

**ANALISIS PERANCANGAN PENJADWALAN
PETUGAS IBADAH GEREJA HKBP PARANGINAN
MENGUNAKAN HYBRID PSO-GENETIC
ALGORITHM DENGAN PENDEKATAN WATERFALL**

TUGAS AKHIR

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

Oleh:

ELSA ELISA YOHANA SIANTURI

122140135



**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI, PRODUKSI DAN INDUSTRI
INSTITUT TEKNOLOGI SUMATERA
LAMPUNG SELATAN**

2024

DAFTAR ISI

DAFTAR ISI.....	ii
DAFTAR TABEL.....	iv
DAFTAR GAMBAR.....	v
DAFTAR RUMUS	vi
DAFTAR LAMPIRAN.....	vii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang Masalah	1
1.2 Identifikasi Masalah.....	3
1.3 Rumusan Masalah.....	3
1.4 Tujuan Penelitian	3
1.5 Batasan Masalah	4
1.7 Korelasi RP – RQ dan RO	4
1.8 Manfaat Penelitian	5
1.9 Sistematika Penulisan	6
1.9.1 Bab I Pendahuluan.....	6
1.9.2 Bab II Tinjauan Pustaka.....	6
1.9.3 Bab III Metode Penelitian.....	6
1.9.5 Bab IV Hasil dan Pembahasan.....	7
1.9.6 Bab V Kesimpulan dan Saran.....	7
BAB II TINJAUAN PUSTAKA	8
2.1 Tinjauan Pustaka.....	8
2.2 Dasar Teori	11
2.2.1 Penjadwalan.....	12

2.2.2 Particle Swarm Optimization Algorithm	13
2.2.2 Genetic Algorithm.....	14
BAB III METODE PENELITIAN	17
3.1 Desain Penelitian	17
3.2 Penjabaran Langkah Penelitian.....	17
3.2.1 Identifikasi Masalah.....	17
3.2.2 Pengumpulan Data.....	19
3.2.3 Perancangan Algoritma.....	21
3.2.4 Simulasi dan Pengujian Algoritma	29
3.2.5 Metode Pengujian	31
DAFTAR PUSTAKA	32
LAMPIRAN.....	34

DAFTAR TABEL

Tabel 1.1	Korelasi RP – RQ dan RO	4
Tabel 2.1	Review Literatur.....	10
Tabel 3.2	Data Set Ketersediaan Pelayan Gereja.....	19
Tabel 3.2	Data Set Operator.....	20
Tabel 3.3	Tabel Data Keluhan Hasil Wawancara.....	21

DAFTAR GAMBAR

Gambar 3.1	Diagram Penelitian.....	17
Gambar 3.2	Diagram Rancangan Algoritma.....	24
Gambar 3.3	Inisialisasi Populasi.....	25
Gambar 3.4	<i>Generate Random Valid Schedules</i>	26
Gambar 3.5	<i>Initialize Personal Best & Global Best</i>	27
Gambar 3.6	<i>Enter Main Loop</i>	28
Gambar 3.7	<i>Return Best Schedule</i>	29
Gambar 3.8	Tampilan Output Kode Program.....	30

DAFTAR RUMUS

Rumus 2.1	Kecepatan Partikel.....	13
Rumus 2.2	Posisi Partikel.....	14
Rumus 2.3	Nilai <i>Fitness</i>	15

DAFTAR LAMPIRAN

Lampiran 1	Kode Program.....	34
------------	-------------------	----

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Di era digital saat ini, pengelolaan organisasi keagamaan menghadapi tantangan yang semakin kompleks dalam hal administrasi dan manajemen. Gereja sebagai institusi keagamaan tidak hanya berfokus pada pelayanan spiritual, tetapi juga perlu mengelola berbagai aspek operasional secara efisien untuk mendukung kegiatan peribadatan. Perubahan zaman telah mendorong manusia untuk selalu berkembang, tidak terkecuali teknologi. Seiring dengan berkembangnya teknologi, teknologi komputer menjadi suatu hal yang sangat lumrah dewasa ini. Komputer merupakan bukti dari perkembangan teknologi ini. Hampir setiap bidang kehidupan telah menggunakan komputer sebagai alat bantu kerja, mulai dari bidang pendidikan, pemerintahan, serta dunia bisnis dan maupun dalam organisasi. Penggunaan teknologi informasi telah menjadi kebutuhan mendasar dalam mendukung pengelolaan organisasi keagamaan modern, mulai dari sistem administrasi jemaat, pengelolaan keuangan, hingga penjadwalan kegiatan dan petugas ibadah. Diperlukannya suatu strategi untuk bisa terus maju dan berkembang[1].

