah yang akan dilakukan untuk menyelesaikan permasalahan dan mencapai tujuan yang telah ditetapkan.

BAB IV : IMPLEMENTASI DAN PENGUJIAN SISTEM

Bab ini menjelaskan tentang pengujian sistem secara umum maupun terperinci mengenai hasil penerapan sistem pada objek penelitian.

BAB V : PENUTUP

Bab ini berisi kesimpulan penelitian dan saran peneliti untuk pengembangan lebih lanjut.

BAB 2

TINJAUAN PUSTAKA

2.1 Optimasi

Menurut Kamus Besar Bahasa Indonesia (Optimasi), optimasi diartikan sebagai optimalisasi, yaitu proses, cara, dan kreasi untuk menghasilkan hal-hal terkini. Sedangkan optimasi juga berasal dari kamus bahasa Inggris yaitu *Optimization* yang artinya optimasi. Menurut Wikipedia Bahasa Indonesia, Optimasi adalah suatu proses untuk mencapai hasil yang ideal atau optimasi (nilai efektif yang dapat dicapai). Optimasi dapat diartikan sebagai suatu bentuk mengoptimalkan sesuatu hal yang sudah ada, ataupun merancang dan membuat sesuatu secara optimal. Menurut Esther (Azzahra et al. 2019) Pengertian Optimasi adalah pencapaian sesuatu keadaan yang paling baik, atau pencapaian solusi untuk suatu masalah yang ditunjukkan pada batas maksimum dan minimum. Secara matematis, optimasi adalah pencarian nilai ekstrim suatu fungsi. Dalam masalah penjadwalan, misalnya, tujuan optimasi bisa berupa pengurangan waktu tunggu (*idle time*) atau pemanfaatan sumber daya yang lebih efisien. Oleh karena itu, optimasi dapat diterapkan untuk menyelesaikan masalah dengan batasan tertentu, seperti kendala waktu, kapasitas ruang, atau jumlah sumber daya (Winston, 2004).

Optimasi secara umum berarti memaksimalkan atau mengoptimalkan sesuatu dengan tujuan mengelola sesuatu yang telah dilakukan. Oleh karena itu, optimasi merupakan kata benda yang berasal dari kata kerja, dan bergantung pada tujuan yang perlu dimaksimalkan, optimasi dapat dianggap sebagai ilmu dan seni. Contoh masalah yang dimaksimalkan adalah masalah keuntungan, dan contoh masalah yang diminimalkan adalah masalah biaya, persediaan, dan sebagainya. Kendala yang sering ditemui antara lain terbatasnya bahan baku, terbatasnya tenaga kerja, dan lain-lain. Kendala tersebut dapat dinyatakan dalam bentuk himpunan persamaan *linier* atau pertidaksamaan variabel. Oleh karena itu, fungsi yang dioptimalkan adalah solusi dengan nilai fungsi tujuan yang diinginkan. Nilai yang diinginkan adalah nilai maksimum yaitu nilai maksimum fungsi tujuan, dan nilai minimum yaitu nilai minimum fungsi tujuan.

2.2 Penjadwalan (*schedulling*)

Pengertian jadwal menurut kamus besar bahasa Indonesia adalah pembagian waktu berdasarkan rencana pengaturan urutan kerja, daftar atau tabel kegiatan atau rencana kegiatan dengan pembagian waktu pelaksanaan yang terperinci, sedangkan pengertian penjadwalan adalah proses, cara, perbuatan menjadwalkan atau memasukkan dalam jadwal. (Tim Penyusun Kamus Pusat Pembinaan dan Pengembangan Bahasa Kamus Besar Bahasa Indonesia, 1997). Pengertian penjadwalan secara umum dapat diartikan seperti “*scheduling is the allocation of resources overtime to perform collection of risk*“, yang artinya penjadwalan adalah pengalokasian sumber daya yang terbatas untuk mengerjakan sejumlah pekerjaan. Permasalahan muncul apabila pada tahapan operasi tertentu beberapa atau seluruh pekerjaan itu membutuhkan stasiun kerja yang sama. Dengan dilakukannya pengurutan pekerjaan ini unit-unit produksi (*resources*) dapat dimanfaatkan secara optimum. Pemanfaatan ini antara dilakukan dengan jalan meningkatkan utilitas unit-unit produksi melalui usaha- usaha mereduksi waktu menganggur (*idle time*) dari unit-unit yang bersangkutan. Pemanfaatan lainnya dapat juga dilakukan dengan cara meminimumkan *in- process* inventory melalui reduksi terhadap waktu rata-rata pekerjaan yang menunggu (antri) dalam baris antrian pada unit-unit produksi.

Penjadwalan (*scheduling*) menurut Conway adalah pengurutan produk secara menyeluruh yang dikerjakan oleh beberapa buah mesin. Sedangkan menurut (Baker and Trietsch 1974), penjadwalan didefinisikan sebagai proses pengalokasian sumber daya untuk memilih sekumpulan tugas dalam jangka waktu tertentu.

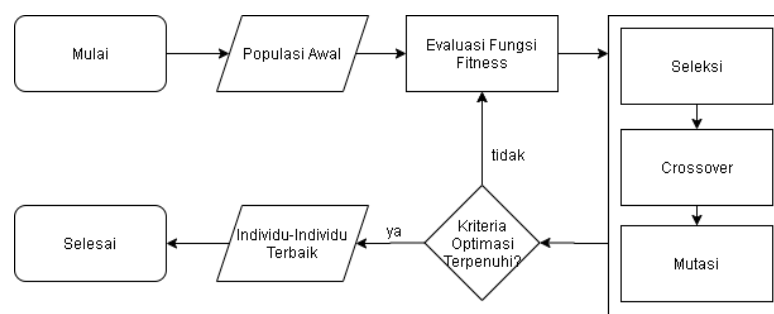
2.3 Algoritma Genetika

Algoritma genetika merupakan algoritma yang dikembangkan dari proses pencarian solusi yang menggunakan pencarian secara random atau acak. Algoritma ini dikembangkan dari proses evolusi. Pada proses evolusi individu akan secara terus menerus mengalami perubahan gen untuk menyesuaikan dengan kondisi yang ada. Hanya individu yang terkuat yang akan bertahan sehingga dalam proses evolusi ini akan menghasilkan individu yang terbaik. Algoritma genetika merupakan sebuah metode pencarian yang telah disesuaikan dengan proses

genetika dari *organisme* – organisme biologi yang berdasarkan pada teori revolusi Charles Dharwin.

Algoritma genetika pertamakali di perkenalkan oleh Jhon Holland pada tahun 1970. algoritma genetika menerapkan pemahaman mengenai teori evolusi untuk mengatasi permasalahan yang terjadi. Pendekatan yang dilakukan oleh algoritma ini adalah dengan menggabungkan secara acak berbagai solusi dari sekumpulan solusi untuk mendapatkan generasi solusi terbaik kemudian di nilai untuk memaksimalkan kecocokannya dan menghasilkan solusi dengan fitness terbaik untuk mendapatkan generasi baru. Generasi selanjutnya akan mempresentasikan perbaikan – perbaikan dari generasi sebelumnya. Individu menyatakan satu solusi yang mungkin. Individu bias dikatakan sama dengan kromosom, yang merupakan kumpulan gen.

Algoritma ini dapat mengoptimumkan solusi dari berbagai permasalahan yang dihadapi, salah satunya adalah optimasi pada sistem penjadwalan. Sistem penjadwalan yang dibuat mengikuti diagram alir siklus algoritma genetika yaitu membangkitkan populasi awal, evaluasi *fitness*, seleksi individu, *crossover*, mutasi, dan *regenerasi* (Mone and Simarmata 2021). Secara umum tahapan dari algoritma genetika dapat dijabarkan dengan alur bagan. Hal tersebut dapat dilihat pada gambar 2.1.



Gambar 2.1 Diagram Alir Algoritma Genetika

Beberapa definisi penting yang perlu diperhatikan dalam algoritma genetika adalah sebagai berikut:

1. Populasi Awal

Pada algoritma genetika, populasi awal adalah kumpulan individu yang menyimbolkan possibility dari solusi pada masalah yang akan dipecahkan (Padaka, Tetik, and Ledi 2023), dimana tahapan awal dari algoritma genetika yaitu membangkitkan secara acak solusi atau pemecahan masalah yang disebut dengan *initial population*.

Setiap kromosom terdiri dari serangkaian gen yang merepresentasikan aktivitas atau tugas yang akan dijadwalkan. Setiap gen pada kromosom mewakili suatu *karakteristik* atau atribut dari solusi yang akan dihasilkan. Dalam algoritma genetika gen bisa berupa bilangan biner, *float*, *integer* maupun karakter. Pada penelitian ini, populasi awal dibangun dengan menggunakan bilangan *random* atau metode acak dengan memperhatikan *range* bilangan yang telah ditentukan (Syawal et al. 2021) Dengan demikian, setiap jadwal atau kromosom dalam populasi awal akan memiliki urutan tugas yang berbeda-beda dan dapat mencakup berbagai kemungkinan solusi untuk masalah penjadwalan yang sedang dihadapi.

2. Evaluasi Fungsi *Fitness*

Dalam algoritma genetika, populasi awal merupakan sekumpulan individu yang merepresentasikan kemungkinan solusi untuk suatu masalah yang ingin dipecahkan (Pangestu et al. 2023). Untuk mengukur kualitas kromosom pada penjadwalan genetika, digunakan suatu fungsi *fitness* yang menentukan seberapa baik sebuah jadwal yang direpresentasikan oleh kromosom dapat memenuhi kriteria-kriteria yang ditetapkan dalam masalah penjadwalan yang diberikan (Hidayat et al. 2019). *Fitness* tersebut sesuai dengan persamaan 2.1.

$$Fitness = \frac{1}{1+penalty} \quad (2.1)$$

Dari persamaan diatas, nilai fitness ditentukan oleh nilai penalty yang memperhitungkan batasan atau constraint pada masalah yang akan dipecahkan. Jika solusi yang dihasilkan melanggar batasan, maka akan dikenakan penalty yang akan mempengaruhi nilai fitness. Semakin sedikit jumlah pelanggaran atau nilai penalty, maka semakin tinggi nilai fitness yang diperoleh. Semakin tinggi nilai fitness, maka semakin baik kualitas jadwal yang dihasilkan (Kurniati, Rahmatulloh, and Rahmawati 2019).

3. Proses Seleksi

Seleksi individu pada suatu populasi pada dasarnya adalah proses pemilihan individu dengan nilai *probabilitas* tinggi yang memiliki kemungkinan lebih besar untuk dipilih pada tahap selanjutnya (Suzanti and Mufarroha 2021). Dalam penelitian ini, dipilih metode seleksi roda *roulette* untuk melakukan seleksi individu dalam populasi. proses seleksi roda *roulette* dimulai dengan menghitung total nilai *fitness* dari seluruh individu dalam populasi. Kemudian, setiap individu diberi jatah ruang di roda *roulette* berdasarkan proporsi nilai *fitness* nya terhadap total nilai *fitness* populasi. Untuk mencari *fitness* relatif setiap kromosom dapat dihitung menggunakan persamaan 2.2.

$$Prob = \frac{fitness(i)}{total\ fitness} \quad (2.2)$$

4. Proses Crossover

Persilangan (*Crossover*) Pertukaran gen untuk meningkatkan nilai adaptif dikenal sebagai persilangan. Pertama, angka acak akan dihasilkan sebagai nilai desimal antara 0 dan 1. Kemudian dibuat perbandingan antara angka acak ini dengan probabilitas silang yang telah ditentukan oleh pengguna sebelum proses perencanaan dimulai (Rizki, Hendriyani, and Novaliendry 2023). Terjadi pertukaran data antara populasi saat ini dan

populasi berikutnya jika bilangan acak lebih kecil dari *probabilitas* silang yang ditentukan:

Sebelum Crossover

Individu 1 = (2, 2, 3, 4)

Individu 2 = (4, 4, 0, 1)

Setelah Crossover

Individu 1 = (2, 2, 0, 1)

Individu 2 = (4, 4, 3, 4)

5. Proses Mutasi

Mutasi adalah proses menciptakan gen baru dengan mengubah gen individu. Proses ini serupa dengan perubahan yang terjadi pada kehidupan alami (Muhandhis et al. 2023). Kemungkinan terjadinya mutasi genetik sangat rendah, serupa dengan kejadian nyata dalam kehidupan. Oleh karena itu, ada kemungkinan terjadinya mutasi genetik, pada tingkat yang rendah. Mutasi yang digunakan pada penelitian ini adalah mutasi titik acak (*random point mutation*). Pada proses mutasi ini, satu kromosom dipilih secara acak dari populasi dan salah satu gen dalam kromosom tersebut diubah nilainya secara acak.

2.4 Review Artikel

Pencarian literatur berfungsi sebagai kerangka teoretis untuk *refleksi* dan berfungsi sebagai gambaran umum dan referensi untuk penelitian ini, maka dengan demikian menyajikan temuan serupa yang dilakukan selama ini disajikan untuk menghindari plagiat. Tabel 2.1 di bawah ini memuat penelitian-penelitian sebelumnya yang terkait dengan survei-survei yang dilakukan, yaitu:

Tabel 2.1 *Review Artikel*

NO	Landasan Literatur	Metode	Masalah	Hasil Penelitian
1	“Penerapan Algoritma Genetika Dalam Penjadwalan Mata Pelajaran” (Pangestu et al. 2023)	Algoritma Genetika (GA)	Masalah yang dihadapi dalam penjadwalan mata pelajaran di SD Kreatif Muhammadiyah 2 Bontang meliputi batasan hard constraint, di mana seorang guru tidak boleh mengajar di dua waktu dan hari yang sama, dan kelas tidak boleh memiliki dua waktu pelajaran yang sama pada satu hari. Selain itu, kompleksitas pengorganisasian data yang melibatkan banyak variabel seperti mata pelajaran, guru, kelas, dan waktu pelajaran juga menjadi tantangan dalam proses penjadwalan.	Hasil penelitian menunjukkan bahwa penerapan algoritma genetika berhasil menghasilkan jadwal yang memenuhi semua batasan yang ditetapkan, serta mampu mengoptimalkan penggunaan waktu dan mengurangi konflik jadwal. Penelitian ini membuktikan bahwa algoritma genetika dapat meningkatkan efisiensi dalam proses belajar mengajar di SD Kreatif Muhammadiyah 2 Bontang, dengan solusi penjadwalan yang lebih terstruktur dan terorganisir.
2	“Aplikasi Algoritma Genetika Dalam Penjadwalan Mata Kuliah” (Mone and Simarmata 2021)	Algoritma Genetika (GA)	Masalah utama yang diidentifikasi dalam penelitian ini adalah waktu yang dibutuhkan oleh tenaga administrasi untuk menyusun jadwal mata kuliah, yang sering kali melebihi satu hari. Selain itu, terdapat kendala seperti bentrok ruang dan waktu, bentrok dosen, serta kebutuhan untuk mengakomodasi preferensi dosen dan jadwal praktikum di laboratorium	Hasil dari penelitian menunjukkan bahwa penerapan algoritma genetik dalam pembuatan jadwal kuliah dapat meningkatkan efisiensi proses penjadwalan. Sistem yang dirancang mampu mengurangi waktu yang diperlukan untuk menyusun jadwal secara signifikan dan memberikan solusi yang lebih optimal untuk masalah penjadwalan yang kompleks, sehingga meningkatkan kualitas dan pelayanan pendidikan
3	“Optimasi Penjadwalan Proyek Dengan Metode	Algoritma Genetika (GA)	Masalah yang diidentifikasi dalam penelitian ini adalah ketidakefisienan dalam proses penjadwalan proyek yang masih dilakukan secara manual, yang mengakibatkan waktu yang lama dalam pembuatan jadwal dan sering terjadinya bentrokan antara kegiatan. Perusahaan yang diteliti belum menggunakan aplikasi khusus untuk penjadwalan, sehingga	Hasil dari penelitian ini menunjukkan bahwa dengan penerapan Program Optimasi Penjadwalan berbasis web menggunakan Algoritma Genetika, proses pembuatan jadwal menjadi lebih efisien dan tidak memerlukan waktu yang lama. Program ini mampu menghasilkan jadwal yang optimal, mengurangi

NO	Landasan Literatur	Metode	Masalah	Hasil Penelitian
	Algoritma Genetika” (Rinaldi and Rismayadi 2022)		proses yang ada tidak optimal dan menyita banyak konsentrasi	kemungkinan bentrokan, dan meningkatkan produktivitas tim proyek. Kesimpulan ini menegaskan bahwa penggunaan teknologi dalam penjadwalan dapat memberikan solusi yang signifikan terhadap masalah yang ada
4	“Implementasi Algoritma Genetika Untuk Penjadwalan Laboratorium Fakultas Ilmu Komputer Universitas Muslim Indonesia” (Syawal et al. 2021)	Algoritma Genetika (GA)	Masalah utama yang dihadapi dalam penelitian ini adalah kompleksitas penjadwalan laboratorium yang melibatkan banyak variabel, seperti menghindari tabrakan jadwal dengan kelas lain, memastikan pengajar tidak mengajar bersamaan, dan menyediakan ruangan yang sesuai untuk matakuliah lab. Selain itu, semua mata kuliah harus diajarkan sesuai dengan dosen dan semester yang berlaku, yang menambah tingkat kesulitan dalam proses penjadwalan.	Hasil penelitian menunjukkan bahwa sistem yang dikembangkan berhasil menghasilkan jadwal dengan memperhatikan delapan aturan yang ditetapkan. Namun, terdapat beberapa aturan yang memerlukan optimasi, terutama dalam hal efisiensi waktu yang dibutuhkan untuk menghasilkan jadwal. Beberapa pengujian terhadap aturan yang diterapkan diterima, sementara penggabungan semua aturan ditolak karena mencapai batas maksimum generasi.
5	“Optimasi Naive Bayes Menggunakan Algoritma Genetika Pada Klasifikasi Komentar Cyberbullying	Algoritma Naive Bayes (ANB) dan Algoritma Genetika (GA)	Masalah yang diangkat dalam penelitian ini adalah meningkatnya kasus <i>cyberbullying</i> di media sosial, yang menjadi perhatian serius di Indonesia. Dengan meningkatnya jumlah pengguna internet, tantangan untuk mendeteksi dan mengklasifikasikan komentar yang mengandung unsur bullying juga meningkat. Penelitian ini bertujuan untuk mengatasi masalah tersebut dengan mengembangkan model yang dapat secara otomatis	Hasil penelitian menunjukkan bahwa model <i>Naive Bayes</i> yang dioptimasi dengan algoritma genetika mencapai akurasi 77,34%, presisi 73,79%, recall 98,17%, dan skor F1 84,25%. Ini menunjukkan peningkatan kinerja yang signifikan dibandingkan dengan model <i>Naive Bayes</i> tanpa optimasi, yang hanya mencapai akurasi 73,09%. Hasil ini menandakan bahwa optimasi dengan algoritma

NO	Landasan Literatur	Metode	Masalah	Hasil Penelitian
	Pada Media Sosial X” (Tahir and Sugianto 2024)		mengidentifikasi komentar bullying, sehingga dapat membantu dalam upaya pencegahan dan penanganan <i>cyberbullying</i> .	genetika efektif dalam meningkatkan kemampuan model dalam mendeteksi dan mengklasifikasikan komentar <i>cyberbullying</i> secara lebih akurat.
6	“Penerapan Algoritma Genetika Dan Jaringan Syaraf Tiruan Dalam Penjadwalan Mata Kuliah Studi Kasus : Prodi Sistem Informasi Universitas Pamulang” (Hartono and Zein 2023)	Algoritma Genetika (GA) dan Jaringan Syaraf Tiruan (JST)	Penelitian ini menghadapi beberapa masalah yang signifikan dalam penjadwalan mata kuliah. Salah satu masalah utama adalah konflik jadwal, di mana terdapat bentrokan antara waktu mata kuliah yang memerlukan ruang yang sama. Selain itu, keterbatasan ruang kelas yang tersedia menjadi tantangan tersendiri, di mana peminjaman ruang harus diatur secara efisien. Selain itu, preferensi dosen dan kebutuhan mahasiswa juga harus dipertimbangkan dalam proses penjadwalan, sehingga menciptakan kebutuhan untuk sistem yang dapat menyesuaikan dengan berbagai kendala ini.	Hasil penelitian menunjukkan bahwa kombinasi algoritma genetika dan jaringan syaraf tiruan berhasil meningkatkan efisiensi proses penjadwalan mata kuliah. Penelitian ini berhasil mengurangi konflik dan tabrakan dalam jadwal, serta memungkinkan penyesuaian penjadwalan yang lebih baik terhadap kebutuhan dosen dan mahasiswa. Selain itu, penerapan teknik-teknik ini juga meningkatkan pengalaman belajar mahasiswa dengan memanfaatkan ruang yang lebih optimal. Dengan demikian, penelitian ini memberikan kontribusi yang signifikan terhadap pengelolaan penjadwalan di institusi pendidikan
7	“Penerapan Algoritma Genetika Dalam Mengatasi Jadwal Mengajar	Algoritma Genetika (GA)	Masalah utama yang dihadapi dalam penelitian ini adalah pengelolaan data penjadwalan mata kuliah yang masih dilakukan secara manual, yang sering mengakibatkan ketidaksesuaian jadwal dan bentrokan antara jadwal mengajar dosen. Dengan banyaknya kelas dan dosen, serta adanya keterbatasan ruang, diperlukan sistem yang dapat	Hasil penelitian menunjukkan bahwa aplikasi penjadwalan yang dirancang dengan menggunakan algoritma genetika berhasil menghasilkan jadwal yang optimal dan efisien. Sistem ini mampu meminimalkan bentrokan jadwal dan meningkatkan kualitas pengelolaan waktu di kampus. Pengujian sistem juga menunjukkan bahwa aplikasi dapat

NO	Landasan Literatur	Metode	Masalah	Hasil Penelitian
	Yang Bentrok Pada Program Studi Informatika Ibi Kosgoro 1957 Jakarta Indonesia” (Firmansyah et al. 2021)		mengelola dan mengoptimalkan penjadwalan agar semua kegiatan dapat berjalan dengan baik tanpa konflik.	berfungsi dengan baik sesuai dengan analisis dan perancangan yang telah dilakukan, memberikan solusi yang efektif untuk masalah penjadwalan di IBI Kosgoro 1957 Jakarta.
8	“Optimalisasi Sistem Penjadwalan Mata Kuliah Jurusan Bahasa Inggris Menggunakan Algoritma Genetika (Studi Kasus Politeknik Negeri Sriwijaya)” (Kristanti et al. 2022)	Algoritma Genetika (GA)	Masalah utama yang dihadapi dalam penyusunan jadwal mata kuliah di jurusan Bahasa Inggris adalah ketidakefektifan proses manual yang dilakukan oleh Sekretaris Jurusan. Proses ini sering kali memakan waktu lama dan berisiko tinggi terhadap kesalahan, seperti bentrok antara mata kuliah, dosen, dan ruang kelas. Oleh karena itu, diperlukan sistem yang lebih efisien dan terkomputerisasi untuk mengatasi masalah ini.	Hasil penelitian menunjukkan bahwa sistem penjadwalan yang dibangun dengan algoritma genetika berhasil menghasilkan jadwal tanpa bentrok antar mata kuliah, dosen, dan waktu perkuliahan. Pengujian dengan metode UAT menunjukkan bahwa semua halaman dalam sistem berfungsi dengan baik. Kombinasi parameter algoritma genetika yang digunakan, seperti <i>probabilitas crossover</i> 0.70 dan <i>probabilitas</i> mutasi 0.40, terbukti efektif dalam menghasilkan jadwal yang optimal. Hasil jadwal dapat diunduh dalam format Excel dan PDF, memberikan kemudahan bagi pengguna.

NO	Landasan Literatur	Metode	Masalah	Hasil Penelitian
9	“Optimasi Penjadwalan Mata Kuliah Menggunakan Metode Algoritma Genetika Dengan Teknik Tournament Selection” (Sari et al. 2019)	Algoritma Genetika (GA)	Masalah utama yang dihadapi dalam penyusunan jadwal perkuliahan adalah kesulitan dalam mengalokasikan mata kuliah, dosen, dan ruangan tanpa terjadi bentrok. Proses manual yang dilakukan saat ini memakan waktu lama dan sering kali tidak efisien, sehingga diperlukan sistem otomatis yang dapat mengatasi kendala-kendala tersebut dan menghasilkan jadwal yang optimal.	Hasil penelitian menunjukkan bahwa penerapan algoritma genetika dapat menghasilkan jadwal perkuliahan yang sesuai dengan batasan-batasan yang telah ditentukan. Pengujian model menunjukkan waktu dan akurasi yang baik, serta kemampuan sistem untuk memenuhi hard constraints yang ada. Dengan demikian, penelitian ini membuktikan bahwa metode algoritma genetika efektif dalam optimasi penjadwalan mata kuliah di perguruan tinggi.
10	“Penerapan Algoritma Genetika untuk Penjadwalan Mata Pelajaran” (Ardiansyah and Junianto 2022)	Algoritma Genetika (GA)	Masalah utama yang dihadapi oleh SD <i>Lazuardi Global Compassionate School</i> adalah proses penjadwalan yang masih dilakukan secara manual, yang memakan waktu lama dan sering menghasilkan ketidaksesuaian dalam jadwal mata pelajaran. Proses ini tidak hanya memerlukan waktu yang cukup lama, tetapi juga berpotensi menyebabkan bentrokan antara jadwal mata pelajaran dan ketersediaan guru serta ruang kelas. Hal ini menunjukkan perlunya sistem penjadwalan yang lebih efisien dan terstruktur.	Hasil penelitian menunjukkan bahwa penerapan algoritma genetika dalam penjadwalan mata pelajaran dapat meningkatkan efisiensi proses penjadwalan secara signifikan. Dengan menggunakan aplikasi yang dibangun berdasarkan algoritma ini, waktu yang diperlukan untuk menyusun jadwal dapat dihemat, dan ketidaksesuaian dalam jadwal dapat diminimalkan. Penelitian ini membuktikan bahwa algoritma genetika adalah solusi yang efektif untuk mengatasi masalah penjadwalan di lembaga pendidikan.

BAB 3

ANALISIS DAN PERANCANGAN SISTEM

3.1 ANALISIS SISTEM

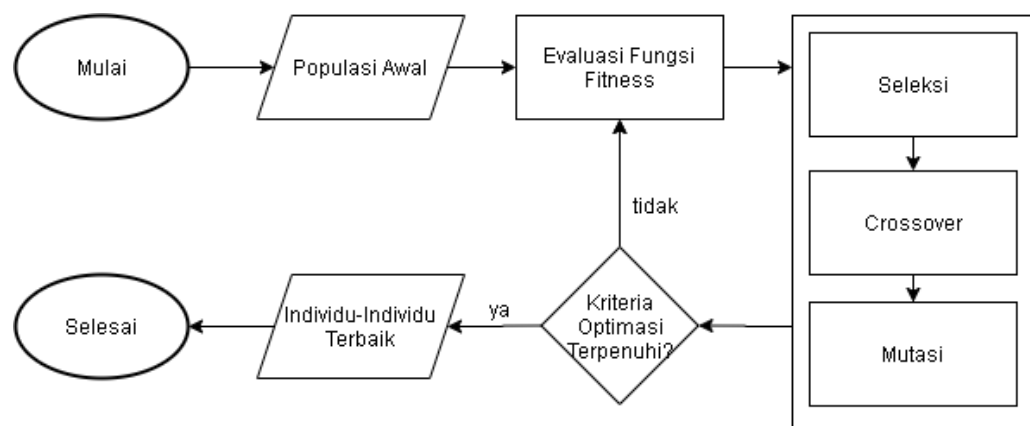
Penjadwalan ruangan di Gedung Dakwah Muhammadiyah Cabang Dukun (GDMCD) dikelola oleh staf administrasi melalui dua cara, yaitu pengajuan langsung ke kantor administrasi atau melalui pesan WhatsApp resmi. Setiap pengajuan harus menyertakan informasi berupa waktu pelaksanaan, jenis kegiatan, jumlah peserta, dan identitas lembaga pengguna. Staf administrasi akan memproses permohonan tersebut dengan memeriksa ketersediaan ruangan sesuai dengan waktu yang diajukan. Setelah permohonan diterima, staf administrasi mengevaluasi ketersediaan ruangan dengan memeriksa jadwal yang dicatat menggunakan buku catatan dan papan pengumuman.

Permasalahan yang terjadi diawali dengan pencatatan yang tidak konsisten dalam mengatur penjadwalan, seperti adanya permohonan yang telah disetujui melalui WhatsApp namun belum diperbarui dalam catatan utama. Setiap permintaan penggunaan ruangan harus mempertimbangkan berbagai faktor seperti waktu, jenis kegiatan, jumlah peserta, dan identitas lembaga peminjam. Proses ini sering menimbulkan bentrokan jadwal akibat ketidakkonsistenan prosedur administratif dan kendala komunikasi. Selain itu, penggunaan buku catatan dan papan pengumuman untuk mencatat jadwal menyulitkan pemantauan ketersediaan ruangan. Masalah utama dalam penjadwalan GDMCD yaitu belum adanya sistem penjadwalan yang mampu menentukan jadwal yang optimal dalam penggunaan ruangan, untuk memastikan efisiensi jadwal bagi pengguna. Karena jadwal yang disusun secara tidak efisien dapat menimbulkan konflik dan tumpang tindih antar pengguna. Dari permasalahan tersebut peneliti membuat Sistem Penjadwalan Ruangan dengan menggunakan metode Algoritma Genetika.

3.2 HASIL ANALISIS SISTEM

Penelitian ini menggunakan Algoritma Genetika untuk menyelesaikan masalah penjadwalan dalam proses penggunaan ruangan di Gedung Dakwah Muhammadiyah Cabang Dukun (GDMCD). Algoritma ini dipilih karena

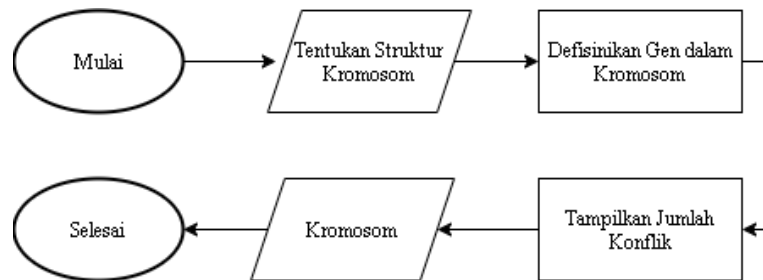
kemampuannya dalam meningkatkan efisiensi proses penjadwalan. Penerapan Algoritma Genetika dalam menentukan jadwal penggunaan ruangan di GDMCD diharapkan dapat memberikan kontribusi yang signifikan terhadap optimalisasi pemanfaatan ruangan. Keputusan menggunakan Algoritma Genetika didasarkan pada kemudahan penerapannya serta hasil optimal yang dapat dicapai. Dalam penelitian ini, dikembangkan sebuah sistem penjadwalan ruangan di GDMCD dengan menggunakan metode Algoritma Genetika.



Gambar 3.1 Diagram Alir Algoritma Genetika

Pada Gambar 3.1 menunjukan bagian yang dikurung kotak dalam diagram menunjukkan tiga operator utama dalam algoritma genetika. Seleksi adalah proses memilih individu terbaik berdasarkan nilai *Fitness* untuk dipertahankan atau direproduksi. Selanjutnya, *Crossover* atau persilangan dilakukan dengan menggabungkan sebagian gen dari dua individu induk untuk menghasilkan individu baru (keturunan). Terakhir, mutasi dilakukan dengan mengubah gen secara acak pada individu baru untuk menjaga keragaman solusi dan mencegah konvergensi prematur. Ketiga operator ini bekerja secara berurutan dalam setiap generasi untuk mencari solusi optimal.

a. Populasi Awal

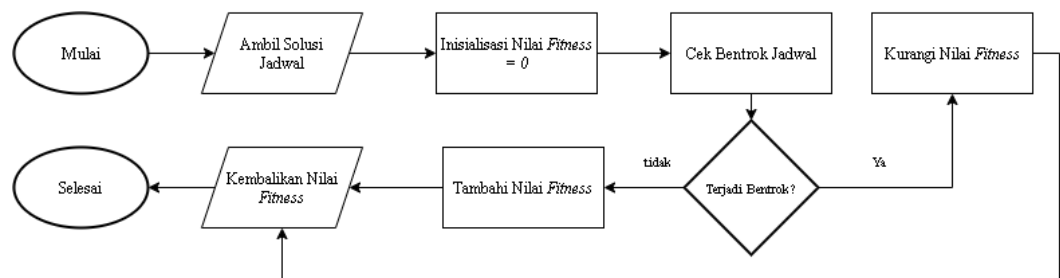


Gambar 3.2 Diagram Populasi Awal

Proses awal dari algoritma genetika dapat dilihat pada gambar 3.2. Diagram ini merupakan proses pembentukan populasi awal. Penjelasan langkah-langkahnya adalah sebagai berikut:

1. Menentukan struktur kromosom sesuai dengan permasalahan.
2. Mendefinisikan gen dalam kromosom untuk merepresentasikan elemen jadwal.
3. Memasukkan data pengguna, ruangan, dan waktu sebagai dasar pembentukan jadwal.
4. Membuat kromosom awal secara acak untuk membentuk populasi awal.

b. Fungsi *Fitness*

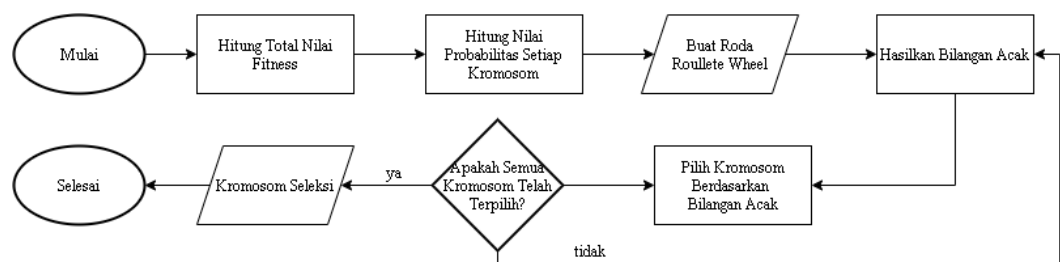


Gambar 3.3 Diagram Alir *Fitness*

Proses evaluasi *fitness* dapat dilihat pada gambar 3.3. Diagram ini menjelaskan langkah-langkah yang dilakukan untuk menghitung nilai *fitness* setiap individu dalam populasi. Tahapan prosesnya adalah sebagai berikut:

1. Mengambil solusi jadwal yang akan dievaluasi.
2. Menginisialisasi nilai *Fitness* dengan 0.
3. Memeriksa bentrok jadwal dalam solusi tersebut.
4. Mengurangi nilai *Fitness* jika terjadi bentrok.
5. Menambahkan nilai *Fitness* jika tidak terjadi bentrok.
6. Mengembalikan nilai *Fitness* akhir sebagai hasil evaluasi.

c. Seleksi

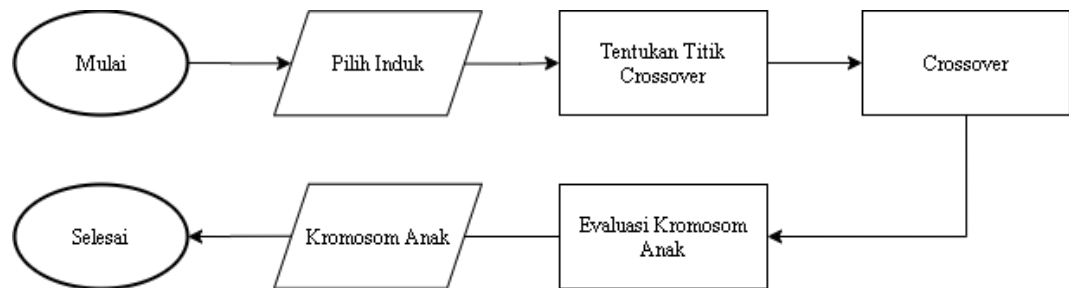


Gambar 3.4 Diagram Alir Seleksi

Proses seleksi dapat dilihat pada gambar 3.4. Diagram ini menunjukkan bagaimana proses seleksi dilakukan dalam algoritma genetika untuk memilih kromosom terbaik berdasarkan nilai *Fitness*. Penjelasan langkah-langkahnya adalah sebagai berikut:

1. Menghitung total nilai *Fitness* dari semua kromosom.
2. Menghitung probabilitas seleksi untuk setiap kromosom berdasarkan nilai *Fitness*.
3. Membuat roda *roulette* sebagai mekanisme seleksi berbasis *probabilitas*.
4. Menghasilkan bilangan acak untuk memilih kromosom.
5. Memilih kromosom berdasarkan bilangan acak hingga semua kromosom terpilih.
6. Menyimpan kromosom hasil seleksi untuk tahap mutasi berikutnya.

d. Crossover

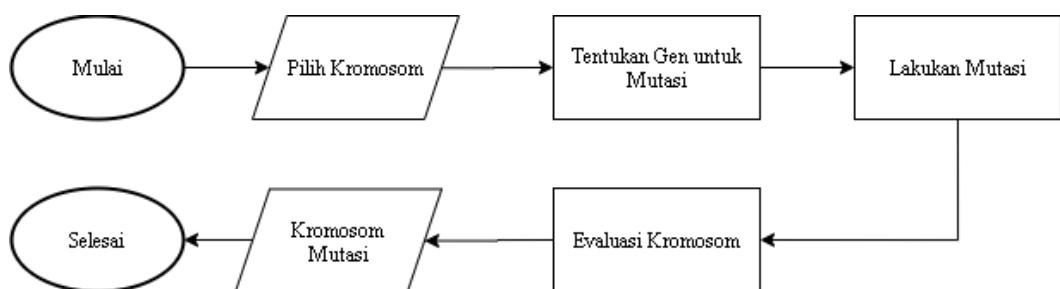


Gambar 3.5 Diagram Alir *Crossover*

Proses *Crossover* dapat dilihat pada gambar 3.5. Diagram ini menjelaskan proses pembuatan kromosom baru (anak) dari dua kromosom induk. Penjelasan langkah-langkahnya adalah sebagai berikut:

1. Memilih dua kromosom induk untuk proses persilangan.
2. Menentukan titik *Crossover* sebagai tempat penggabungan elemen kromosom.
3. Melakukan *Crossover* untuk menghasilkan kromosom anak.
4. Mengevaluasi kromosom anak untuk memastikan validitasnya.
5. Menyimpan kromosom anak ke dalam populasi baru.

e. Mutasi



Gambar 3.6 Diagram Alir Mutasi

Proses mutasi dapat dilihat pada gambar 3.6, Diagram ini menjelaskan perubahan kecil pada kromosom untuk menjaga keberagaman populasi. Penjelasan langkah-langkahnya adalah sebagai berikut:

1. Memilih kromosom dari populasi untuk proses mutasi.

2. Menentukan gen dalam kromosom yang akan dimutasi.
3. Melakukan perubahan pada gen sesuai mekanisme mutasi.
4. Mengevaluasi kromosom hasil mutasi untuk memastikan validitasnya.
5. Menyimpan kromosom hasil mutasi ke dalam populasi.
6. Menyimpan kromosom anak ke dalam populasi baru.

Berdasarkan analisis sistem yang dilakukan, asumsi-asumsi yang digunakan dalam proses penjadwalan penggunaan ruangan di GDMCD adalah sebagai berikut:

Tabel 3.1 Data Lembaga

TABEL LEMBAGA			
NO	Id_lembaga	NAMA LEMBAGA	JUMLAH PESERTA
1	MTsMTQ	MTs Maarif Tanwirul Qulub	90
2	MTsMH	MTs Mathlabul Huda	250
3	MTsM10	MTs Muhammadiyah 10	190
4	MTsIU	MTs Ihyaul Ulum	220
5	MTsTA	MTs Tarbiyatul Athfal	130
...
10	MTsM1	MTs Muhammadiyah 1	220

Nama-nama lembaga yang terlibat dalam penelitian, beserta jumlah peserta dari masing-masing lembaga, dapat dilihat pada tabel 3.1. Data ini digunakan untuk mengelompokkan peserta berdasarkan lembaga asalnya. Pengelompokan ini menjadi bagian dari variabel GEN 1 dalam proses algoritma genetika.