Salah satu tantangan utama dalam pengelolaan gereja adalah penjadwalan petugas ibadah yang melibatkan banyak variabel dan konstrain. Di HKBP Paranginan, penjadwalan petugas ibadah masih dilakukan secara manual, yang sering kali mengakibatkan beberapa permasalahan seperti:

- Distribusi tugas yang tidak merata antar petugas
- Konflik jadwal antara petugas yang memiliki *multiple roles*
- Kesulitan dalam melakukan perubahan jadwal secara cepat ketika ada petugas yang berhalangan
- Waktu yang cukup lama dalam menyusun jadwal karena harus mempertimbangkan berbagai kendala
- Kesulitan dalam memenuhi preferensi waktu dari masing-masing petugas ibadah

Optimasi penjadwalan menjadi krusial mengingat setiap minggu ada beberapa jenis pelayanan yang membutuhkan petugas berbeda, seperti *worship leader*, pembaca warta, pemusik, pembawa doa dan petugas *multimedia*. Setiap petugas memiliki kemampuan dan ketersediaan waktu yang berbeda-beda. Selain itu, jadwal ibadah yang dilakukan di HKBP Paranginan sebanyak 2 sesi ibadah untuk setiap hari Minggunya menambah konstrain yang harus terpenuhi dalam pembuatan jadwal petugas ibadah. Kompleksitas ini membutuhkan pendekatan komputasional yang dapat menghasilkan jadwal optimal dengan mempertimbangkan berbagai konstrain yang ada.

Permasalahan penjadwalan di HKBP Paranginan semakin kompleks dengan adanya berbagai aturan liturgi dan tradisi gereja yang harus dipatuhi. Selain itu, rotasi petugas harus mempertimbangkan aspek keadilan dan pengembangan talenta jemaat, di mana setiap anggota jemaat yang memiliki kemampuan harus diberi kesempatan untuk melayani.

Penggunaan algoritma *hybrid PSO-Genetic Algorithm* dipilih karena kemampuannya dalam menangani masalah optimasi kompleks dengan *multiple constraints*. *Particle Swarm Optimization (PSO)* memiliki kemampuan eksplorasi yang baik dalam mencari solusi global[2], sementara *Genetic Algorithm (GA)* unggul dalam eksploitasi solusi lokal[3]. Kombinasi kedua algoritma ini diharapkan dapat menghasilkan solusi penjadwalan yang lebih optimal dibandingkan penggunaan algoritma tunggal.

Metode *Waterfall* dipilih dalam perancangan sistem ini karena pendekatan bertahapnya yang terstruktur, dimulai dari analisis kebutuhan hingga perancangan sistem yang rinci. Dalam konteks perancangan penjadwalan petugas ibadah di HKBP Paranginan, setiap tahap perancangan dapat didefinisikan dengan jelas untuk memastikan bahwa kebutuhan organisasi dapat diterjemahkan ke dalam desain sistem yang sesuai. Selain itu, metode ini memungkinkan dokumentasi yang lengkap di setiap fase, sehingga memudahkan evaluasi dan perbaikan di masa mendatang[4]. Dengan metode *Waterfall*, kompleksitas permasalahan penjadwalan yang melibatkan banyak konstrain dapat dianalisis dan diuraikan secara sistematis, menghasilkan perancangan solusi yang terorganisir dan sesuai dengan kebutuhan liturgi serta tradisi gereja.

1.2 Identifikasi Masalah

Berdasarkan latar belakang yang telah diuraikan, berikut adalah identifikasi masalah yang dihadapi oleh HKBP Paranginan terkait pengelolaan penjadwalan petugas ibadah:

1. Keterbatasan Sistem Manual dalam Memenuhi Konstrain Penjadwalan
Penjadwalan petugas ibadah di HKBP Paranginan dilakukan secara manual, yang sering kali tidak mampu memenuhi seluruh konstrain, seperti aturan liturgi, rotasi yang adil, dan distribusi tugas yang merata.
2. Kompleksitas dalam Mengelola Konflik Jadwal
Petugas yang memiliki *multiple roles* sering menghadapi konflik jadwal.
3. Kesulitan Memenuhi Preferensi dan Kebutuhan Dinamis Gereja
Proses manual tidak fleksibel dalam mengakomodasi preferensi waktu setiap petugas maupun kebutuhan yang dinamis, seperti perubahan jadwal atau penyesuaian terhadap anggota baru.
4. Kinerja Sistem Manual yang Tidak Efisien
Waktu yang dibutuhkan untuk menyusun jadwal secara manual cukup lama, sementara hasilnya sering tidak optimal dalam hal efisiensi distribusi tugas dan keadilan antar petugas.