Tabel 3.2 Data Kegiatan

TABEL KEGIATAN		
NO	id_kegiatan	KEGIATAN
1	kg-01	Pusat Dakwah
2	kg-02	Pendidikan
3	kg-03	Sosial
4	kg-04	Rapat dan Koordinasi
5	kg-05	Umum/Non Lembaga

Nama-nama kegiatan yang terlibat dalam penelitian, dapat dilihat pada tabel 3.2. Data ini digunakan untuk mengelompokkan kegiatan berdasarkan kategori atau jenisnya. Pengelompokan ini menjadi bagian dari variabel GEN 1 dalam proses algoritma genetika.

Tabel 3.3 Data Hari

TABEL HARI		
NO	id_hari	HARI
1	h1	SENIN
2	h2	SELASA
3	h3	RABU
4	h4	KAMIS
5	h5	JUM'AT

Informasi mengenai hari-hari kegiatan yang terlibat dalam penelitian, dapat dilihat pada tabel 3.2. Data ini digunakan untuk mengelompokkan waktu berdasarkan hari, yang menjadi bagian dari variabel GEN 3 dalam proses algoritma genetika.

Tabel 3.4 Data Jam

TABEL JAM		
NO	id_jam	JAM
1	1	08:00 - 12:00
2	2	13:00 - 17:00

Informasi mengenai dua sesi waktu dalam satu hari, yang digunakan dalam penelitian, dapat dilihat pada tabel 3.4. Data ini digunakan untuk mengelompokkan waktu ke dalam sesi tertentu, yang juga menjadi bagian dari variabel GEN 3 dalam proses algoritma genetika.

Tabel 3.5 Data Ruangan

TABEL RUANGAN			
NO	KODE RUANGAN	RUANGAN	KAPASITAS
1	r1	Lantai 1 - Ruang 1	50
2	r2	Lantai 1 - Ruang 2	100

3	r3	Lantai 1 - Ruang 3	150
4	r4	Lantai 1 - Ruang 4	250
5	r5	Lantai 1 - Ruang 5	400

Daftar ruangan yang tersedia untuk kegiatan, lengkap dengan informasi kapasitas dapat dilihat pada tabel 3.5. Data ini menjadi bagian dari variabel GEN 2 dalam proses algoritma genetika.

Tabel 3.6 Data Kelompok Pengguna

TABEL PENGGUNA (GEN 1)			
NO	KEGIATAN	LEMBAGA	PESERTA
1	kg-01	MTsMTQ	90
2	kg-02	MTsMH	250
3	kg-03	MTsM10	190
4	kg-04	MTsIU	220
5	kg-05	MTsTA	130
...
10	kg-05	MTsMI	220

Pada tabel 3.6, dapat dilihat terdapat satu entitas pengguna yang terdiri atas lembaga dengan jumlah peserta spesifik yang terlibat dalam kegiatan tertentu. Variabel ini menjadi bagian dari pengelompokan GEN 1 dalam proses algoritma genetika.

Tabel 3.7 Data Kelompok Ruangan

TABEL RUANGAN (GEN 2)		
NO	RUANGAN	KAPASITAS
1	r1	50
2	r2	100
3	r3	150
4	r4	250
5	r5	400

Daftar ruangan yang tersedia untuk kegiatan, dengan informasi tambahan kapasitas dan kode identifikasi ruangan dapat dilihat pada tabel 3.7.

Variabel ini tetap menjadi bagian penting dalam pengelompokan GEN 2 dalam proses algoritma genetika.

Tabel 3.8 Data Kelompok Waktu

TABEL WAKTU (GEN 3)		
NO	HARI	JAM
1	SENIN	1
2	SENIN	2
3	SELASA	1
4	SELASA	2
5	RABU	1
6	RABU	2
7	KAMIS	1
8	KAMIS	2
9	JUM'AT	1
10	JUM'AT	2

Kombinasi hari dan sesi waktu yang tersedia untuk pelaksanaan kegiatan dapat dilihat pada tabel 3.8. Data ini menjadi bagian dari variabel GEN 3 dalam proses algoritma genetika.

Tabel 3.9 Matriks Populasi Awal

Populasi Awal dan Hasil Matrik					
Kromosom	idv 1	idv 2	idv 3	idv 4	idv 5
k1	[1, 3, 8]	[1, 5, 5]	[1, 5, 7]	[1, 4, 2]	[1, 3, 7]
k2	[2, 2, 4]	[2, 4, 10]	[2, 3, 6]	[2, 2, 5]	[2, 1, 4]
k3	[3, 4, 2]	[3, 4, 3]	[3, 4, 4]	[3, 1, 1]	[3, 4, 3]
k4	[4, 5, 3]	[4, 3, 6]	[4, 2, 3]	[4, 2, 3]	[4, 2, 2]
k5	[5, 4, 2]	[5, 5, 2]	[5, 4, 7]	[5, 4, 4]	[5, 3, 1]
k6	[6, 2, 3]	[6, 3, 2]	[6, 2, 5]	[6, 2, 7]	[6, 2, 10]
k7	[7, 5, 6]	[7, 3, 1]	[7, 3, 9]	[7, 5, 10]	[7, 3, 3]
k8	[8, 3, 4]	[8, 3, 5]	[8, 1, 4]	[8, 3, 4]	[8, 4, 1]
k9	[9, 3, 1]	[9, 2, 6]	[9, 4, 10]	[9, 2, 1]	[9, 5, 2]
k10	[10, 1, 9]	[10, 4, 8]	[10, 1, 1]	[10, 1, 1]	[10, 3, 4]

Matriks populasi awal dalam algoritma genetika ini terdiri dari 10 kromosom (k1-k10) dan 5 individu (idv 1-5) seperti pada tabel 3.9, dimana

setiap sel berisi kromosom dengan format [GEN1, GEN2, GEN3] yang merepresentasikan pengguna_id (1-10), ruangan_id (1-5), dan waktu_id (1-10). Pembangkitan nilai dilakukan secara acak dengan GEN1 tetap berurutan 1-10 untuk setiap kromosom, sementara GEN2 dan GEN3 dibatasi sesuai jumlah ruangan dan slot waktu yang tersedia. Sebagai contoh, kromosom k1 pada idv 1 [1,3,8] menunjukkan pengguna 1 mendapat ruangan 3 pada waktu 8, dan kromosom k4 pada idv 3 [4,2,3] berarti pengguna 4 mendapat ruangan 2 pada waktu 3. Total terdapat 50 kombinasi jadwal yang menyediakan beragam solusi awal untuk proses evolusi, membentuk basis untuk operasi genetika selanjutnya, dan memungkinkan eksplorasi berbagai kombinasi jadwal yang akan diproses melalui tahapan algoritma genetika (seleksi, crossover, mutasi) untuk menghasilkan solusi penjadwalan yang optimal.

3.3 REPRESENTASI MODEL

Algoritma genetika terdiri dari empat proses utama. Pertama, inisialisasi awal digunakan menemukan solusi awal untuk masalah. Kemudian, proses reproduksi yang mencakup *mutation* dan *Crossover*, menghasilkan keturunan baru. Selanjutnya, evaluasi dilakukan untuk mengukur nilai kecocokan setiap Kromosom dalam populasi. Terakhir, seleksi digunakan untuk memilih Kromosom baru berdasarkan nilai kecocokan, dengan memprioritaskan mereka yang memiliki nilai kecocokan yang paling tinggi.

a. Inisialisasi Parameter Algoritma Genetika

Langkah awal algoritme genetika, kita menentukan parameter-parameter seperti jumlah kromosom, jumlah individu, jumlah generasi, *crossover rate*, dan mutasi *rate*. Parameter yang digunakan dalam penelitian ini untuk menyelesaikan masalah dengan algoritma genetika adalah:

- Jumlah Kromosom : 10
- Jumlah Individu : 5
- Jumlah Generasi : 1
- *Crossover Rate* : 0,75

- *Mutation Rate* : 0,20

b. Mendefinisikan Gen

Gen dalam konteks ini merupakan gabungan dari beberapa variabel yang merepresentasikan penggunaan ruangan yang diajukan melalui sistem, di mana Gen 1, Gen 2, dan Gen 3 memiliki peran tersendiri. Representasi kelompok gen berdasarkan **Tabel 3.6, Tabel 3.7, dan Tabel 3.8** adalah sebagai berikut:

Gen 1 = Kelompok Pengguna [Kegiatan, Lembaga, dan Jumlah Peserta]

Gen 2 = Kelompok Ruangan [Ruangan dan Kapasitas]

Gen 3 = Kelompok Waktu [Hari dan Jam]

Pada satu kromosom, terdapat tiga gen yang terdiri dari 1 hingga 3. Gen-gen ini membentuk suatu urutan jadwal yang tidak teratur. Proses optimasi dilakukan untuk menemukan susunan jadwal yang paling optimal.

c. Membangkitkan Populasi Awal

Pada penelitian ini, diasumsikan bahwa setiap populasi awal akan terdiri dari 10 kromosom yang berbeda. Pada tahap ini, populasi awal dibuat dengan mengacak gen2 dan gen3 untuk menciptakan variasi pada populasi awal. Agar mempermudah proses implementasi algoritma genetika, peneliti menggunakan data dengan skala kecil sebagai acuan. Berikut adalah tabel yang menampilkan proses inisialisasi populasi:

Tabel 3.10 Individu Ke-1 Pembentukan Populasi Awal

Kro	Gen 1			Gen 2		Gen 3	
	Nama Kegiatan	Lembaga	Jumlah Peserta	Ruangan	Kapasitas	Hari	Jam
1	KG-01	MTsMTQ	90	r3	150	KAMIS	2
2	KG-02	MTsMH	250	r2	100	SELASA	2
3	KG-03	MTsM10	190	r4	200	SENIN	2
4	KG-04	MTsIU	220	r5	400	SELASA	1
5	KG-05	MTsTA	130	r4	200	SENIN	2

Kro	Gen 1			Gen 2		Gen 3	
	Nama Kegiatan	Lembaga	Jumlah Peserta	Ruangan	Kapasitas	Hari	Jam
6	KG-01	MTsNR	30	r2	100	SELASA	1
7	KG-02	MTsNH	90	r5	400	RABU	2
8	KG-03	MTsYKUISG	80	r3	150	SELASA	2
9	KG-04	MTsAK1	40	r3	150	SENIN	1
10	KG-05	MTsM1	220	r1	50	JUM'AT	1

Keterangan Pinalti: Lembaga Menggunakan Ruangannya yang Tidak Sesuai Kapasitasnya dan Ruang Konflik

Tabel 3.11 Individu Ke-2 Pembentukan Populasi Awal

Kro	Gen 1			Gen 2		Gen 3	
	Nama Kegiatan	Lembaga	Jumlah Peserta	Ruangan	Kapasitas	Hari	Jam
1	KG-01	MTsMTQ	90	r5	400	RABU	1
2	KG-02	MTsMH	250	r4	250	JUM'AT	2
3	KG-03	MTsM10	190	r4	200	SELASA	1
4	KG-04	MTsIU	220	r3	150	RABU	2
5	KG-05	MTsTA	130	r5	400	SENIN	2
6	KG-01	MTsNR	30	r3	150	SENIN	2
7	KG-02	MTsNH	90	r3	150	SENIN	1
8	KG-03	MTsYKUISG	80	r3	150	RABU	1
9	KG-04	MTsAK1	40	r2	100	RABU	2
10	KG-05	MTsM1	220	r4	200	KAMIS	2

Keterangan Pinalti: Lembaga Menggunakan Ruangannya yang Tidak Sesuai Kapasitasnya

Tabel 3.12 Individu Ke-3 Pembentukan Populasi Awal

Kro	Gen 1			Gen 2		Gen 3	
	Nama Kegiatan	Lembaga	Jumlah Peserta	Ruangan	Kapasitas	Hari	Jam
1	KG-01	MTsMTQ	90	r5	400	KAMIS	1
2	KG-02	MTsMH	250	r3	150	RABU	2
3	KG-03	MTsM10	190	r4	200	SELASA	2
4	KG-04	MTsIU	220	r2	100	SELASA	1
5	KG-05	MTsTA	130	r4	200	KAMIS	1
6	KG-01	MTsNR	30	r2	100	RABU	1
7	KG-02	MTsNH	90	r3	150	JUM'AT	1
8	KG-03	MTsYKUISG	80	r1	50	SELASA	2
9	KG-04	MTsAK1	40	r4	200	JUM'AT	2

Kro	Gen 1			Gen 2		Gen 3	
	Nama Kegiatan	Lembaga	Jumlah Peserta	Ruangan	Kapasitas	Hari	Jam
10	KG-05	MTsM1	220	r1	50	SENIN	1

Keterangan Pinalti: Lembaga Menggunakan Ruangan yang Tidak Sesuai Kapasitasnya

Tabel 3.13 Individu Ke-4 Pembangkitan Populasi Awal

Kro	Gen 1			Gen 2		Gen 3	
	Nama Kegiatan	Lembaga	Jumlah Peserta	Ruangan	Kapasitas	Hari	Jam
1	KG-01	MTsMTQ	90	r4	200	SENIN	2
2	KG-02	MTsMH	250	r2	100	RABU	1
3	KG-03	MTsM10	190	r1	50	SENIN	1
4	KG-04	MTsIU	220	r2	100	SELASA	1
5	KG-05	MTsTA	130	r4	200	SELASA	2
6	KG-01	MTsNR	30	r2	100	KAMIS	1
7	KG-02	MTsNH	90	r5	400	JUM'AT	2
8	KG-03	MTsYKUISG	80	r3	150	SELASA	2
9	KG-04	MTsAK1	40	r2	100	SENIN	1
10	KG-05	MTsM1	220	r1	50	SENIN	1

Keterangan Pinalti: Lembaga Menggunakan Ruangan yang Tidak Sesuai Kapasitasnya dan Ruang Konflik

Tabel 3.14 Individu Ke-5 Pembangkitan Populasi Awal

Kro 5	Gen 1			Gen 2		Gen 3	
	Nama Kegiatan	Lembaga	Jumlah Peserta	Ruangan	Kapasitas	Hari	Jam
1	KG-01	MTsMTQ	90	r3	150	KAMIS	1
2	KG-02	MTsMH	250	r1	50	SELASA	2
3	KG-03	MTsM10	190	r4	250	SELASA	1
4	KG-04	MTsIU	220	r2	100	SENIN	2
5	KG-05	MTsTA	130	r3	150	SENIN	1
6	KG-01	MTsNR	30	r2	100	JUM'AT	2
7	KG-02	MTsNH	90	r3	150	SELASA	1
8	KG-03	MTsYKUISG	80	r4	250	SENIN	1
9	KG-04	MTsAK1	40	r5	400	SENIN	2
10	KG-05	MTsM1	220	r3	150	SELASA	2

Keterangan Pinalti: Lembaga Menggunakan Ruangan yang Tidak Sesuai Kapasitasnya

d. Menentukan Nilai *Fitness*

Dalam penelitian ini, nilai *Fitness* dihitung berdasarkan tingkat bentrok atau konflik pada setiap kromosom dalam individu. Bentrok dihitung dengan mempertimbangkan batasan yang telah ditentukan sebelumnya pada bagian 3.2. Jika batasan-batasan tersebut terpenuhi, maka nilai konflik pada individu akan ditambah 1. Setelah melakukan pengecekan terhadap pelanggaran pada setiap gen dalam kromosom, hasil pelanggaran tersebut akan digunakan untuk menghitung nilai *Fitness* guna mencari tau seberapa optimal solusi yang dihasilkan dari individu yang ada. Perhitungan nilai *Fitness* dapat dihitung dengan persamaan 2.1 berikut:

$$F = \frac{1}{1+(KR+KK)}$$

Keterangan:

$F = \text{Fitness}$

KR = Konflik Ruang

KK = Konflik Kapasitas

Berikut perhitungan nilai *Fitness* berdasarkan konflik yang telah didapatkan pada populasi awal:

1	$F \text{ Individu 1} = \frac{1}{1 + (1 + 2)} =$	0.250
2	$F \text{ Individu 2} = \frac{1}{1 + (0 + 2)} =$	0.333
3	$F \text{ Individu 3} = \frac{1}{1 + (0 + 4)} =$	0.200
4	$F \text{ Individu 4} = \frac{1}{1 + (1 + 4)} =$	0.167
5	$F \text{ Individu 5} = \frac{1}{1 + (0 + 3)} =$	0.250
<i>Total Nilai fitness =</i>		1.200

e. Proses Seleksi

Metode seleksi yang digunakan pada penelitian ini adalah metode seleksi *roulette wheel*. Pada metode ini, setiap individu akan mendapatkan

proporsi dalam lingkaran berdasarkan nilai *Fitness*-nya (Elva 2019). Proses seleksi pada *roulette wheel* dibagi menjadi beberapa proses yaitu:

a. Menghitung Nilai Probabilitas

Nilai *Fitness* dari setiap individu akan dibagi dengan total nilai *Fitness* yang telah dihitung sebelumnya. Proses perhitungan nilai probabilitas ini dapat dilihat pada rumus persamaan 2.2 berikut:

Tabel 3.15 Hitung Nilai Probabilitas

Kromosom	Probabilitas	Hasil Probabilitas
1	0.250/1.200	0.208
2	0.333/1.200	0.278
3	0.200/1.200	0.167
4	0.167/1.200	0.139
5	0.250/1.200	0.208

Pada tabel 3.15 Menjelaskan nilai probabilitas dihitung dengan membagi nilai *Fitness* setiap kromosom dengan total nilai *Fitness*. Hasil *probabilitas* ini akan digunakan dalam proses seleksi *roulette-wheel* selanjutnya.

b. Menghitung Nilai Komulatif

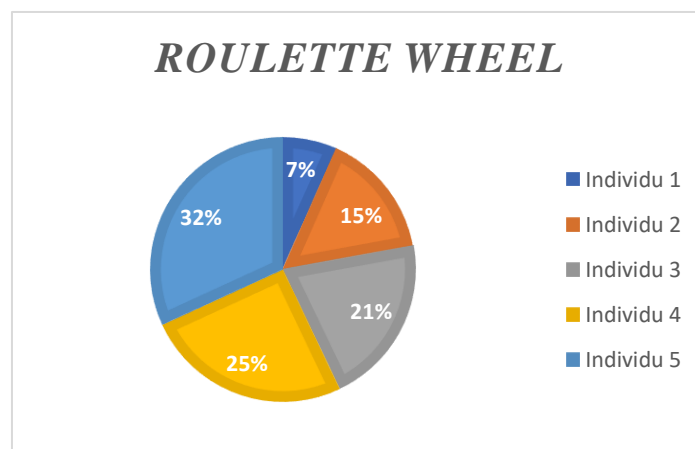
Pada langkah ini, hasil nilai probabilitas dari setiap individu akan dijumlahkan dengan nilai inisialisasi pada penjumlahan kumulatif sebelumnya.

Tabel 3.16 Hitung Nilai Komulatif

Kromosom	Kumulatif	Hasil Komulatif
1	0+0.208	0.208
2	0.208+0.278	0.486
3	0.486+0.167	0.653
4	0.653+0.139	0.792
5	0.792+0.208	1.000

Pada tabel 3.16 dilakukan perhitungan kumulatif dengan menjumlahkan nilai *probabilitas* dari individu sebelumnya. Misalnya, individu 1 memiliki nilai *probabilitas* 0.208, maka nilai kumulatifnya menjadi $0 + 0.208 = 0.208$. Kemudian, Individu 2 memiliki nilai probabilitas 0.288, sehingga nilai kumulatifnya menjadi $0.208 + 0.278 = 0,486$. Proses ini dilakukan secara berkelanjutan hingga mencapai individu terakhir, di mana nilai kumulatifnya mencapai 1.

Berdasarkan nilai kumulatif yang telah dihasilkan, dapat dilakukan pemetaan untuk setiap individu. Setelah melakukan pemetaan untuk individu, langkah selanjutnya adalah melakukan proses seleksi individu. Langkah pertama dalam proses seleksi adalah menghasilkan bilangan acak antara 0 hingga 1 sebanyak jumlah individu pada populasi. Diasumsikan, nilai acak yang dihasilkan adalah 0,49, 0,26, 0,50, 0,96 dan 0,79.



Gambar 3.7 *Probabilitas Kumulatif dalam Roulette wheel*

Pada gambar 3.7 dapat dilihat bahwa individu yang memiliki nilai 0,49 adalah individu 3, dengan interval nilai antara 0,48 hingga 0,65. Oleh karena itu, individu 3 akan menempati posisi individu 1. Selanjutnya, individu yang memiliki nilai 0,26 adalah individu 2 dengan *interval* nilai antara 0,28 hingga 0,48. Maka, individu 2 akan menempati posisi individu 2, proses ini akan terus berlangsung hingga semua bilangan acak yang dibangkitkan telah diletakan pada posisinya masing-masing. Berikut susunan populasi baru hasil dari proses seleksi:

Tabel 3.17 Populasi Baru Hasil Seleksi

Hasil <i>Roulette wheel</i>		
NO	KETERANGAN	HASIL
1	individu [1]	individu [3]
2	individu [2]	individu [2]
3	individu [3]	individu [3]
4	individu [4]	individu [5]
5	individu [5]	individu [5]

Hasil seleksi populasi dengan menggunakan metode *Roulette Wheel* dapat dilihat pada tabel 3.17 ini menampilkan hasil seleksi populasi menggunakan metode *Roulette wheel*. Setiap individu dalam populasi awal dinilai berdasarkan nilai *Fitness*, dan beberapa individu dipilih untuk membentuk populasi baru berdasarkan probabilitas seleksi.

Tabel 3.18 Populasi Baru dengan Induk

Hasil Seleksi					
Kromosom	idv 1	idv 2	idv 3	idv 4	idv 5
k1	[1, 5, 7]	[1, 5, 5]	[1, 5, 7]	[1, 3, 7]	[1, 3, 7]
k2	[2, 3, 6]	[2, 4, 10]	[2, 3, 6]	[2, 1, 4]	[2, 1, 4]
k3	[3, 4, 4]	[3, 4, 3]	[3, 4, 4]	[3, 4, 3]	[3, 4, 3]
k4	[4, 2, 3]	[4, 3, 6]	[4, 2, 3]	[4, 2, 2]	[4, 2, 2]
k5	[5, 4, 7]	[5, 5, 2]	[5, 4, 7]	[5, 3, 1]	[5, 3, 1]
k6	[6, 2, 5]	[6, 3, 2]	[6, 2, 5]	[6, 2, 10]	[6, 2, 10]
k7	[7, 3, 9]	[7, 3, 1]	[7, 3, 9]	[7, 3, 3]	[7, 3, 3]
k8	[8, 1, 4]	[8, 3, 5]	[8, 1, 4]	[8, 4, 1]	[8, 4, 1]
k9	[9, 4, 10]	[9, 2, 6]	[9, 4, 10]	[9, 5, 2]	[9, 5, 2]
k10	[10, 1, 1]	[10, 4, 8]	[10, 1, 1]	[10, 3, 4]	[10, 3, 4]

Berdasarkan hasil pada tabel 3.18, individu 1 dan 4 tidak terpilih dalam proses seleksi karena tidak masuk ke dalam bilangan acak yang dihasilkan oleh metode *roulette wheel*. Populasi baru hasil seleksi ini kemudian akan digunakan sebagai input pada tahap *Crossover* untuk menghasilkan solusi yang lebih optimal.

c. Proses *Crossover*

Langkah awal dalam proses *Crossover* adalah menentukan tingkat nilai probabilitas *Crossover*. Dalam kasus ini, nilai *Crossover rate* ialah 0,75. Setelah nilai nilai *Crossover rate* ditentukan langkah selanjutnya ialah menentukan bilangan acak sesuai dengan jumlah individu yang ada. Berikut bilangan acak yang telah di bangkitkan secara acak 0,08, 0,27, 0,76, 0,55, dan 0,62.

Proses *Crossover* hanya dilakukan jika nilai bilangan acak yang dihasilkan individu lebih kecil daripada nilai *Crossover rate* yang telah ditetapkan. Berdasarkan penyesuaian antara nilai *Crossover rate* dan bilangan acak, didapatkan bahwa kromosom yang akan mengalami proses *Crossover* adalah kromosom 1,2,4,5.

Setelah mendapatkan individu yang akan melakukan proses *Crossover* langkah selanjutnya ialah menentukan titik potong pada setiap proses *Crossover* secara acak. Bilangan acak yang digunakan untuk menentukan titik potong adalah 1 sampai dengan jumlah panjang kromosom dikurangi 1, berdasarkan aturan tersebut didapatkan *range* atau jarak yang akan digunakan ialah 1 sampai dengan 9.

Kemudian bilangan acak tersebut akan dibangkitkan sebanyak tingkat *Crossover* yang akan dilakukan. Dalam hal ini bilangan acak yang digunakan untuk pemotongan ialah 5,8,2,4. Berikut adalah proses *Crossover*:

Tabel 3.19 Proses *Crossover* 1

Proses <i>Crossover</i> 1			
Kromosom	idv 1	idv 2	<i>offspring</i> 1
k1	[1, 5, 7]	[1, 5, 5]	[1, 5, 7]
k2	[2, 3, 6]	[2, 4, 10]	[2, 3, 6]
k3	[3, 4, 4]	[3, 4, 3]	[3, 4, 4]
k4	[4, 2, 3]	[4, 3, 6]	[4, 2, 3]
k5	[5, 4, 7]	[5, 5, 2]	[5, 4, 7]
k6	[6, 2, 5]	[6, 3, 2]	[6, 3, 2]
k7	[7, 3, 9]	[6, 3, 2]	[6, 3, 2]
k8	[8, 1, 4]	[8, 3, 5]	[8, 3, 5]

k9	[9, 4, 10]	[9, 2, 6]	[9, 2, 6]
k10	[10, 1, 1]	[10, 4, 8]	[10, 4, 8]

Tabel 3.20 Proses *Crossover 2*

Proses <i>Crossover 2</i>			
Kromosom	idv 2	idv 4	offspring 2
k1	[1, 5, 5]	[1, 3, 7]	[1, 5, 5]
k2	[2, 4, 10]	[2, 1, 4]	[2, 4, 10]
k3	[3, 4, 3]	[3, 4, 3]	[3, 4, 3]
k4	[4, 3, 6]	[4, 2, 2]	[4, 3, 6]
k5	[5, 5, 2]	[5, 3, 1]	[5, 5, 2]
k6	[6, 3, 2]	[6, 2, 10]	[6, 3, 2]
k7	[7, 3, 1]	[7, 3, 3]	[7, 3, 1]
k8	[8, 3, 5]	[8, 4, 1]	[8, 3, 5]
k9	[9, 2, 6]	[9, 5, 2]	[9, 5, 2]
k10	[10, 4, 8]	[10, 3, 4]	[10, 3, 4]

Tabel 3.21 Proses *Crossover 3*

Proses <i>Crossover 3</i>			
Kromosom	idv 4	idv 5	offspring 3
k1	[1, 3, 7]	[1, 3, 7]	[1, 3, 7]
k2	[2, 1, 4]	[2, 1, 4]	[2, 1, 4]
k3	[3, 4, 3]	[3, 4, 3]	[3, 4, 3]
k4	[4, 2, 2]	[4, 2, 2]	[4, 2, 2]
k5	[5, 3, 1]	[5, 3, 1]	[5, 3, 1]
k6	[6, 2, 10]	[6, 2, 10]	[6, 2, 10]
k7	[7, 3, 3]	[7, 3, 3]	[7, 3, 3]
k8	[8, 4, 1]	[8, 4, 1]	[8, 4, 1]
k9	[9, 5, 2]	[9, 5, 2]	[9, 5, 2]
k10	[10, 3, 4]	[10, 3, 4]	[10, 3, 4]

Tabel 3.22 Proses *Crossover 4*

Proses <i>Crossover 4</i>			
Kromosom	idv 5	idv 1	offspring 4
k1	[1, 3, 7]	[1, 5, 7]	[1, 3, 7]
k2	[2, 1, 4]	[2, 3, 6]	[2, 1, 4]
k3	[3, 4, 3]	[3, 4, 4]	[3, 4, 3]
k4	[4, 2, 2]	[4, 2, 3]	[4, 2, 2]

k5	[5, 4, 7]	[5, 4, 7]	[5, 4, 7]
k6	[5, 3, 1]	[6, 2, 5]	[6, 2, 5]
k7	[7, 3, 3]	[7, 3, 9]	[7, 3, 9]
k8	[8, 4, 1]	[8, 1, 4]	[8, 1, 4]
k9	[9, 5, 2]	[9, 4, 10]	[9, 4, 10]
k10	[10, 3, 4]	[10, 1, 1]	[10, 1, 1]

Dapat dilihat pada tabel 3.19, proses *Crossover* 1 dilakukan dengan mengawinkan dua individu induk menggunakan titik potong yang telah ditentukan untuk menghasilkan individu baru. Pada tabel tersebut, individu 1 yang terdiri dari urutan [1, 5, 7], [2, 3, 6], [3, 4, 4], [4, 2, 3], dan [5, 4, 7] (ditandai dengan warna hijau) dijadikan induk 1, sementara individu 2 yang terdiri dari urutan [6, 3, 2], [6, 3, 2], [8, 3, 5], [9, 2, 6], dan [10, 4, 8] (ditandai dengan warna *orange*) dijadikan induk 2. Proses *Crossover* dilakukan dengan menukar kromosom pada titik potong yang telah ditentukan, menghasilkan individu anak yang terdiri dari gabungan kromosom dari kedua induk, seperti yang ditunjukkan dalam hasil kawin silang: [1, 5, 7], [2, 3, 6], [3, 4, 4], [4, 2, 3], [5, 4, 7], [6, 3, 2], [6, 3, 2], [8, 3, 5], [9, 2, 6], dan [10, 4, 8]. Proses *Crossover* ini berlanjut hingga semua individu induk menjalani kawin silang dan menghasilkan individu anak. Setelah itu, nilai fitness dari individu anak dihitung untuk mengevaluasi apakah solusi yang dihasilkan dari proses *Crossover* memberikan dampak positif pada populasi yang ada dan memenuhi kriteria penjadwalan. Berikut perhitungan nilai *Fitness* individu anak:

$$1 \quad F \text{ individu anak 1} = \frac{1}{1 + (0 + 3)} = 0.250$$

$$2 \quad F \text{ individu anak 2} = \frac{1}{1 + (0 + 2)} = 0.333$$

$$3 \quad F \text{ individu anak 3} = \frac{1}{1 + (2 + 2)} = 0.200$$

$$4 \quad F \text{ individu anak 4} = \frac{1}{1 + (0 + 3)} = 0.250$$

$$\text{Total Nilai fitness} = 1.033$$

Berdasarkan perhitungan nilai *Fitness*, dihasilkan nilai *Fitness* tertinggi adalah 0,333, dari hasil ini populasi yang dihasilkan melalui melalui proses *Crossover* belum dapat dikatakan optimal. Oleh Karena itu, diperlukan langkah selanjutnya yaitu proses mutasi.

d. Mutasi

Mutasi bertujuan untuk mengubah gen gen yang ada dalam kromosom, dengan potensi menghasilkan peningkatan atau penurunan nilai *Fitness* pada individu. Pada proses mutasi, beberapa gen dipilih secara acak untuk diubah, terutama pada gen gen kelompok ruangan dan waktu. Tingkat *probabilitas* mutasi yang digunakan dalam penelitian ini diasumsikan sebesar 20%. Untuk menentukan posisi gen yang akan dimutasi, jumlah total gen dalam populasi dihitung dengan mengalikan jumlah kromosom dalam setiap individu dalam populasi. Dalam kasus ini, terdapat 5 individu yang terdiri dari 10 kromosom, sehingga total kromosom dalam populasi adalah 50. Setelah mengetahui total kromosom dalam populasi, langkah selanjutnya adalah mengalikan jumlah total kromosom dengan tingkat *probabilitas* mutasi. Dalam hal ini, tingkat mutasi adalah 20% atau 0,20. Sehingga, hasil perkalian tersebut adalah $0,20 \times 50 = 10$. Langkah berikutnya adalah memilih bilangan acak sebanyak 10 kali dari rentang 1 hingga 50. Contoh bilangan acak yang didapatkan adalah 2, 45, 38, 5, 46, 16, 10, 32, 11, dan 26. Proses penukaran gen dapat dilihat pada tabel 3.23 dan 3.24.

Tabel 3.23 Sebelum Mutasi

Sebelum Mutasi					
Kromosom	anak 1	anak 2	induk 3	anak 4	anak 5
k1	[1, 5, 7]	[1, 5, 5]	[1, 5, 7]	[1, 3, 7]	[1, 3, 7]
k2	[2, 3, 6]	[2, 3, 6]	[2, 4, 10]	[2, 1, 4]	[2, 1, 4]
k3	[3, 4, 4]	[3, 4, 4]	[3, 4, 3]	[3, 4, 3]	[3, 4, 3]
k4	[4, 2, 3]	[4, 2, 3]	[4, 3, 6]	[4, 2, 2]	[4, 2, 2]
k5	[5, 4, 7]	[5, 4, 7]	[5, 5, 2]	[5, 3, 1]	[5, 4, 7]
k6	[6, 2, 5]	[6, 3, 2]	[6, 3, 2]	[6, 2, 10]	[6, 2, 5]
k7	[7, 3, 9]	[7, 3, 1]	[7, 3, 1]	[7, 3, 3]	[7, 3, 9]

k8	[8, 1, 4]	[8, 3, 5]	[8, 3, 5]	[8, 4, 1]	[8, 4, 1]
k9	[9, 4, 10]	[9, 2, 6]	[9, 5, 2]	[9, 5, 2]	[9, 4, 10]
k10	[10, 1, 1]	[10, 4, 8]	[10, 3, 4]	[10, 3, 4]	[10, 1, 1]

Pada tabel 3.23 menunjukkan populasi sebelum proses mutasi. Gen-gen yang akan mengalami mutasi telah ditandai dengan warna oranye berdasarkan bilangan acak yang dihasilkan. Adapun bilangan acak yang digunakan untuk proses mutasi pada gen adalah [3,9], [2,3], [3,9], [5,10], [3,7], [1,2], [2,2], [3,2], [2,6], dan [5,4].

Tabel 3.24 Sesudah Mutasi

Sesudah Mutasi					
Kromosom	anak 1	anak 2	induk 3	anak 4	anak 5
k1	[1, 5, 7]	[2, 3, 6]	[1, 5, 7]	[1, 3, 7]	[1, 3, 7]
k2	[2, 3, 9]	[2, 3, 6]	[2, 1, 2]	[2, 1, 4]	[2, 1, 4]
k3	[3, 4, 4]	[4, 2, 3]	[3, 4, 3]	[3, 4, 3]	[3, 4, 3]
k4	[4, 2, 3]	[4, 2, 3]	[4, 3, 6]	[4, 2, 2]	[4, 2, 2]
k5	[5, 2, 3]	[5, 4, 7]	[5, 5, 2]	[5, 3, 1]	[5, 2, 6]
k6	[6, 2, 5]	[6, 3, 7]	[6, 2, 2]	[6, 2, 10]	[6, 5, 4]
k7	[7, 3, 9]	[7, 3, 1]	[7, 3, 1]	[7, 3, 3]	[7, 3, 9]
k8	[8, 1, 4]	[8, 3, 5]	[8, 3, 5]	[8, 3, 2]	[8, 4, 1]
k9	[9, 4, 10]	[9, 2, 6]	[9, 5, 2]	[9, 5, 2]	[9, 4, 10]
k10	[10, 3, 9]	[10, 4, 8]	[10, 3, 4]	[10, 3, 4]	[10, 1, 1]

Berdasarkan tabel 3.24, terlihat bahwa setiap individu masih memiliki konflik. Pada anak 1 terdapat 5 konflik, anak 2 memiliki 2 konflik kapasitas; induk 3 memiliki 5 konflik kapasitas, anak 4 memiliki 3 konflik kapasitas, dan anak 5 memiliki 5 konflik kapasitas. Langkah selanjutnya adalah menghitung nilai *Fitness* untuk semua individu dalam populasi. Proses perhitungan nilai *Fitness* dapat dijelaskan melalui persamaan berikut:

$$1 \quad F \text{ individu 1} = \frac{1}{1 + (2 + 4)} = 0.143$$

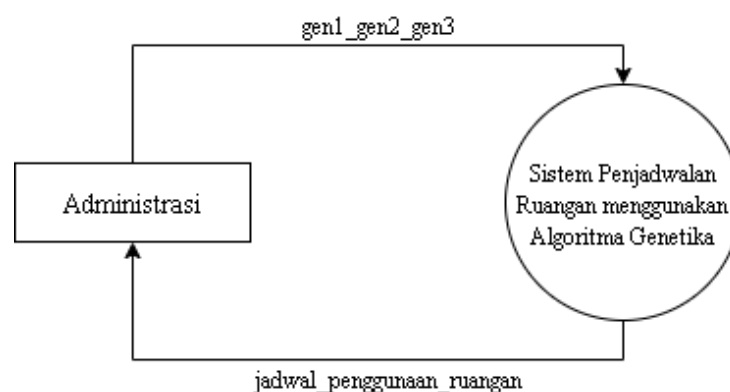
2	$F \text{ individu 2} = \frac{1}{1 + (0 + 2)} =$	0.333
3	$F \text{ individu 3} = \frac{1}{1 + (1 + 3)} =$	0.200
4	$F \text{ individu 4} = \frac{1}{1 + (0 + 3)} =$	0.250
5	$F \text{ individu 5} = \frac{1}{1 + (2 + 3)} =$	0.167
	<i>Total Nilai fitness =</i>	1.093

Menurut hasil perhitungan nilai *Fitness* di atas, masih terjadi bentrokan dalam proses penjadwalan. Oleh karena itu, diperlukan seleksi ulang atau pengulangan proses yang sama hingga diperoleh nilai *Fitness* terbaik, yaitu 1, yang menunjukkan bahwa solusi optimal telah ditemukan tanpa adanya bentrokan.

3.4 Perancangan Sistem

3.4.1 Diagram Konteks

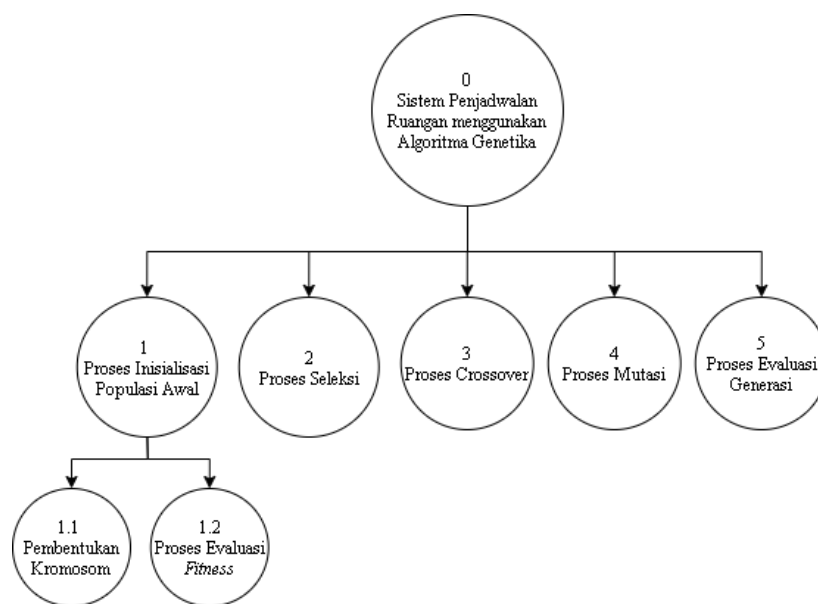
Diagram ini menunjukkan entitas di luar atau terlibat dalam sistem, termasuk aktivitas dari data yang menghubungkan entitas sistem.



Gambar 3.8 Diagram Konteks Sistem Penjadwalan Ruangan menggunakan Algoritma Genetika

Diagram konteks sistem penjadwalan ruangan dapat dilihat pada gambar 3.8. Diagram ini menggambarkan di mana Administrasi berperan sebagai entitas utama dalam pengelolaan sistem tersebut. Proses dimulai dengan Administrasi memasukkan input berupa data gen1, gen2, dan gen3. Melalui proses analisis dalam sistem, dihasilkan keluaran berupa Jadwal Penggunaan Ruangan.