1.3 Rumusan Masalah

1. Bagaimana merancang penjadwalan petugas ibadah HKBP Paranginan menggunakan *hybrid PSO-Genetic Algorithm* yang dapat memenuhi seluruh konstrain dan kebutuhan gereja?
2. Bagaimana perancangan pendekatan *Waterfall* dalam penjadwalan untuk mengakomodasi kebutuhan yang dinamis dari HKBP Paranginan?
3. Bagaimana tingkat efektivitas sistem penjadwalan yang dikembangkan dalam mengatasi permasalahan distribusi tugas, konflik jadwal, dan preferensi waktu petugas dibandingkan dengan sistem manual?

1.4 Tujuan Penelitian

1. Perancangan Algoritma Penjadwalan
 - a. Merancang algoritma *hybrid Particle Swarm Optimization* (PSO) dan *Genetic Algorithm* (GA) untuk sistem penjadwalan petugas ibadah di HKBP Paranginan.

- b. Memastikan algoritma yang dirancang dapat memenuhi *multiple constraint*
- c. Menghasilkan jadwal yang optimal secara teoritis berdasarkan desain algoritma yang telah dirancang.

2. Analisis Sistem Penjadwalan

Dalam tugas akhir ini, akan dilakukan analisis terhadap efektivitas algoritma hybrid PSO-GA yang dirancang. Analisis ini mencakup:

- a. Evaluasi performa algoritma menggunakan data simulasi berbasis data historis dari HKBP Paranginan.
- b. Perbandingan hasil penjadwalan yang dihasilkan oleh algoritma dengan metode manual, untuk mengukur tingkat efisiensi, keadilan distribusi tugas, dan kemampuan mengatasi konflik jadwal.
- c. Kajian terhadap fleksibilitas algoritma dalam menghadapi perubahan kebutuhan gereja.

1.5 Batasan Masalah

- 1. Penelitian difokuskan pada penjadwalan petugas untuk 2 sesi ibadah mingguan
- 2. Data petugas yang digunakan terbatas pada anggota jemaat HKBP Paranginan yang aktif
- 3. Optimasi jadwal hanya mempertimbangkan konstrain yang telah didefinisikan.

1.7 Korelasi RP – RQ dan RO

Tabel 1.1 Korelasi RP – RQ dan RO

RP	RQ	RO
Keterbatasan sistem manual dalam memenuhi konstrain penjadwalan	Bagaimana merancang penjadwalan petugas ibadah HKBP Paranginan menggunakan <i>hybrid</i> PSO-GA yang dapat memenuhi seluruh konstrain dan kebutuhan gereja?	Merancang algoritma hybrid PSO-GA yang memenuhi <i>multiple constraints</i> dan menghasilkan jadwal yang optimal.
Kompleksitas dalam mengelola konflik jadwal	Bagaimana tingkat efektivitas algoritma penjadwalan yang	Menganalisis efektivitas algoritma <i>hybrid</i> PSO-GA dalam mengatasi konflik

	dikembangkan dalam mengatasi permasalahan distribusi tugas, konflik jadwal, dan preferensi waktu petugas dibandingkan dengan sistem manual?	jadwal dibandingkan metode manual.
Kesulitan memenuhi preferensi dan kebutuhan dinamis gereja	Bagaimana mengimplementasikan pendekatan <i>Waterfall</i> dalam perancangan sistem penjadwalan untuk mengakomodasi kebutuhan yang dinamis dari HKBP Paranginan?	Mengkaji fleksibilitas algoritma dalam menghadapi perubahan kebutuhan dan memastikan kebutuhan dinamis terpenuhi.
Kinerja sistem manual yang tidak efisien	Bagaimana tingkat efektivitas sistem penjadwalan yang dikembangkan dalam mengatasi permasalahan distribusi tugas, konflik jadwal, dan preferensi waktu petugas dibandingkan dengan sistem manual?	Membandingkan efisiensi dan keadilan distribusi tugas antara algoritma hybrid PSO-GA dan sistem manual.