3.4.2 Diagram Jenjang Sistem



Gambar 3.9 Diagram Berjenjang Sistem Penjadwalan Ruangan menggunakan Algoritma Genetika

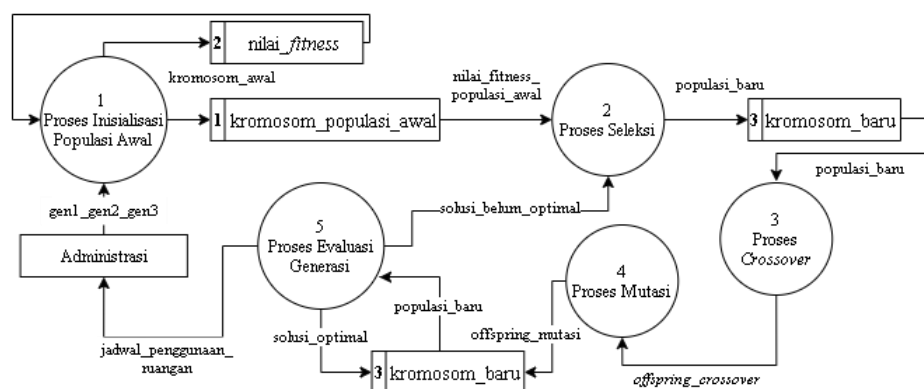
Pada Gambar 3.9 menggambarkan diagram berjenjang sistem yang terdiri dari:

1. Top level : Sistem Penjadwalan Ruangan menggunakan Algoritma Genetika.
2. Level 1 : Hasil dari pembagian proses aplikasi sistem menjadi beberapa sub proses yang meliputi Inisialisasi Populasi Awal, Proses Seleksi, Proses *Crossover*, Proses Mutasi dan Proses Evaluasi Generasi.

3. Level 2 : Bagian dari proses yang ada pada Level 1 Inisialisasi populasi mencakup langkah-langkah yaitu pembentukan kromosom dan menghitung nilai *Fitness*.

3.4.3 Data Flow Diagram (DFD) Sistem

3.4.3.1 DFD Level 1 Sistem Penjadwalan Ruangan menggunakan Algoritma Genetika



Gambar 3.10 DFD Level 1 Sistem Penjadwalan Ruangan di Gedung Dakwah Muhammadiyah Cabang Dukun

Aktivitas DFD level 1 pada Sistem Penjadwalan Ruangan menggunakan Algoritma Genetika terdapat beberapa proses antara lain yaitu :

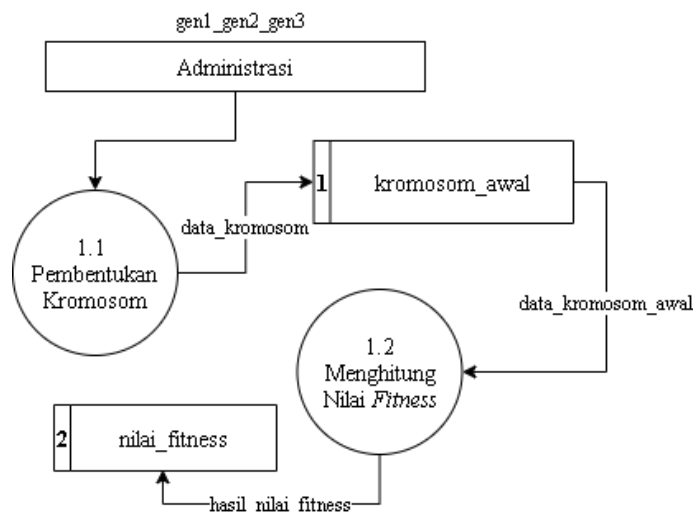
1. Pada Proses 1: Inisialisasi Populasi Awal, dilakukan pembentukan populasi awal berupa kumpulan kromosom yang merepresentasikan individu. Data input berupa *gen1_gen2_gen3* digunakan untuk membentuk kromosom awal dengan pendekatan randomisasi. Kromosom yang dihasilkan disimpan dalam *database* 1:kromosom_populasi_awal untuk digunakan dalam proses berikutnya.
2. Pada Proses 2: Seleksi, bertujuan untuk memilih individu dengan kualitas terbaik berdasarkan nilai *fitness*.

Kromosom yang tersimpan dalam Database 1 diambil dan dievaluasi menggunakan fungsi *fitness* yang dirancang untuk mengukur sejauh mana solusi memenuhi kriteria optimalitas. Seleksi dilakukan dengan menggunakan metode *roulette wheel selection*, di mana individu dengan nilai *fitness* lebih tinggi memiliki peluang lebih besar untuk dipilih. Namun, peluang seleksi juga diberikan kepada kromosom dengan *fitness* lebih rendah untuk menjaga keragaman populasi. Kromosom terpilih kemudian disimpan dalam Database 3: kromosom_baru untuk diolah pada tahap selanjutnya.

3. Pada Proses 3: *Crossover*, individu hasil seleksi dikombinasikan menggunakan teknik *One-point crossover*, di mana segmen genetik dari dua individu induk ditukar pada titik potong tertentu. Tujuan utama dari proses ini adalah untuk menghasilkan solusi baru dan berpotensi lebih optimal, yang menghasilkan keturunan yang disebut sebagai *offspring_crossover*, yang selanjutnya disimpan dalam *database*: 3 kromosom_baru.
4. Pada Proses 4: mutasi, bertujuan untuk meningkatkan keragaman genetik dalam populasi. Individu hasil *crossover* dimodifikasi dengan mengubah nilai gen tertentu secara acak untuk menciptakan solusi baru yang belum ada dalam populasi sebelumnya. Proses ini penting untuk mencegah algoritma terjebak pada solusi sub optimal (*local optimum*). Untuk menjaga keseimbangan antara solusi baru dan solusi yang ada. Hasil mutasi, yang disebut *offspring_mutasi*, disimpan kembali ke *database* 3: kromosom_baru.
5. Pada Proses 5: Evaluasi Generasi, Kromosom yang telah mengalami proses mutasi dievaluasi kembali untuk

menentukan apakah solusi yang dihasilkan sudah mencapai solusi optimal. Evaluasi ini dilakukan dengan membandingkan nilai *fitness* setiap individu dengan batas optimal yang telah ditentukan. Jika ditemukan solusi optimal, data tersebut disimpan dalam *database* 3: kromosom_baru dan diteruskan ke proses Administrasi untuk digunakan sebagai jadwal final. Jika solusi belum optimal, populasi baru dikembalikan ke proses 3: seleksi untuk menjalani iterasi berikutnya hingga mencapai solusi terbaik.

3.4.3.2 DFD Level 2 Sistem Penjadwalan Ruangan menggunakan Algoritma Genetika



Gambar 3.11 DFD Level 2 Sistem Penjadwalan Ruangan menggunakan Algoritma Genetika

Aktivitas DFD level 2 pada Sistem Penjadwalan Ruangan menggunakan Algoritma Genetika terdapat beberapa proses antara lain yaitu :

1. Proses 1.1: Pembentukan Kromosom

Pada tahap ini, data Gen 1, Gen 2, dan Gen 3 digunakan untuk membentuk kromosom awal yang merepresentasikan

solusi penjadwalan. Kromosom ini kemudian disimpan dalam *database* 1: kromosom_awal.

2. Proses 1.2: Menghitung Nilai *Fitness*

Kromosom yang terbentuk selanjutnya dihitung nilai *fitness*-nya untuk menilai kualitas solusi berdasarkan kriteria penjadwalan. Hasil perhitungan nilai *fitness* disimpan dalam *database* 2: nilai_ *fitness* .

3.5 Perancangan Basis Data

Basis data (*Database*) adalah kumpulan dari data yang berhubungan antara satu dengan yang lainnya, tersimpan di perangkat keras komputer dan menggunakan perangkat lunak untuk memanipulasinya. Adapun perancangan database mengenai sistem informasi ini diperlukan beberapa tabel sebagai berikut:

3.5.1 Tabel User

Tabel user menyimpan data pengguna dan memungkinkan pengguna mengakses aplikasi.

Tabel 3.25 Tabel User

#	Name	Type	Length	Key
1	id	int	15	PK
2	nama	varchar	155	-
3	username	varchar	155	-
4	password	password	155	-
5	created_at	timestamp	-	-
6	update_at	timestamp	-	-

3.5.2 Tabel Pengguna (GEN 1)

Tabel pengguna adalah data Identitas yang digunakan untuk melakukan proses penggunaan ruangan.

Tabel 3.26 Tabel Lembaga

#	Name	Type	Length	Key
1	id	int	15	PK
2	kode_lembaga	varchar	155	FK
3	ktam	int	155	-
4	nama_lembaga	varchar	155	-

5	email	Varchar	155	-
6	no_telp	Int	155	-
7	alamat	text	155	-
8	created_at	timestamp	-	-
9	update_at	timestamp	-	-

Tabel 3.27 Tabel Kegiatan

#	Name	Type	Length	Key
1	id	int	15	PK
2	id_kegiatan	varchar	155	FK
3	kegiatan	varchar	155	-
4	jumlah_peserta	int	155	-
5	created_at	timestamp	-	-
6	update_at	timestamp	-	-

3.5.3 Tabel Ruangan (GEN 2)

Tabel ruangan menyimpan data ruangan yang akan digunakan dalam proses penggunaan.

Tabel 3.28 Tabel Ruangan

#	Name	Type	Length	Key
1	id	int	15	PK
2	id_ruangan	varchar	155	FK
3	nama_ruangan	varchar	155	-
4	kapasitas	password	155	-
5	created_at	timestamp	-	-
6	update_at	timestamp	-	-

3.5.4 Tabel Waktu (GEN 3)

Tabel waktu adalah data untuk menentukan jadwal penggunaan ruangan.

Tabel 3.29 Tabel Hari

#	Name	Type	Length	Key
1	id	int	15	PK
2	id_hari	varchar	155	FK
3	nama_hari	varchar	155	-
5	created_at	timestamp	-	-
6	update_at	timestamp	-	-

Tabel 3.30 Tabel Jam

#	Name	Type	Length	Key
1	id	int	15	PK
2	id_jam	varchar	155	FK
3	jam	date	155	-
5	created_at	timestamp	-	-
6	update_at	timestamp	-	-

3.5.5 Tabel Kromosom Populasi Awal

Tabel kromosom populasi awal merupakan tabel yang menampung dari setiap tabel yang dibutuhkan untuk melakukan proses penjadwalan dengan menggunakan algoritma genetika.

Tabel 3.31 Tabel kromosom_populasi_awal

#	Name	Type	Length	Key
1	id	bigint	155	PK
2	lembaga_id	bigint	155	-
3	kegiatan_id	bigint	155	-
4	Jumlah_peserta_id	bigint	155	-
5	ruangan_id	bigint	155	-
6	kapasitas_id	bigint	155	-
7	hari_id	int	155	-
8	sesi_id	int	155	-
9	fitness	bigint	-	FK
10	created_at	timestamp	-	-
11	update_at	timestamp	-	-

3.5.6 Tabel Kromosom Baru

Tabel kromosom baru merupakan tabel yang menyimpan hasil perubahan dari setiap proses, yang dibutuhkan untuk menemukan penjadwalan optimal menggunakan algoritma genetika.

Tabel 3.32 Tabel kromosom_baru

#	Name	Type	Length	Key
1	id	bigint	155	PK
2	lembaga_id	bigint	155	-
3	kegiatan_id	bigint	155	-
4	Jumlah_peserta_id	bigint	155	-
5	ruangan_id	bigint	155	-
6	kapasitas_id	bigint	155	-

7	hari_id	int	155	-
8	sesi_id	int	155	-
9	fitness	bigint	155	FK
10	offspring_crossover	Int	155	-
11	offspring_mutasi	Int	155	-
12	generasi	int	155	-
13	created_at	timestamp	-	-
14	update_at	timestamp	-	-

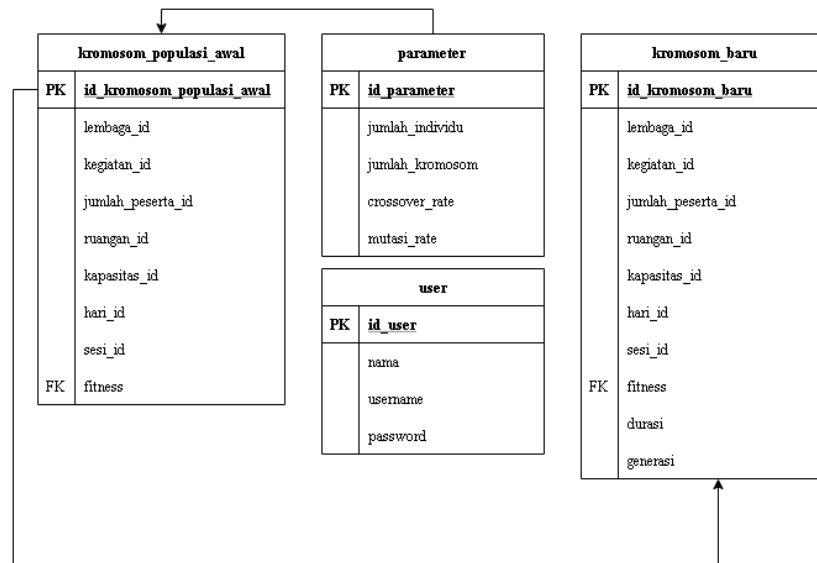
3.5.7 Tabel Parameter

Tabel parameter adalah tabel yang berisi input proses perhitungan Algoritma Genetika.

Tabel 3.33 Tabel Parameter

#	Name	Type	Length	Key
1	id	int	155	PK
2	Jumlah_gen	int	155	-
3	Jumlah_populasi	int	155	-
4	Crossover_rate	int	155	-
5	Mutasi_rate	int	155	-
6	created_at	timestamp	-	-
7	update_at	timestamp	-	-

Sistem ini memiliki tabel *user* yang terhubung dengan tabel kromosom_populasi_awal, parameter, dan kromosom_baru. Relasi antara user dan kedua tabel kromosom adalah *one-to-many*, yang berarti satu pengguna dapat memiliki banyak kromosom. Selain itu, relasi antara parameter dengan kedua tabel kromosom juga *one-to-many*, yang menunjukkan bahwa satu set parameter dapat mempengaruhi banyak kromosom, seperti yang digambarkan pada gambar 3.12.



Gambar 3.12 ERD Sistem Penjadwalan Ruangan menggunakan Algoritma Genetika

3.6 Perancangan Antarmuka Sistem

3.6.1 Halaman Login

Halaman login digunakan oleh administrasi yang ingin menjadwalkan ruangan. Administrasi harus memasukkan *username* dan *password* yang benar. Jika informasi yang dimasukkan tidak sesuai dengan data yang tersimpan, sistem akan mengeluarkan peringatan yang menunjukkan bahwa *username* dan *password* yang diberikan salah.

logo

LOGIN

Gambar 3.13 Halaman Login

3.6.2 Halaman Dashboard

Halaman dashboard berfungsi sebagai antarmuka utama yang disajikan kepada administrasi setelah berhasil masuk ke sistem. Halaman ini mencakup informasi terkait data ruangan, aktivitas, dan detail relevan lainnya. Selain itu, halaman dashboard memberikan informasi jadwal penggunaan ruangan saat ini.

GDM ADMIN	DASHBOARD nama user profile			
	DASHBOARD ADMIN Data Pengguna GEN 1 Data Ruangan GEN 2 Data Waktu GEN 3 Penjadwalan Generate Jadwal Lihat Jadwal	TOTAL USER	TOTAL RUANGAN	TOTAL KEGIATAN
PENGATURAN	TOTAL LEMBAGA			
	JADWAL PENGGUNAAN RUANGAN			

Gambar 3.14 Halaman Dashboard

3.6.3 Halaman Input Pengguna

Suatu halaman akan dibuat untuk penyertaan data pengguna, yang akan digunakan dalam proses penjadwalan.

<

Gambar 3.15 Halaman Input Pengguna

3.6.4 Halaman Proses Algoritma Genetika

Halaman Proses Algoritma Genetika merupakan formulir yang digunakan untuk memproses data yang telah diinputkan sebelumnya dengan

menggunakan parameter-parameter algoritma genetika, yaitu jumlah gen, banyaknya kromosom, serta *probabilitas Crossover* dan Mutasi. Rancangan *form* parameter algoritma genetika untuk pemrosesan jadwal dapat dilihat pada gambar 3.16.

GDM ADMIN DASHBOARD ADMIN Data Pengguna GEN 1 Data Ruangan GEN 2 Data Waktu GEN 3 Penjadwalan Generate Jadwal Lihat Jadwal PENGATURAN	JADWAL NAMA USER			
	JUMLAH GENERASI	JUMLAH INDIVIDU	CROSSOVER RATE	MUTASI RATE
	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
	<input type="button" value="GENERATE JADWAL"/>			
	PROSES 1 POPULASI AWAL PROSES 2 SELEKSI PROSES 3 CROSSOVER PROSES 4 MUTASI PROSES 5 EVALUASI GENERASI			

Gambar 3.16 Halaman Proses Algoritma Genetika Admin

Form ini digunakan untuk menampilkan hasil akhir dari proses penjadwalan, yaitu jadwal penggunaan yang telah dihasilkan. Untuk melihat hasil akhir tersebut, pengguna dapat mengklik tombol "Generate Jadwal" dan hasilnya akan ditampilkan. Form ini dapat dilihat pada Gambar 3.17.

GDM ADMIN DASHBOARD ADMIN Data Pengguna GEN 1 Data Ruangan GEN 2 Data Waktu GEN 3 Penjadwalan Generate Jadwal Lihat Jadwal PENGATURAN	JADWAL							NAMA USER
	NO	NAMA LEMBAGA	NAMA KEGIATAN	JUMLAH PESERTA	RUANGAN	KAPASITAS	HARI	JAM
	1	MTs Maarif Tanwirul Qulub	Pusat Dakwah	90	Lantai 1 - Ruang 3	150	SENIN	13:00 - 17:00
	2	MTs Mathlabul Huda	Pendidikan	250	Lantai 1 - Ruang 3	150	SENIN	08:00 - 12:00
	3	MTs Muhammadiyah 10	Sosial	190	Lantai 1 - Ruang 4	250	SELASA	13:00 - 17:00
	4	MTs Ihyaul Ulum	Rapat dan Koordinasi	220	Lantai 1 - Ruang 5	400	SELASA	08:00 - 12:00
	5	MTs Tarbiyatul Athfal	Umum	130	Lantai 1 - Ruang 3	400	RABU	08:00 - 12:00
	6	MTs Nur Rahmat	Seminar	30	Lantai 1 - Ruang 2	150	RABU	13:00 - 17:00

Gambar 3.17 Halaman Hasil Penjadwalan

3.7 Spesifikasi Pengembangan Sistem

1. Kebutuhan Perangkat Keras

Perangkat keras adalah alat yang digunakan untuk menunjang dalam pembuatan sistem. Dalam pembuatan sistem ini perangkat keras yang digunakan yaitu laptop dengan spesifikasi:

- a. *Processor Intel Core I5*
- b. *RAM 16 GB*
- c. *SSD 512 GB*
- d. *HDD 1024 GB*
- e. *Monitor 14"*
- f. *Mouse*
- g. *Keyboard*

2. Kebutuhan Perangkat Lunak

Perangkat lunak adalah program atau aplikasi yang digunakan untuk membangun sistem. Perangkat lunak yang dibutuhkan dalam pembuatan sistem ini adalah:

- a. *Windows 11*
- b. *Visual Studio Code*
- c. *Web server Apache*
- d. *Bahasa PHP Framework Laravel versi 11*
- e. *Bootstrap*
- f. *Database server MySQL*

3.8 Skenario Pengujian Sistem

Adapun perencanaan pengujian dilakukan sebagai berikut:

1. Data yang digunakan:
 - a. Pengguna yang terdiri atas:
 - Lembaga
 - Kegiatan
 - Jumlah Peserta
 - b. Ruangan terdiri atas ruangan dan kapasitas ruangan
 - c. Waktu terdiri atas hari dan sesi
2. Melakukan pengujian terhadap

- 5 Data
- 10 Data
- 15 Data

Dengan melakukan 5 kali generate tiap pengujian

Tabel 3.34 Tabel Skenario Pengujian

Jumlah Data	Pengujian	Jumlah Generasi	Crossover Rate	Mutas Rate	Waktu
5	1				
	2				
	3				
	4				
	5				
10	1				
	2				
	3				
	4				
	5				
15	1				
	2				
	3				
	4				
	5				

3. Hasil tiap pengujian ditampilkan seperti pada tabel 3.35.

Tabel 3.35 Hasil Pengujian

Lembaga	Kegiatan	Jumlah Peserta	Ruangan	Kapasitas	Hari	Sesi

BAB 4

IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1 Implementasi Sistem

4.1.1 Implementasi Metode Algoritma Genetika

Pada proses implementasi sistem, terdapat batasan dalam jumlah iterasi yang digunakan, yaitu maksimal 1000 iterasi. Batasan ini ditentukan berdasarkan spesifikasi perangkat yang telah dijelaskan pada Subbab 3.7 Spesifikasi Pengembangan Sistem. Saat iterasi melebihi 1000, perangkat yang digunakan mengalami error selama proses generate. Hal ini disebabkan oleh keterbatasan penyimpanan dan durasi waktu yang dibutuhkan. Oleh karena itu, peneliti membatasi jumlah iterasi hingga 1000 untuk memastikan proses pencarian jadwal optimal dapat berjalan dengan lancar tanpa mengganggu kinerja sistem.

Proses perhitungan menggunakan metode Algoritma Genetika terdapat beberapa tahap dalam menghitung data dalam sistem.

Tabel 4.1 Source Code Pembentukan Populasi Awal

```
// Proses Generate Populasi Awal
for ($i = 1; $i <= $jumlahIndividu; $i++) {
    $individu = []; // Menyimpan data individu per
generasi

    foreach ($penjadwalan as $jadwal) { // Loop over each
'penjadwalan'
        // Randomly select ruangan and waktu
        $gen2 = $ruangan->random()-
>only(['kode_ruangan', 'kapasitas']);
        $gen3 = $waktu->random();

        $kodeRuangan = $gen2['kode_ruangan'] ?? null;
        $kapasitasRuangan = $gen2['kapasitas'] ?? null;

        if (!$kodeRuangan || !$kapasitasRuangan ||
!$gen3) {
            continue; // Lewati jika data tidak valid
        }

        $isBentrokKapasitas = $jadwal->jumlah_peserta >
$kapasitasRuangan;

        $bentrokRuangan = [];
        foreach ($individu as $existing) {
            if (
```

```

$kodeRuangan && $existing['gen2']['ruangan'] ==
$gen3->kode_waktu $existing['gen3']['kode_waktu'] ==
) {
    $bentrokRuangan[] =
    $existing['gen1']['kode_pengguna'];
}

$individu[] = [
    'gen1' => [
        'kode_pengguna' => $jadwal->
        >kode_pengguna,
        'lembaga' => $jadwal->lembaga->
        >nama_lembaga ?? 'N/A',
        'kegiatan' => $jadwal->kegiatan->
        >nama_kegiatan ?? 'N/A',
        'jumlah_peserta' => [
            'value' => $jadwal->jumlah_peserta,
            'is_bentrok' => $isBentrokKapasitas,
        ],
    ],
    'gen2' => [
        'ruangan' => $kodeRuangan,
        'kapasitas' => [
            'value' => $kapasitasRuangan,
            'is_bentrok' => $isBentrokKapasitas,
        ],
    ],
    'gen3' => [
        'kode_waktu' => $gen3->kode_waktu,
        'hari' => $gen3->kode_hari,
        'sesi' => $gen3->kode_sesi,
        'is_bentrok' => !empty($bentrokRuangan),
        // Pastikan kunci ini selalu ada
        'bentrok_ruangan' => $bentrokRuangan,
    ],
    'is_bentrok' => $isBentrokKapasitas ||
    !empty($bentrokRuangan),
];

// Add the individual to the population
$generasiData[] = $individu;
}

```

Proses generate populasi awal pada Tabel 4.1 membentuk kumpulan individu, di mana setiap individu merepresentasikan kombinasi jadwal yang terdiri dari informasi pengguna (kode pengguna, lembaga, kegiatan, dan jumlah peserta), ruangan (kode dan kapasitas), serta waktu (hari dan sesi) yang dipilih secara acak. Setiap individu diuji untuk mendeteksi potensi

konflik, seperti jumlah peserta yang melebihi kapasitas ruangan (konflik kapasitas) atau penggunaan ruangan yang sama pada waktu yang sama (konflik ruangan). Populasi awal yang dihasilkan ini menjadi dasar untuk tahapan optimasi lebih lanjut dalam algoritma genetika.

Tabel 4.2 Source Code Mencari Nilai Fitness

```

    foreach ($individu as $innerKey => $innerKromosom) {
        if (
            $key !== $innerKey && // Jangan
membandingkan dengan dirinya sendiri
            $kromosom['gen2']['ruangan'] ===
$innerKromosom['gen2']['ruangan'] && // Ruangan sama
            $kromosom['gen3']['hari'] ===
$innerKromosom['gen3']['hari'] && // Hari sama
            $kromosom['gen3']['sesi'] ===
$innerKromosom['gen3']['sesi'] // Sesi sama
        ) {
            $totalKR++;
            break; // Hanya hitung satu kali per
konflik
        }
    }

    // Hitung nilai fitness individu dengan rumus: 1
/ (1 + KR + KK)
    $fitnessValue = 1 / (1 + $totalKR + $totalKK);
    $fitnessResults[] = [
        'individu' => $individuIndex + 1,
        'fitness' => $fitnessValue,
        'kr' => $totalKR,
        'kk' => $totalKK
    ];
    $totalFitness += $fitnessValue;

```

Pada Tabel 4.2, kode ini menghitung nilai **fitness** setiap individu berdasarkan konflik ruangan (KR) dan konflik kapasitas (KK). Konflik ruangan dihitung dengan memeriksa jadwal yang menggunakan ruangan, hari, dan sesi yang sama dalam satu individu. Nilai fitness dihitung menggunakan rumus: $\text{fitness} = 1 / (1 + \text{jumlah konflik ruangan} + \text{jumlah konflik kapasitas})$, di mana semakin sedikit konflik, semakin tinggi nilai fitness. Hasil perhitungan mencakup indeks individu, nilai fitness, jumlah konflik ruangan, dan kapasitas, yang menjadi dasar untuk proses seleksi algoritma genetika.

Tabel 4.3 Source Code Proses Seleksi

```

// Langkah 1: Hitung Probabilitas
$probabilities = [];
foreach ($fitnessResults as $fitness) {
    $probabilities[] = $fitness['fitness'] /
$totalFitness;
}

// Langkah 2: Hitung Kumulatif
$cumulativeProbabilities = [];
$currentCumulative = 0;
foreach ($probabilities as $probability) {
    $currentCumulative += $probability;
    $cumulativeProbabilities[] = $currentCumulative;
}

// Langkah 3: Bangkitkan Bilangan Acak
$randomNumbers = [];
$numOfIndividuals = count($fitnessResults);
for ($i = 0; $i < $numOfIndividuals; $i++) {
    $randomNumbers[] = mt_rand(0, 100) / 100; //
Generate random number between 0 and 1
}

// Langkah 4: Menggantikan Individu Lama berdasarkan
Random Number
$newGeneration = [];
foreach ($randomNumbers as $random) {
    foreach ($cumulativeProbabilities as $index =>
$cumulative) {
        if ($random <= $cumulative) {
            $newGeneration[] =
$generasiData[$index];
            break;
        }
    }
}
}

```

Pada Tabel 4.3, kode ini menjelaskan proses seleksi individu dalam algoritma genetika menggunakan metode roulette wheel selection. Proses dimulai dengan menghitung probabilitas seleksi untuk setiap individu berdasarkan nilai fitness relatif terhadap total fitness populasi. Kemudian, dihitung probabilitas kumulatif untuk membentuk rentang seleksi individu. Selanjutnya, dihasilkan bilangan acak antara 0 dan 1 yang digunakan untuk memilih individu dari populasi berdasarkan probabilitas kumulatif tersebut.

Hasil akhirnya adalah generasi baru yang terdiri dari individu-individu terpilih untuk melanjutkan ke tahap optimasi berikutnya.

Tabel 4.4 Source Code Proses *Crossover*

```
// Step 1: Pilih individu berdasarkan crossover rate
for ($i = 0; $i < $numOfIndividuals; $i++) {
    $random = mt_rand(0, 100) / 100; // Generate
bilangan acak antara 0 dan 1
    $output['randomNumbers'][] = $random;

    if ($random <= $crossoverRate) {
        $output['selectedIndividuals'][] = $i; //
Simpan indeks individu yang terpilih
    } else {
        $output['originalIndividuals'][] =
$newGeneration[$i]; // Simpan individu yang tidak terpilih
    }
}

$selected = $output['selectedIndividuals'];
$numOfSelected = count($selected);

// Step 2: Bentuk pasangan dan lakukan crossover jika
lebih dari 1 individu terpilih
if ($numOfSelected > 1) {
    for ($i = 0; $i < $numOfSelected; $i++) {
        $parent1 = $selected[$i];
        $parent2 = $selected[(($i + 1) %
$numOfSelected)]; // Pasangan cyclic

        $crossoverPoint = mt_rand(1,
count($newGeneration[$parent1]) - 1);

        $offspring1 = array_merge(
            array_slice($newGeneration[$parent1],
0, $crossoverPoint),
            array_slice($newGeneration[$parent2],
$crossoverPoint)
        );

        $output['pairs'][] = [
            'parent1' => $parent1 + 1,
            'parent2' => $parent2 + 1,
            'crossoverPoint' => $crossoverPoint,
            'offspring' => [$offspring1],
        ];
        $output['offspring'][] = $offspring1;
    }
}
```

Pada Tabel 4.4, kode ini menjelaskan proses crossover dalam algoritma genetika, yang bertujuan menghasilkan individu baru (offspring) dengan

menggabungkan gen dari dua individu induk. Proses dimulai dengan memilih individu berdasarkan crossover rate, di mana bilangan acak digunakan untuk menentukan apakah individu terpilih atau tidak. Induk yang terpilih kemudian dipasangkan secara siklis, dan proses crossover dilakukan dengan menentukan crossover point secara acak. Gen sebelum crossover point diambil dari induk pertama, sedangkan sisanya diambil dari induk kedua. Hasilnya adalah individu baru (offspring) yang akan digunakan dalam generasi berikutnya untuk meningkatkan keberagaman populasi.

Tabel 4.5 Source Code Mutasi

```
// Identifikasi kromosom bentrok
foreach ($newGeneration as $individuIndex =>
$individu) {
    foreach ($individu as $kromosomIndex =>
$kromosom) {
        if ($kromosom['is_bentrok'] ?? false) {
            $bentrokKromosom[] = [
                'individu' => $individuIndex + 1,
                'kromosom' => $kromosomIndex + 1,
                'detail' => $kromosom,
            ];
        }
    }
}

$totalBentrok = count($bentrokKromosom);

// Pilih gen untuk dimutasi
$totalGenes = count($newGeneration) *
count($newGeneration[0]);
$numGenesToMutate = floor($totalGenes *
$mutationRate);
$randomGenes = [];

while (count($randomGenes) < $numGenesToMutate) {
    $randomIndex = mt_rand(1, $totalGenes);
    if (!in_array($randomIndex, $randomGenes)) {
        $randomGenes[] = $randomIndex;
    }
}
```

Pada Tabel 4.5, kode ini menjelaskan proses mutasi dalam algoritma genetika, yang bertujuan memperbaiki individu dengan konflik atau meningkatkan keberagaman populasi. Proses diawali dengan

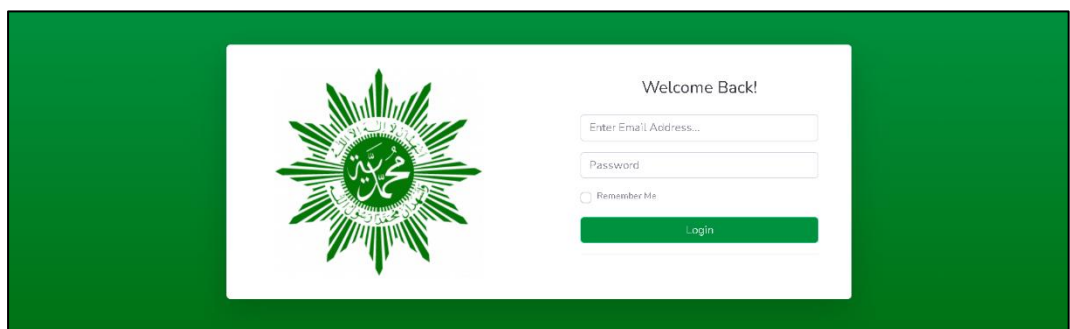
mengidentifikasi kromosom bentrok dalam setiap individu, yaitu kromosom yang menyebabkan konflik seperti ruangan atau waktu yang tumpang tindih. Selanjutnya, ditentukan jumlah gen yang akan dimutasi berdasarkan mutation rate, lalu dipilih secara acak gen-gen tersebut menggunakan bilangan acak. Proses mutasi ini mengganti gen yang terpilih dengan data baru, seperti ruangan atau waktu yang berbeda, untuk mengurangi konflik dalam populasi. Setelah ditemukan individu dengan nilai fitness sebesar 1, individu tersebut akan dianggap sebagai solusi optimal dan digunakan sebagai jadwal akhir yang kemudian disimpan ke dalam database.

4.2 Pengujian Sistem

Pengembangan sistem jadwal penggunaan ruangan di Gedung Dakwah Muhammadiyah cabang Dukun menggunakan software teks editor “Visual Studio Code” dan software web server “XAMPP”. Pengembangan sistem berbasis website dengan bahasa pemrograman PHP Framework Laravel.

4.2.1 Antarmuka Halaman Login

Antarmuka halaman login adalah tampilan awal Ketika sistem dijalankan. Pada halaman ini user harus memasukkan email dan password agar bisa masuk tampilan selanjutnya. Berikut tampilan antarmuka halaman login ditunjukkan pada gambar 4.1.



Gambar 4.1 Halaman Login

4.2.2 Halaman Dashboard

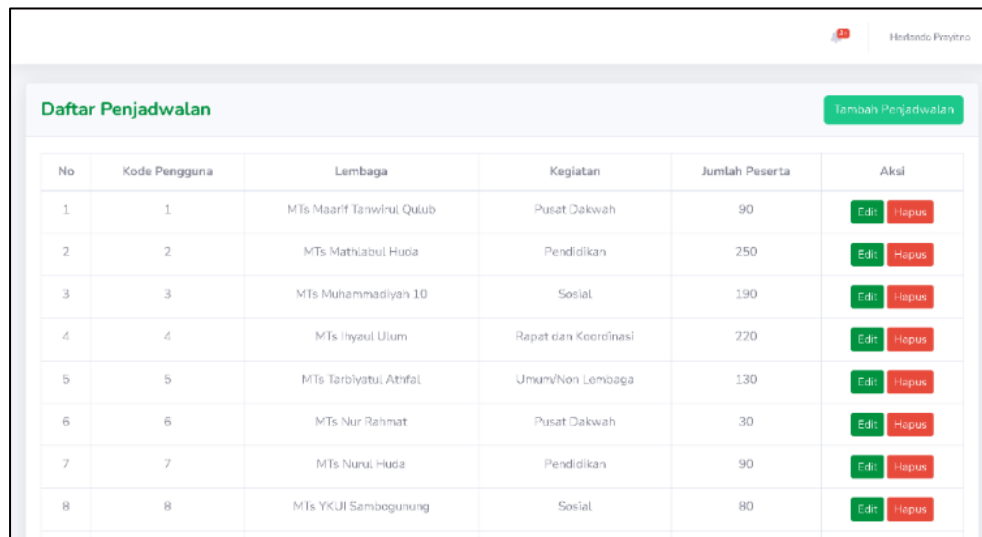
Antarmuka halaman awal atau dashboard adalah tampilan awal ketika aplikasi di jalankan. Pada halaman ini berisi informasi nama sistem dan ucapan selamat datang. Berikut tampilan antarmuka halaman dashboard ditunjukkan pada gambar 4.2:



Gambar 4.2 Halaman Dashboard

4.2.3 Halaman Data Pengguna

Antarmuka halaman data pengguna adalah tampilan pengguna yang akan dilibatkan dalam penjadwalan, halaman ini menjadi GEN 1 dalam proses algoritma genetika. Pada halama ini admin bisa menambah pengguna, edit dan delete data pengguna. Berikut tampilan antarmuka halaman data pengguna ditunjukkan pada gambar 4.3.

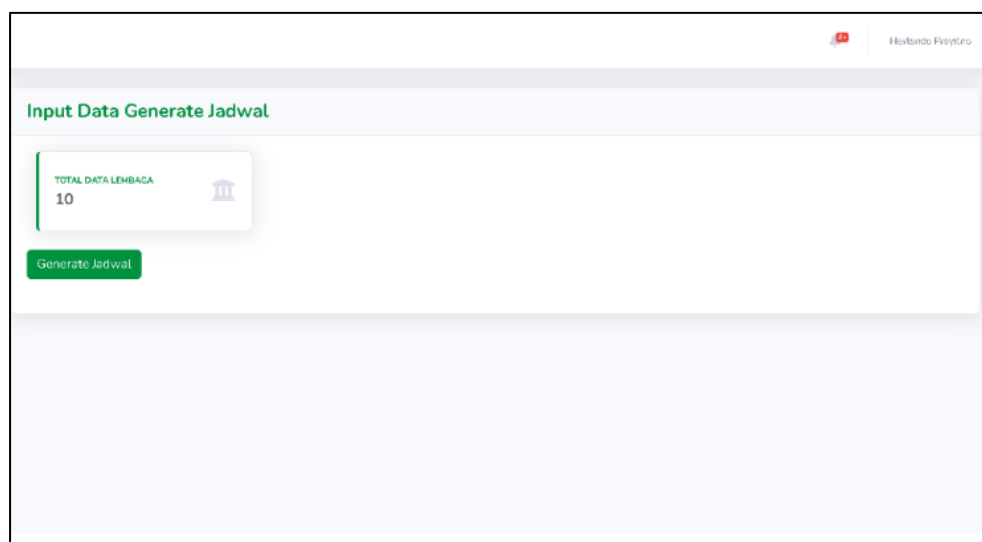


No	Kode Pengguna	Lembaga	Kegiatan	Jumlah Peserta	Aksi
1	1	MTs Maarif Tanwirul Qulub	Pusat Dakwah	90	Edit Hapus
2	2	MTs Mathiabul Huda	Pendidikan	250	Edit Hapus
3	3	MTs Muhammadiyah 10	Sosial	190	Edit Hapus
4	4	MTs Ihyaul Ulum	Rapat dan Koordinasi	220	Edit Hapus
5	5	MTs Tarbiyatul Atrifal	Umum/Non Lembaga	130	Edit Hapus
6	6	MTs Nur Rahmat	Pusat Dakwah	30	Edit Hapus
7	7	MTs Nurul Huda	Pendidikan	90	Edit Hapus
8	8	MTs YKUI Sambogunung	Sosial	80	Edit Hapus

Gambar 4.3 Halaman Pengguna

4.2.4 Halaman *Generate Jadwal Algoritma Genetika*

Antarmuka halaman *generate* jadwal Algoritma Genetika dapat dilihat pada gambar 4.4, tombol *generate* tersebut yang digunakan untuk memproses data dengan parameter yang telah dilakukan secara acak. Bagian pemrosesan jadwal dapat dilihat pada gambar 4.5.



Input Data Generate Jadwal

TOTAL DATA LEMBAGA
10

[Generate Jadwal](#)

Gambar 4.4 Halaman *Generate Jadwal*

Gambar 4.4 menunjukkan antarmuka Halaman *Generate Jadwal* untuk memulai proses optimasi jadwal menggunakan algoritma genetika. Pengguna

dapat memasukkan jumlah individu yang akan diproses, kemudian menekan tombol "Generate Jadwal" untuk memulai pengolahan data dengan parameter seperti crossover rate dan mutation rate yang dihasilkan secara acak. Hasil jadwal yang dioptimalkan akan ditampilkan pada halaman output seperti yang terlihat pada Gambar 4.5.

4.2.5 Halaman Jadwal

Antarmuka halaman jadwal adalah halaman yang menampilkan hasil jadwal yang telah dihasilkan melalui proses *generate* jadwal. Jadwal penggunaan ruangan dapat dilihat pada Gambar 4.5.