1.8 Manfaat Penelitian

Tugas akhir ini diharapkan dapat memberikan kontribusi praktis bagi HKBP Paranginan dalam meningkatkan efisiensi pengelolaan penjadwalan petugas ibadah. Penerapan sistem yang dikembangkan akan memberikan solusi terhadap berbagai

permasalahan yang selama ini dihadapi dalam penjadwalan manual. Berikut adalah manfaat praktis yang dapat diperoleh:

1. Membantu HKBP Paranginan mengoptimalkan proses penjadwalan petugas ibadah dengan sistem yang lebih efisien dibandingkan sistem manual, sehingga menghemat waktu dan tenaga dalam penyusunan jadwal.
2. Meningkatkan keadilan dalam distribusi tugas antar petugas ibadah, sehingga setiap anggota jemaat yang memiliki kemampuan mendapat kesempatan yang merata untuk melayani.
3. Mempermudah pengelolaan perubahan jadwal secara dinamis ketika ada petugas yang berhalangan hadir, sehingga tidak mengganggu kelancaran ibadah.
4. Membantu pengurus gereja dalam memenuhi preferensi waktu dari masing-masing petugas ibadah secara lebih optimal, sehingga meningkatkan kenyamanan dan partisipasi petugas.

1.9 Sistematika Penulisan

1.9.1 Bab I Pendahuluan

Bab ini menjelaskan konteks penelitian, identifikasi masalah, tujuan penelitian, batasan masalah, manfaat penelitian, dan sistematika penulisan. Latar belakang yang dibahas menguraikan pentingnya perancangan sistem penjadwalan berbasis algoritma hybrid PSO-GA untuk mengatasi permasalahan penjadwalan petugas ibadah di Gereja HKBP Paranginan.

1.9.2 Bab II Tinjauan Pustaka

Bab ini menyajikan tinjauan terhadap penelitian sebelumnya yang relevan sebagai dasar teoritis penelitian. Bagian ini mencakup kajian teori tentang algoritma PSO, Genetic Algorithm, kombinasi keduanya dalam *hybrid optimization*, metodologi Waterfall, dan penelitian serupa lainnya terkait optimasi penjadwalan.

1.9.3 Bab III Metode Penelitian

Bab ini menguraikan metode yang digunakan dalam penelitian, termasuk alur penelitian, langkah-langkah perancangan sistem, alat dan bahan yang digunakan, rancangan algoritma hybrid PSO-GA, pendekatan Waterfall dalam

pengembangan sistem, serta metode pengujian untuk mengevaluasi performa sistem penjadwalan.

1.9.5 Bab IV Hasil dan Pembahasan

Bab ini membahas hasil implementasi algoritma hybrid PSO-GA dan sistem penjadwalan yang dirancang. Analisis hasil mencakup evaluasi performa algoritma, perbandingan dengan metode manual, dan diskusi mengenai efektivitas pendekatan Waterfall dalam memenuhi kebutuhan dinamis gereja.

1.9.6 Bab V Kesimpulan dan Saran

Bab ini menyimpulkan hasil penelitian secara keseluruhan, menjawab rumusan masalah, dan memberikan rekomendasi untuk pengembangan lebih lanjut. Saran yang diberikan diharapkan dapat membantu optimalisasi implementasi sistem penjadwalan di gereja maupun pengembangan algoritma di masa mendatang.

BAB II

TINJAUAN PUSTAKA

2.1 Tinjauan Pustaka

Penelitian yang dilakukan oleh Hartono dan Zein pada tahun 2023 berfokus pada penerapan algoritma genetika dan jaringan syaraf tiruan dalam penjadwalan mata kuliah di Program Studi Sistem Informasi Universitas Pamulang. Masalah utama yang diangkat adalah kompleksitas penjadwalan yang melibatkan pengaturan ruang, waktu, preferensi dosen, dan kebutuhan mahasiswa.

Penelitian tersebut menggunakan pendekatan metode *System Development Life Cycle* (SDLC) model *waterfall* untuk pengembangan sistem penjadwalan. Algoritma genetika diterapkan untuk menghasilkan solusi awal yang efisien, sementara jaringan syaraf tiruan digunakan untuk mengoptimalkan jadwal berdasarkan berbagai kendala dan parameter relevan[5].

Hasil penelitian menunjukkan peningkatan efisiensi dalam proses penjadwalan, pengurangan konflik jadwal, serta kemampuan untuk menyesuaikan penjadwalan dengan kebutuhan institusi. Namun, penelitian ini belum mengintegrasikan analisis berbasis data waktu nyata atau model adaptif yang dapat memperhitungkan perubahan mendadak, seperti pembatalan kelas atau perubahan jadwal dosen.