No	Nama Lembaga	Kegiatan	Jumlah Peserta	Ruangan	Kapasitas Ruangan	Hari	Sesi
1	MTs YKUI Sambogunung	Sosial	80	Lantai 1 - Ruang 3	150	SENIN	1
2	MTs Muhammadiyah 1	Umum/Non Lembaga	220	Lantai 1 - Ruang 4	250	SENIN	2
3	MTs Maarif Tanwirul Qulub	Pusat Dakwah	90	Lantai 1 - Ruang 4	250	SELASA	2
4	MTs YKUI Maskumambang	Rapat dan Koordinasi	40	Lantai 1 - Ruang 4	250	SELASA	1
5	MTs Muhammadiyah 10	Sosial	190	Lantai 1 - Ruang 5	400	RABU	1
6	MTs Nur Rahmat	Pusat Dakwah	30	Lantai 1 - Ruang 4	250	RABU	1
7	MTs Mathlabul Huda	Pendidikan	250	Lantai 1 - Ruang 5	400	KAMIS	2
8	MTs Ihyaul Ulum	Rapat dan Koordinasi	220	Lantai 1 - Ruang 4	250	JUMAT	1
9	MTs Tarbiyatul Athfal	Umum/Non Lembaga	130	Lantai 1 - Ruang 4	250	JUMAT	2
10	MTs Nurul Huda	Pendidikan	90	Lantai 1 - Ruang 5	400	JUMAT	2

Gambar 4.5 Halaman Jadwal

Pada gambar 4.5 menampilkan hasil jadwal penggunaan ruangan yang dihasilkan melalui proses *generate* jadwal. Tabel tersebut menyajikan informasi jadwal secara terstruktur, meliputi nomor urut, nama kegiatan, lembaga penyelenggara, jumlah peserta, ruangan, kapasitas ruangan, hari, dan sesi waktu. Jadwal ini merupakan hasil optimasi menggunakan algoritma genetika.

4.3 Hasil Analisis Sistem

Adapun pengujian sistem dilakukan sebagai berikut:

1. Data yang digunakan:
 - a. Pengguna yang terdiri atas:

- Lembaga
 - Kegiatan
 - Jumlah Peserta
- b. Ruangan terdiri atas ruangan dan kapasitas ruangan
- c. Waktu terdiri atas hari dan sesi
2. Melakukan pengujian terhadap
- 5 Data Kromosom
 - 10 Data Kromosom
 - 15 Data Kromosom

Dengan melakukan 5 kali generate tiap pengujian

Tabel 4.6 Skenario Pengujian

Jumlah Data Kromosom	Pengujian	Jumlah Generasi	Crossover Rate (%)	Mutasi Rate (%)	Waktu
5	1	15	95	25	0,0592
	2	1	89	69	0,0186
	3	3	39	76	0,0453
	4	1	84	100	0,0166
	5	74	72	6	0,0573
10	1	9	71	23	0,0726
	2	3	34	57	0,0698
	3	4	23	30	0,0338
	4	22	84	19	0,1626
	5	14	2	18	0,0599
15	1	52	96	51	1,8500
	2	234	83	8	0,8621
	3	22	90	10	0,1010
	4	55	89	4	0,1197
	5	18	72	6	0,0914

3. Hasil tiap pengujian ditampilkan seperti pada tabel 4.7.

Tabel 4.7 Hasil Pengujian

Lembaga	Kegiatan	Jumlah Peserta	Ruangan	Kapasitas	Hari	Sesi
MTs Maarif Tanwirul Qulub	Pusat Dakwah	90	Lantai 1 – Ruangan 1	150	Senin	2
MTs Mathlabul Huda	Pendidikan	250	Lantai 1 – Ruangan 4	250	Selasa	2

MTs Muhammadiyah 10	Sosial	190	Lantai 1 – Ruang 5	400	Selasa	1
MTs Ihyaul Ulum	Rapat dan Koordinasi	220	Lantai 1 – Ruang 5	400	Senin	1
MTs Tarbiyatul Athfal	Umum/Non Lembaga	130	Lantai 1 – Ruang 5	400	Kamis	2

Pada tabel 4.7 menampilkan hasil pengujian jadwal optimal untuk data yang digunakan dalam pengujian pertama dengan 5 data, 15 generasi, crossover rate 95%, mutation rate 25%, dan waktu eksekusi 0,0592 detik. Setiap jadwal mencakup informasi lembaga, kegiatan, jumlah peserta, ruangan yang digunakan, kapasitas ruangan, hari, dan sesi. Contoh hasil menunjukkan jadwal yang berhasil mengalokasikan ruangan dan waktu tanpa konflik.

- Lembaga: Mengindikasikan lembaga yang melakukan kegiatan.
- Kegiatan: Jenis kegiatan yang dijadwalkan.
- Jumlah Peserta: Total peserta yang hadir dalam kegiatan.
- Ruangan dan Kapasitas: Alokasi ruangan yang sesuai dengan jumlah peserta.
- Hari dan Sesi: Waktu pelaksanaan kegiatan.

Hasil pengujian menunjukkan masih adanya ketidak relevan antara jumlah peserta dan kapasitas ruangan, seperti pada kegiatan MTs Muhammadiyah 10 dengan 190 peserta di ruangan berkapasitas 400, menyisakan kelebihan 210 tempat. Hal serupa terjadi pada MTs Ihyaul Ulum (220 peserta) dan MTs Tarbiyatul Athfal (130 peserta) yang menggunakan ruangan berkapasitas 400, sehingga efisiensi pemanfaatan ruangan belum optimal. Hal tersebut disebabkan karena penelitian ini difokuskan pada optimasi penjadwalan untuk menghindari konflik jadwal, sehingga penyesuaian jumlah peserta dengan kapasitas ruangan tidak menjadi prioritas utama. Akibatnya, algoritma genetika lebih mengutamakan alokasi jadwal yang efisien tanpa mempertimbangkan secara mendalam kesesuaian

kapasitas ruangan dengan jumlah peserta. Namun, algoritma genetika tetap berhasil menghasilkan jadwal tanpa konflik, dengan rincian pengujian disajikan pada Lampiran 6 hingga Lampiran 20.

BAB 5

PENUTUP

5.1 Kesimpulan

Penelitian ini berhasil menentukan jadwal penggunaan ruangan yang optimal di Gedung Dakwah Muhammadiyah Cabang Dukun dengan menerapkan Algoritma Genetika. Metode ini terbukti efektif dalam menyelesaikan konflik penjadwalan serta mendukung pengelolaan ruangan secara terstruktur dan otomatis.

Berdasarkan hasil pengujian algoritma genetika dengan berbagai jumlah data (5, 10, dan 15) untuk mengevaluasi performa sistem dalam menghasilkan jadwal optimal. Setiap pengujian dilakukan sebanyak 5 kali dengan variasi parameter seperti crossover rate, mutation rate, jumlah generasi, dan waktu eksekusi. Berikut poin-poin utama dari analisis:

- a. Data Kecil (5 Data):
 - Waktu eksekusi rata-rata cepat ($<0,1$ detik).
 - Jumlah generasi yang dibutuhkan sedikit, menunjukkan efisiensi pada data kecil.
- b. Data Menengah (10 Data):
 - Waktu eksekusi meningkat, tetapi tetap di bawah 0,2 detik.
 - Variasi jumlah generasi mulai terlihat, menunjukkan peningkatan kompleksitas proses.
- c. Data Besar (15 Data):
 - Waktu eksekusi dan jumlah generasi cenderung meningkat signifikan.
 - Algoritma memerlukan lebih banyak iterasi untuk menemukan solusi optimal.

5.2 Saran

Penelitian ini tentunya tidak terlepas dari kekurangan dan kelemahan. Oleh karena itu, peneliti perlu memberikan saran untuk penelitian selanjutnya agar lebih baik. Saran yang ingin peneliti berikan adalah sebagai berikut:

1. Mengembangkan sistem berbasis desktop sebagai alternatif untuk membandingkan efisiensi proses generate jadwal menggunakan Algoritma Genetika antara platform website dan desktop.
2. Peneliti selanjutnya disarankan menambahkan batasan jumlah peserta guna meminimalkan ketidaksesuaian dalam penggunaan ruangan, sehingga hasil yang diperoleh lebih optimal.
3. Peneliti selanjutnya disarankan menggunakan laptop / PC dengan spesifikasi tinggi dan database yang lebih optimal selain *MySQL* untuk mempercepat proses pencarian waktu dan pengelolaan data dalam skala besar.
4. Masih terdapat hasil yang kurang optimal dalam penyesuaian antara kapasitas ruangan dan jumlah peserta. Sebagai contoh, peserta berjumlah 130 orang tetapi mendapatkan ruangan dengan kapasitas 400 orang, dan kasus serupa lainnya.

DAFTAR PUSTAKA

- Andriyadi, Anggi and Halimah. 2022. “Optimasi Algoritma Genetika Dalam Perancangan Sistem Informasi Penjadwalan Seminar Dan Sidang Skripsi Mahasiswa Institut Informatika Dan Bisnis (IIB) Darmajaya.” *Teknika* 16(1):133–1400.
- Ardiansyah, Hendri and Mochammad Bagoes Satria Junianto. 2022. “Penerapan Algoritma Genetika Untuk Penjadwalan Mata Pelajaran.” *Jurnal Media Informatika Budidarma* 6(1):329.
- Azzahra, Shafira, Fusvita Merdekawati, Acep Tantan Hardiana, Yuliansyah Sundara Mulia, and Ernawati Ernawati. 2019. “OPTIMASI VOLUME TEMPLAT DNA DAN SUHU DENATURASI UNTUK DETEKSI *Brugia Malayi* MENGGUNAKAN REAL-TIME PCR.” *Jurnal Riset Kesehatan Poltekkes Depkes Bandung* 11(2):198–203.
- Baker, Kenneth R. and Dan Trietsch. 1974. “Principles Of Sequencing And Scheduling.”
- Elva, Yesri. 2019. “Sistem Penjadwalan Mata Pelajaran Menggunakan Algoritma Genetika.” *Jurnal Teknologi Informasi* 3(1):49.
- Firmansyah, Boy, Dwi Sidik Permana, Natalia Evianti, Asep Mulyana Wihandar, and Ari Kurniawan. 2021. “Penerapan Algoritma Genetika Dalam Mengatasi Jadwal Mengajar Yang Bentrok Pada Program Studi Informatika Ibi Kosgoro 1957 Jakarta Indonesia.” *Jurnal Sistem Informasi Bisnis (JUNSIBI)* 2(2):106–25.
- Hartono, Rahmat and Afrizal Zein. 2023. “Penerapan Algoritma Genetika Dan Jaringan Syaraf Tiruan Dalam Penjadwalan Mata Kuliah.” 6 /*Jurnal Ilmu Komputer JIK* VI(03):6–7.
- Hidayat, Ilham, Sugeha Revo, L. Inkiriwang, and Pingkan A. K. Pratas. 2019. “Optimasi Penjadwalan Menggunakan Metode Algoritma Genetika Pada Proyek Rehabilitasi Puskesmas Minanga.” *Jurnal Sipil Statik* 7(12):1669–80.
- Informatics, Applied and Article Info. 2024. “Pada Aplikasi Agenda Rapat.” 7(3):440–48.
- Kristanti, Siska Dwi, Program Studi, Teknik Informatika, Universitas Bina Darma,

- and Kota Palembang. 2022. "4862-Article Text-14051-1-10-20221026." 14(2):22–31.
- Kurniati, Neng Ika, Alam Rahmatulloh, and Dewi Rahmawati. 2019. "Perbandingan Performa Algoritma Koloni Semut Dengan Algoritma Genetika – Tabu Search Dalam Penjadwalan Kuliah." *Computer Engineering, Science and System Journal* 4(1):17.
- Mone, Ferdinandus and Justin Eduardo Simarmata. 2021. "Aplikasi Algoritma Genetika Dalam Penjadwalan Mata Kuliah." *BAREKENG: Jurnal Ilmu Matematika Dan Terapan* 15(4):615–28.
- Muhandhis, Isnaini, M. Shubhan, Ib Dani, Arkila Rakasyah, Alven S. Ritonga, and Usniyah Sari. 2023. *Muhandhis, Pencarian Rute Terpendek Tim Promosi Kampus Dengan Menggunakan Algoritma Genetik 6 Pencarian Rute Terpendek Tim Promosi Kampus Dengan Menggunakan Algoritma Genetik.*
- Padaka, Emirensiana, Yulius Nahak Tetik, and Dian Fransiska Ledi. 2023. "Penerapan Algoritma Genetika Untuk Penjadwalan Mata Pelajaran Di SMK Negeri 1 Kota Tambolaka." *Jurnal Pendidikan, Sains Dan Teknologi* 2(4):966–74.
- Pangestu, Lintang Aji, Sayekti Harits Suryawan, and Asslia Johar Latipah. 2023. "Penerapan Algoritma Genetika Dalam Penjadwalan Mata Pelajaran." *Jurnal Informatika* 10(2):194–205.
- Rinaldi, Aldi and Ali Akbar Rismayadi. 2022. "Optimasi Penjadwalan Proyek Dengan Metode Algoritma Genetika." *EProsiding Teknik Informatika (PROTEKTIF)* 3(1):107–20.
- Rizki, Putri Afifah, Yeka Hendriyani, and Dony Novaliendry. 2023. "690.+Putri+Afifah+Rizki+24787-24797." *Jurnal Pendidikan Tambusai* 7:24787–98.
- Sari, Yuslena, Muhammad Alkaff, Eka Setya Wijaya, Syarifah Soraya, and Dany Primanita Kartikasari. 2019. "Optimasi Penjadwalan Mata Kuliah Menggunakan Metode Algoritma Genetika Dengan Teknik Tournament Selection." *Jurnal Teknologi Informasi Dan Ilmu Komputer* 6(1):85–92.
- Suzanti, Ika Oktavia and Fifin Ayu Mufarroha. 2021. "Implementasi Relevant

Feedback Menggunakan Algoritma Genetika Pada Dokumen Bahasa Indonesia (Implementation of Relevant Feedback Using Genetic Algorithm in Indonesian Documents).” *JURNAL IPTEKKOM Jurnal Ilmu Pengetahuan & Teknologi Informasi* 23(2):125–39.

Syawal, Muh, Poetri Lestari, and Abdul Rachman. 2021. “Implementasi Algoritma Genetika Untuk Penjadwalan Laboratorium Fakultas Ilmu Komputer Universitas Muslim Indonesia.” 2(1):29–37.

Tahir, Syifa Fauzia and Castaka Agus Sugianto. 2024. “Optimasi Naive Bayes Menggunakan Algoritma Genetika Pada Klasifikasi Komentar Cyberbullying Pada Media Sosial X.” *Jurnal Informatika Dan Teknik Elektro Terapan* 12(3):3350–56.

LAMPIRAN

Lampiran 1 Data Penjadwalan

TABEL PENGGUNA (GEN 1)			
NO	KEGIATAN	LEMBAGA	PESERTA
1	kg-01	MTsMTQ	90
2	kg-02	MTsMH	250
3	kg-03	MTsM10	190
4	kg-04	MTsIU	220
5	kg-05	MTsTA	130
...
15	kg-05	MTsAF	170

TABEL RUANGAN (GEN 2)		
NO	RUANGAN	KAPASITAS
1	r1	50
2	r2	100
3	r3	150
4	r4	250
5	r5	400

TABEL WAKTU (GEN 3)		
NO	HARI	JAM
1	SENIN	1
2	SENIN	2
3	SELASA	1
4	SELASA	2
5	RABU	1
6	RABU	2
7	KAMIS	1
8	KAMIS	2
9	JUM'AT	1
10	JUM'AT	2

Lampiran 2 Source Code Login

```

<div class="container d-block justify-content-center align-items-center">
  <!-- Outer Row -->
  <div class="row justify-content-center">
    <div class="col-xl-10 col-lg-12 col-md-9">
      <div class="card o-hidden border-0 shadow-lg my-5">
        <div class="card-body p-0">
          <!-- Nested Row within Card Body -->
          <div class="row">
            <!-- Image Section -->
            <div

```

```

class="col-lg-6 d-none d-lg-flex
bg-login-image justify-content-center align-items-center">
    
    </div>
    <!-- Form Section -->
    <div class="col-lg-6">
        <div class="p-5">
            <div class="text-center">
                <h1 class="h4 text-gray-
900 mb-4">Welcome Back!</h1>
            </div>
            <!-- Login Form -->
            <form method="POST"
action="{{ route('login.post') }}">
                @csrf
                <div class="form-group">
                    <input type="email"
class="form-control form-control-user" name="email"
placeholder="Ent
er Email Address..." required
value="{{
old('email') }}">
                @if ($errors-
>has('email'))
                    <span
class="text-danger">{{ $errors->first('email') }}</span>
                @endif
            </div>
            <div class="form-group">
                <input
type="password" class="form-control form-control-user"
name="password"
placeholder="Password" required>
                @if ($errors-
>has('password'))
                    <span
class="text-danger">{{ $errors->first('password') }}</span>
                @endif
            </div>
            <div class="form-group">
                <div class="custom-
control custom-checkbox small">
                    <input
type="checkbox" class="custom-control-input" id="customCheck"
name="rememb
er">
                    <label
class="custom-control-label" for="customCheck">Remember
Me</label>
                </div>
            </div>
            <button type="submit"
class="btn btn-success btn-user btn-block"

```



```

        <th>Lembaga</th>
        <th>Kegiatan</th>
        <th>Jumlah Peserta</th>
        <th>Aksi</th>
    </tr>
</thead>
<tbody>
    @foreach ($penjadwalan as $index => $item)
    <tr class="text-center">
        <td>{{ $index + 1 }}</td>
        <td>{{ $item->kode_pengguna }}</td>
        <td>{{ $item->lembaga->nama_lembaga
    <td>{{ $item->kegiatan->nama_kegiatan
    <td>{{ $item->jumlah_peserta }}</td>
        <td class="text-center">
            <a href="{{
route('penjadwalan.edit', $item->id) }}" class="btn btn-primary
btn-sm">Edit</a>
            <button class="btn btn-danger btn-
sm" data-toggle="modal" data-target="#deleteModal-{{ $item->id
}}">Hapus</button>

            <!-- Modal Konfirmasi Hapus -->
            <div class="modal fade"
id="deleteModal-{{ $item->id }}" tabindex="-1" role="dialog"
aria-labelledby="deleteModalLabel" aria-hidden="true">
                <div class="modal-dialog"
role="document">
                    <div class="modal-content">
                        <div class="modal-
header">
                            <h5 class="modal-
title" id="deleteModalLabel">Konfirmasi Hapus</h5>
                            <button
type="button" class="close" data-dismiss="modal" aria-
label="Close">
                                <span aria-
hidden="true">&times;</span>
                            </button>
                        </div>
                        <div class="modal-body">
                            Apakah Anda yakin
                            ingin menghapus jadwal ini?
                        </div>
                        <div class="modal-
footer">
                            <button
type="button" class="btn btn-secondary" data-
dismiss="modal">Batal</button>
                            <form action="{{
route('penjadwalan.destroy', $item->id) }}" method="POST">
                                @csrf
                                @method('DELETE'
)

```


[illegible]

Lampiran 5 Source Code Generate dan Jadwal

```
@extends('admin.layouts.base')

@section('title', 'Generate Jadwal')

@section('content')
    <div class="card shadow mb-4">
        <div class="card-header py-3">
            <h4 class="m-0 font-weight-bold text-primary">Input
Data Generate Jadwal</h4>
        </div>
        <div class="card-body">
            <form action="{{ route('process.generate.jadwal')
}}" method="POST">
                @csrf
                <div class="form-group">
                    <label for="jumlah_individu">Jumlah
Individu</label>
                    <input type="number" name="jumlah_individu"
id="jumlah_individu" class="form-control"
placeholder="Masukkan jumlah individu"
required>
                </div>
                <button type="submit" class="btn btn-
primary">Generate Jadwal</button>
            </form>

            @if (!empty($generasiData))
                <h4 class="text-primary text-center mt-
4">Populasi Awal</h4>
                @foreach ($generasiData as $index => $individu)
                    <div class="table-responsive mt-4">
                        <h5>Individu {{ $index + 1 }}</h5>
                        <table class="table table-bordered">
                            <thead>
                                <tr class="text-center">
                                    <th>Kro</th>
                                    <th>Kegiatan</th>
                                </tr>
                            </thead>
                        </table>
                    </div>
                </foreach>
            </if>
        </div>
    </div>
</section>
```

```

                <th>Lembaga</th>
                <th>Peserta</th>
                <th>Ruangan</th>
                <th>Kapasitas</th>
                <th>Hari</th>
                <th>Sesi</th>
            </tr>
        </thead>
        <tbody>
            @foreach ($individu as
$kromosom)
                <tr class="text-center">
                    <td>{{ $loop->iteration
}}</td>
                    <td>{{
$kromosom['gen1']['kegiatan'] }}</td>
                    <td>{{
$kromosom['gen1']['lembaga'] }}</td>
                    <td>{{
$kromosom['gen1']['jumlah_peserta']['value'] }}</td>
                    <td>{{
$kromosom['gen2']['ruangan'] }}</td>
                    <td>{{
$kromosom['gen2']['kapasitas']['value'] }}</td>
                    <td>{{
$kromosom['gen3']['hari'] }}</td>
                    <td>{{
$kromosom['gen3']['sesi'] }}</td>
                </tr>
            @endforeach
        </tbody>
    </table>
</div>
    @endforeach
    @else
        <p class="text-muted">Tidak ada data untuk
ditampilkan.</p>
    @endif

    @if (!empty($optimalIndividual))
        <h4 class="text-primary text-center mt-4">Hasil
Optimal</h4>
        <div class="table-responsive">
            <table class="table table-bordered">
                <thead>
                    <tr class="text-center">
                        <th>Kro</th>
                        <th>Kegiatan</th>
                        <th>Lembaga</th>
                        <th>Peserta</th>
                        <th>Ruangan</th>
                        <th>Kapasitas</th>
                        <th>Hari</th>
                        <th>Sesi</th>
                    </tr>
                </thead>