Penelitian yang dilakukan oleh Muhammad Muhardeny, Muhammad Haviz Irfani, dan Juhaini Alie pada tahun 2023 berfokus pada penerapan algoritma *Particle Swarm Optimization* (PSO) dalam penjadwalan mata pelajaran di SMPIT Mufidatul Ilmi. Masalah utama yang diangkat adalah kompleksitas dalam penyusunan jadwal yang melibatkan banyak faktor, seperti kapasitas ruang kelas, kemampuan guru, waktu yang sesuai untuk setiap semester, dan potensi terjadinya konflik antara jadwal mengajar guru dan ruang yang tersedia.

Penelitian tersebut menggunakan metode pengembangan sistem dengan model *Waterfall* dalam proses pembuatan aplikasi penjadwalan. Algoritma PSO diterapkan untuk menyusun jadwal pelajaran dengan cara mencari solusi terbaik melalui optimasi posisi dan kecepatan partikel dalam suatu populasi. Dalam hal ini, setiap partikel

merepresentasikan solusi jadwal yang potensial, dan proses optimisasi dilakukan dengan tujuan meminimalkan pelanggaran terhadap berbagai kendala yang ada, seperti waktu dan ruang kelas yang terbatas[6].

Hasil penelitian menunjukkan bahwa penerapan PSO dapat menghasilkan jadwal yang lebih efisien dengan meminimalkan konflik pengajaran dan mengoptimalkan penggunaan ruang kelas. Namun, penelitian ini masih terbatas pada penggunaan data yang statis dan tidak mengakomodasi perubahan mendadak dalam jadwal, seperti pembatalan kelas atau perubahan jadwal guru.

Penelitian yang dilakukan oleh Siregar, dkk pada tahun 2021 mengembangkan sistem penjadwalan petugas ibadah gereja menggunakan algoritma genetika untuk mengurangi konflik jadwal dan meningkatkan efisiensi distribusi tugas. Meskipun penelitian ini berhasil menunjukkan penerapan algoritma genetika dalam penjadwalan, metode yang digunakan terbatas pada satu jenis algoritma optimasi dan belum menggabungkan berbagai algoritma untuk hasil yang lebih optimal. Penelitian yang akan dilakukan oleh penulis bertujuan untuk mengatasi keterbatasan tersebut dengan menggunakan *Hybrid PSO-Genetic Algorithm* yang mengkombinasikan keunggulan kedua algoritma dalam eksplorasi dan eksploitasi solusi optimal, serta pendekatan *Waterfall* untuk memberikan kerangka kerja yang lebih sistematis dalam merancang penjadwalan petugas ibadah Gereja HKBP Paranginan, yang diharapkan dapat memberikan solusi yang lebih efisien dan fleksibel dibandingkan dengan penelitian terdahulu[7].

Penelitian Yumarnis, dkk (2022) menggunakan algoritma genetika untuk menentukan jadwal kerja karyawan di Kantor PLN ULP Medan Selatan selama pandemi Covid-19. Penelitian ini bertujuan mengatasi masalah benturan jadwal yang sering terjadi akibat proses manual menggunakan *Microsoft Excel*. Algoritma genetika diaplikasikan dengan prinsip evolusi biologis untuk menghasilkan jadwal kerja yang optimal melalui kombinasi kromosom induk dan pembentukan kromosom baru. Hasil penelitian menunjukkan bahwa aplikasi penjadwalan yang dikembangkan mampu meningkatkan efisiensi kinerja karyawan di PLN Medan Selatan. Namun, penelitian ini terbatas pada satu jenis metode optimasi dan tidak mengeksplorasi metode lain yang dapat meningkatkan akurasi solusi[8].

Penelitian Syahputra, dkk (2023) mengimplementasikan algoritma *Particle Swarm Optimization* (PSO) dalam sistem informasi penjadwalan kelas di SMK Panca Budi 1 Medan. Penelitian ini bertujuan untuk mengoptimalkan pembagian waktu pelajaran dengan memastikan tidak adanya benturan jadwal antara guru yang mengajar di dua jurusan. Metode PSO digunakan untuk menyusun laporan jadwal secara terstruktur dan efisien, memberikan kemudahan bagi pengguna dalam membuat laporan jadwal kelas. Hasil penelitian ini menunjukkan bahwa algoritma PSO efektif dalam menghasilkan jadwal yang tidak bertabrakan, sehingga mendukung kelancaran proses pembelajaran di sekolah[9].