```

```

<tbody>
  @foreach ($optimalIndividual as $key
=> $kromosom)
    <tr class="text-center">
      <td>{{ $key + 1 }}</td>
      <td>{{
$kromosom['gen1']['kegiatan'] }}</td>
      <td>{{
$kromosom['gen1']['lembaga'] }}</td>
      <td>{{
$kromosom['gen1']['jumlah_peserta']['value'] }}</td>
      <td>{{
$kromosom['gen2']['ruangan'] }}</td>
      <td>{{
$kromosom['gen2']['kapasitas']['value'] }}</td>
      <td>{{
$kromosom['gen3']['hari'] }}</td>
      <td>{{
$kromosom['gen3']['sesi'] }}</td>
    </tr>
  @endforeach
</tbody>
</table>
</div>
@else
  <p class="text-muted text-center">Tidak ada
individu optimal ditemukan.</p>
@endif
</div>
</div>
@endsection

```

Lampiran 6 Data 5 Pengujian 1

Proses selesai! Total Generasi: 15 Waktu Proses: 0.0592 detik Crossover Rate: 98% Mutation Rate: 25%							
Kro	Nama Kegiatan	Lembaga	Jumlah Peserta	Ruangan	Kapasitas	Hari	Sesi
1	Pusat Dakwah	MTs Maarif Tanwirul Qulub	90	3	150	h1	s2
2	Pendidikan	MTs Mathlabul Huda	250	4	250	h2	s2
3	Sosial	MTs Muhammediyah 10	190	5	400	h2	s1
4	Rapat dan Koordinasi	MTs Ihyaul Ulum	220	5	400	h1	s1
5	Umum/Non Lembaga	MTs Tarbiyatul Athfal	130	5	400	h4	s2

Lampiran 7 Data 5 Pengujian 2

Proses selesai! Total Generasi: 1 Waktu Proses: 0.0186 detik Crossover Rate: 89% Mutation Rate: 69%							
Kro	Nama Kegiatan	Lembaga	Jumlah Peserta	Ruangan	Kapasitas	Hari	Sesi
1	Pusat Dakwah	MTs Maarif Tanwirul Qulub	90	5	400	h3	s1
2	Pendidikan	MTs Mathlabul Huda	250	4	250	h5	s1
3	Sosial	MTs Muhammediyah 10	190	4	250	h4	s1
4	Rapat dan Koordinasi	MTs Ihyaul Ulum	220	5	400	h3	s2
5	Umum/Non Lembaga	MTs Tarbiyatul Athfal	130	4	250	h1	s1

Lampiran 8 Data 5 Pengujian 3

Proses selesai! Total Generasi: 15 Waktu Proses: 0.0592 detik Crossover Rate: 95% Mutation Rate: 25%							
Kro	Nama Kegiatan	Lembaga	Jumlah Peserta	Ruangan	Kapasitas	Hari	Sesi
1	Pusat Dakwah	MTs Maarif Tanwinul Qulub	90	3	150	h1	s2
2	Pendidikan	MTs Mathlabul Huda	250	4	250	h2	s2
3	Sosial	MTs Muhammadiyah 10	190	5	400	h2	s1
4	Rapat dan Koordinasi	MTs Ihyaul Ulum	220	5	400	h1	s1
5	Umum/Non Lembaga	MTs Tarbiyatul Athfal	130	5	400	h4	s2

Lampiran 9 Data 5 Pengujian 4

Proses selesai! Total Generasi: 1 Waktu Proses: 0.0166 detik Crossover Rate: 84% Mutation Rate: 100%							
Kro	Nama Kegiatan	Lembaga	Jumlah Peserta	Ruangan	Kapasitas	Hari	Sesi
1	Pusat Dakwah	MTs Maarif Tanwinul Qulub	90	5	400	h4	s2
2	Pendidikan	MTs Mathlabul Huda	250	4	250	h5	s2
3	Sosial	MTs Muhammadiyah 10	190	5	400	h3	s1
4	Rapat dan Koordinasi	MTs Ihyaul Ulum	220	5	400	h2	s2
5	Umum/Non Lembaga	MTs Tarbiyatul Athfal	130	5	400	h1	s2

Lampiran 10 Data 5 Pengujian 5

Proses selesai! Total Generasi: 74 Waktu Proses: 0.0573 detik Crossover Rate: 72% Mutation Rate: 6%							
Kro	Nama Kegiatan	Lembaga	Jumlah Peserta	Ruangan	Kapasitas	Hari	Sesi
1	Pusat Dakwah	MTs Maarif Tanwinul Qulub	90	2	100	h2	s1
2	Pendidikan	MTs Mathlabul Huda	250	5	400	h3	s2
3	Sosial	MTs Muhammadiyah 10	190	5	400	h2	s1
4	Rapat dan Koordinasi	MTs Ihyaul Ulum	220	5	400	h2	s2
5	Umum/Non Lembaga	MTs Tarbiyatul Athfal	130	4	250	h5	s2

Lampiran 11 Data 10 Pengujian 1

Proses selesai! Total Generasi: 9 Waktu Proses: 0.0726 detik Crossover Rate: 71% Mutation Rate: 23%							
Kro	Nama Kegiatan	Lembaga	Jumlah Peserta	Ruangan	Kapasitas	Hari	Sesi
1	Pusat Dakwah	MTs Maarif Tanwinul Qulub	90	3	150	h1	s1
2	Pendidikan	MTs Mathlabul Huda	250	4	250	h2	s2
3	Sosial	MTs Muhammadiyah 10	190	5	400	h2	s2
4	Rapat dan Koordinasi	MTs Ihyaul Ulum	220	5	400	h4	s1
5	Umum/Non Lembaga	MTs Tarbiyatul Athfal	130	3	150	h5	s2
6	Pusat Dakwah	MTs Nur Rahmat	30	5	400	h3	s1
7	Pendidikan	MTs Nurd Huda	90	4	250	h2	s1
8	Sosial	MTs YKUR Sembagunung	80	4	250	h4	s2
9	Rapat dan Koordinasi	MTs YKUR Meksombong	40	1	50	h4	s2
10	Umum/Non Lembaga	MTs Muhammadiyah 1	220	4	250	h1	s1

Lampiran 12 Data 10 Pengujian 2

Proses selesai! Total Generasi: 3 Waktu Proses: 0.0698 detik Crossover Rate: 34% Mutation Rate: 57%							
Kro	Nama Kegiatan	Lembaga	Jumlah Peserta	Ruangan	Kapasitas	Hari	Sesi
1	Pusat Dakwah	MTs Maarif Tanwirul Qulub	90	5	400	h4	s2
2	Pendidikan	MTs Mathlabul Huda	250	5	400	h1	s1
3	Sosial	MTs Muhammadiyah 10	190	4	250	h1	s2
4	Rapat dan Koordinasi	MTs Ihyaul Ulum	220	5	400	h3	s2
5	Umum/Non Lembaga	MTs Tarbiyatul Athfal	130	4	250	h1	s1
6	Pusat Dakwah	MTs Nur Rahmat	30	3	150	h3	s1
7	Pendidikan	MTs Nurul Huda	90	4	250	h4	s2
8	Sosial	MTs YKUI Sambogunung	80	3	150	h2	s2
9	Rapat dan Koordinasi	MTs YKUI Maskumambang	40	3	150	h3	s2
10	Umum/Non Lembaga	MTs Muhammadiyah 1	220	5	400	h2	s2

Lampiran 13 Data 10 Pengujian 3

Proses selesai! Total Generasi: 4 Waktu Proses: 0.0338 detik Crossover Rate: 23% Mutation Rate: 30%							
Kro	Nama Kegiatan	Lembaga	Jumlah Peserta	Ruangan	Kapasitas	Hari	Sesi
1	Pusat Dakwah	MTs Maarif Tanwirul Qulub	90	4	250	h2	s2
2	Pendidikan	MTs Mathlabul Huda	250	4	250	h1	s2
3	Sosial	MTs Muhammadiyah 10	190	5	400	h3	s2
4	Rapat dan Koordinasi	MTs Ihyaul Ulum	220	4	250	h4	s1
5	Umum/Non Lembaga	MTs Tarbiyatul Athfal	130	4	250	h5	s1
6	Pusat Dakwah	MTs Nur Rahmat	30	1	50	h4	s1
7	Pendidikan	MTs Nurul Huda	90	5	400	h2	s2
8	Sosial	MTs YKUI Sambogunung	80	3	150	h4	s1
9	Rapat dan Koordinasi	MTs YKUI Maskumambang	40	2	100	h3	s1
10	Umum/Non Lembaga	MTs Muhammadiyah 1	220	5	400	h2	s1

Lampiran 14 Data Pengujian 4

Proses selesai! Total Generasi: 22 Waktu Proses: 0.1626 detik Crossover Rate: 84% Mutation Rate: 19%							
Kro	Nama Kegiatan	Lembaga	Jumlah Peserta	Ruangan	Kapasitas	Hari	Sesi
1	Pusat Dakwah	MTs Maarif Tanwirul Qulub	90	4	250	h3	s1
2	Pendidikan	MTs Mathlabul Huda	250	5	400	h5	s2
3	Sosial	MTs Muhammadiyah 10	190	5	400	h2	s1
4	Rapat dan Koordinasi	MTs Ihyaul Ulum	220	5	400	h1	s1
5	Umum/Non Lembaga	MTs Tarbiyatul Athfal	130	5	400	h4	s2
6	Pusat Dakwah	MTs Nur Rahmat	30	5	400	h3	s1
7	Pendidikan	MTs Nurul Huda	90	4	250	h4	s2
8	Sosial	MTs YKUI Sambogunung	80	2	100	h3	s1
9	Rapat dan Koordinasi	MTs YKUI Maskumambang	40	2	100	h1	s2
10	Umum/Non Lembaga	MTs Muhammadiyah 1	220	4	250	h2	s1

Lampiran 15 Data 10 Pengujian 5

Proses selesai! Total Generasi: 14 Waktu Proses: 0.0599 detik Crossover Rate: 2% Mutation Rate: 19%							
Kno	Nama Kegiatan	Lembaga	Jumlah Peserta	Ruangan	Kapasitas	Hari	Sesi
1	Pusat Dakwah	MTs Maarif Tanwirul Qulub	90	3	150	h1	s1
2	Pendidikan	MTs Mathlabul Huda	250	4	250	h3	s2
3	Sosial	MTs Muhammadiyah 10	190	5	400	h2	s1
4	Rapat dan Koordinasi	MTs Ihyaul Ulum	220	4	250	h1	s1
5	Umum/Non Lembaga	MTs Tarbiyatul Ahtfal	130	4	250	h1	s2
6	Pusat Dakwah	MTs Nur Rahmat	30	2	100	h4	s2
7	Pendidikan	MTs Nurul Huda	90	2	100	h3	s1
8	Sosial	MTs YKUI Sambogunung	80	4	250	h2	s1
9	Rapat dan Koordinasi	MTs YKUI Maskumambang	40	1	50	h1	s2
10	Umum/Non Lembaga	MTs Muhammadiyah 1	220	5	400	h4	s1

Lampiran 16 Data 15 Pengujian 1

Proses selesai! Total Generasi: 62 Waktu Proses: 1.8500 detik Crossover Rate: 96% Mutation Rate: 51%							
Kno	Nama Kegiatan	Lembaga	Jumlah Peserta	Ruangan	Kapasitas	Hari	Sesi
1	Pusat Dakwah	MTs Maarif Tanwirul Qulub	90	3	150	h5	s1
2	Pendidikan	MTs Mathlabul Huda	250	5	400	h1	s2
3	Sosial	MTs Muhammadiyah 10	190	4	250	h4	s2
4	Rapat dan Koordinasi	MTs Ihyaul Ulum	220	5	400	h2	s1
5	Umum/Non Lembaga	MTs Tarbiyatul Ahtfal	130	4	250	h3	s1
6	Pusat Dakwah	MTs Nur Rahmat	30	4	250	h1	s1
7	Pendidikan	MTs Nurul Huda	90	5	400	h2	s2
8	Sosial	MTs YKUI Sambogunung	80	3	150	h3	s2
9	Rapat dan Koordinasi	MTs YKUI Maskumambang	40	2	100	h5	s2
10	Umum/Non Lembaga	MTs Muhammadiyah 1	220	5	400	h4	s2
11	Pusat Dakwah	MTs Hidayatus Salam	180	4	250	h5	s2
12	Pendidikan	MTs Tanwirul Qulub YPPMU	190	5	400	h3	s2
13	Sosial	MTs YKUI Selargadung	80	2	100	h4	s2
14	Rapat dan Koordinasi	MTs Al Karimi 2	90	5	400	h3	s1
15	Umum/Non Lembaga	MTs Alfalalahiyah	170	4	250	h1	s2

Lampiran 17 Data 15 Pengujian 2

Proses selesai! Total Generasi: 234 Waktu Proses: 0.8622 detik Crossover Rate: 51% Mutation Rate: 5%							
Kno	Nama Kegiatan	Lembaga	Jumlah Peserta	Ruangan	Kapasitas	Hari	Sesi
1	Pusat Dakwah	MTs Maarif Tanwirul Qulub	90	5	400	h2	s1
2	Pendidikan	MTs Mathlabul Huda	250	4	250	h2	s2
3	Sosial	MTs Muhammadiyah 10	190	4	250	h3	s2
4	Rapat dan Koordinasi	MTs Ihyaul Ulum	220	4	250	h1	s1
5	Umum/Non Lembaga	MTs Tarbiyatul Ahtfal	130	5	400	h1	s1
6	Pusat Dakwah	MTs Nur Rahmat	30	3	150	h5	s1
7	Pendidikan	MTs Nurul Huda	90	3	150	h4	s2
8	Sosial	MTs YKUI Sambogunung	80	4	250	h3	s1
9	Rapat dan Koordinasi	MTs YKUI Maskumambang	40	1	50	h2	s1
10	Umum/Non Lembaga	MTs Muhammadiyah 1	220	4	250	h5	s2
11	Pusat Dakwah	MTs Hidayatus Salam	180	5	400	h3	s1
12	Pendidikan	MTs Tanwirul Qulub YPPMU	190	5	400	h5	s2
13	Sosial	MTs YKUI Selargadung	80	2	100	h2	s2
14	Rapat dan Koordinasi	MTs Al Karimi 2	90	2	100	h1	s1
15	Umum/Non Lembaga	MTs Alfalalahiyah	170	4	250	h2	s1

Lampiran 18 Data 15 Pengujian 3

Proses selesai! Total Generasi: 22 Waktu Proses: 0.1010 detik Crossover Rate: 80% Mutation Rate: 10%							
Kno	Nama Kegiatan	Lembaga	Jumlah Peserta	Ruangan	Kapasitas	Hari	Sex
1	Pusat Dakwah	MTs Maarif Tanwinul Qulub	90	5	400	h3	x2
2	Pendidikan	MTs Mathhabul Huda	250	5	400	h5	x2
3	Sosial	MTs Muhammadiyah 10	190	4	250	h4	x1
4	Rapat dan Koordinasi	MTs Ihsanul Uloom	220	5	400	h5	x1
5	Umum/Non Lembaga	MTs Tarbiyatul Ahdal	130	3	150	h2	x1
6	Pusat Dakwah	MTs Nur Rahmat	30	1	50	h1	x1
7	Pendidikan	MTs Nurul Huda	90	2	100	h2	x2
8	Sosial	MTs YKUI Sambogunung	80	3	150	h2	x2
9	Rapat dan Koordinasi	MTs YKUI Maskumambang	40	4	250	h4	x2
10	Umum/Non Lembaga	MTs Muhammadiyah 1	220	4	250	h5	x2
11	Pusat Dakwah	MTs Hidayatus Salam	160	5	400	h4	x1
12	Pendidikan	MTs Tanwinul Qulub YPPMU	180	5	400	h1	x2
13	Sosial	MTs YKUI Selargadung	80	3	150	h4	x2
14	Rapat dan Koordinasi	MTs Al Karimi 2	90	3	150	h5	x1
15	Umum/Non Lembaga	MTs Aifalahiyah	170	4	250	h2	x1

Lampiran 19 Data 15 Pengujian 4

Proses selesai! Total Generasi: 55 Waktu Proses: 0.1117 detik Crossover Rate: 80% Mutation Rate: 4%							
Kno	Nama Kegiatan	Lembaga	Jumlah Peserta	Ruangan	Kapasitas	Hari	Sex
1	Pusat Dakwah	MTs Maarif Tanwinul Qulub	90	2	100	h2	x2
2	Pendidikan	MTs Mathhabul Huda	250	4	250	h3	x1
3	Sosial	MTs Muhammadiyah 10	190	5	400	h4	x2
4	Rapat dan Koordinasi	MTs Ihsanul Uloom	220	5	400	h3	x1
5	Umum/Non Lembaga	MTs Tarbiyatul Ahdal	130	3	150	h3	x2
6	Pusat Dakwah	MTs Nur Rahmat	30	2	100	h4	x2
7	Pendidikan	MTs Nurul Huda	90	2	100	h3	x1
8	Sosial	MTs YKUI Sambogunung	80	3	150	h3	x1
9	Rapat dan Koordinasi	MTs YKUI Maskumambang	40	5	400	h2	x2
10	Umum/Non Lembaga	MTs Muhammadiyah 1	220	4	250	h3	x1
11	Pusat Dakwah	MTs Hidayatus Salam	160	5	400	h3	x2
12	Pendidikan	MTs Tanwinul Qulub YPPMU	180	5	400	h5	x1
13	Sosial	MTs YKUI Selargadung	80	3	150	h4	x1
14	Rapat dan Koordinasi	MTs Al Karimi 2	90	3	150	h5	x2
15	Umum/Non Lembaga	MTs Aifalahiyah	170	5	400	h2	x1

Lampiran 20 Data 15 Pengujian 5

Proses selesai! Total Generasi: 18 Waktu Proses: 0.0914 detik Crossover Rate: 72% Mutation Rate: 6%							
Kno	Nama Kegiatan	Lembaga	Jumlah Peserta	Ruangan	Kapasitas	Hari	Sex
1	Pusat Dakwah	MTs Maarif Tanwinul Qulub	90	4	250	h2	x2
2	Pendidikan	MTs Mathhabul Huda	250	4	250	h3	x1
3	Sosial	MTs Muhammadiyah 10	190	5	400	h2	x1
4	Rapat dan Koordinasi	MTs Ihsanul Uloom	220	4	250	h5	x2
5	Umum/Non Lembaga	MTs Tarbiyatul Ahdal	130	3	150	h2	x1
6	Pusat Dakwah	MTs Nur Rahmat	30	2	100	h4	x2
7	Pendidikan	MTs Nurul Huda	90	4	250	h1	x1
8	Sosial	MTs YKUI Sambogunung	80	2	100	h4	x1
9	Rapat dan Koordinasi	MTs YKUI Maskumambang	40	5	400	h5	x2
10	Umum/Non Lembaga	MTs Muhammadiyah 1	220	5	400	h1	x2
11	Pusat Dakwah	MTs Hidayatus Salam	160	5	400	h3	x1
12	Pendidikan	MTs Tanwinul Qulub YPPMU	180	4	250	h2	x1
13	Sosial	MTs YKUI Selargadung	80	5	400	h5	x1
14	Rapat dan Koordinasi	MTs Al Karimi 2	90	4	250	h4	x1
15	Umum/Non Lembaga	MTs Aifalahiyah	170	5	400	h2	x2