Tabel 2.1 Review Literatur

No	Penulis [Judul] [Tahun]	Permasalahan	Metode	Hasil Penelitian	Perbedaan
1.	Krisno & Kuswanto [Sistem Informasi Penjadwalan Kunjungan Teknisi Menggunakan Algoritma Genetika, 2024]	Penjadwalan rute teknisi yang dilakukan manual menyebabkan pemborosan waktu dan biaya.	Algoritma Genetika, pendekatan heuristik.	Meningkatkan efisiensi waktu, penghematan bahan bakar, dan pengurangan kesalahan penjadwalan.	Fokus pada optimasi rute logistik teknisi menggunakan TSP, sedangkan penelitian Saya fokus pada penjadwalan petugas gereja dengan banyak konstrain berbeda.
2.	Muhardeny et al. [Penjadwalan Mata Pelajaran Menggunakan PSO, 2023]	Kompleksitas penjadwalan mata pelajaran dengan kendala ruang, waktu, dan preferensi guru.	<i>Particle Swarm Optimization</i> (PSO).	Menghasilkan jadwal dengan tingkat efisiensi tinggi dan meminimalkan konflik.	Penelitian ini menggunakan PSO untuk pendidikan, sedangkan penelitian Saya menggabungkan PSO dan GA untuk mengoptimalkan jadwal petugas gereja dengan preferensi waktu.

3.	Hartono & Zein [Penerapan Algoritma Genetika dan Jaringan Syaraf Tiruan dalam Penjadwalan Mata Kuliah, 2023]	Konflik jadwal dan kebutuhan prediksi jumlah peserta kuliah.	Algoritma Genetika dan Jaringan Syaraf Tiruan.	Efisiensi dalam penjadwalan, pengurangan konflik, dan prediksi kebutuhan mata kuliah lebih akurat.	Mengintegrasikan prediksi menggunakan Jaringan Syaraf Tiruan, sementara penelitian Saya berfokus pada <i>hybrid</i> PSO-GA untuk menyelesaikan masalah optimasi.
4.	Hindra Syahputra et al. [<i>Application of Particle Swarm Optimization Algorithm in Information Systems Class Scheduling</i> , 2023]	Penjadwalan kelas manual menyebabkan benturan jadwal antara guru di dua jurusan berbeda.	<i>Particle Swarm Optimization</i> (PSO)	Menghasilkan jadwal kelas terorganisasi tanpa benturan jadwal antara guru.	Menggunakan PSO untuk optimasi jadwal kelas berbasis sekolah, sedangkan penelitian Saya mengoptimalkan jadwal petugas gereja dengan banyak kendala dan preferensi waktu.
5.	Nina Yumarnis et al. [<i>The Use of Genetic Algorithm in Determining Employee Schedules at PLN ULP Office Selatan</i> , 2023]	Jadwal shift karyawan manual sering menyebabkan benturan dan kinerja yang kurang optimal.	Algoritma Genetika	Meningkatkan efisiensi dan akurasi dalam penjadwalan shift karyawan.	Menggunakan Algoritma Genetika untuk penjadwalan shift karyawan, sementara penelitian Saya menggabungkan PSO dan GA untuk jadwal petugas gereja.

2.2 Dasar Teori

Penjadwalan adalah proses pengaturan sumber daya terbatas, seperti waktu, tenaga pengajar, dan ruangan, agar dapat digunakan secara efisien untuk mendukung pelaksanaan kegiatan tertentu, seperti perkuliahan atau kegiatan ibadah. Dalam

implementasinya, penjadwalan melibatkan berbagai kendala (*constraints*), seperti ketersediaan dosen, ruangan, serta slot waktu, sehingga penyusunan secara manual sering kali menjadi rumit dan membutuhkan waktu yang lama. Berbagai penelitian telah menunjukkan bahwa algoritma optimasi, seperti *Genetic Algorithm* (GA), *Particle Swarm Optimization* (PSO), dan *Ant Colony Optimization* (ACO), dapat diterapkan untuk menyelesaikan masalah penjadwalan dengan menghasilkan solusi yang optimal. Algoritma-algoritma tersebut dikelompokkan ke dalam metode *heuristic*, *metaheuristic*, dan *hyperheuristic*, yang masing-masing memiliki keunggulan dalam meningkatkan efisiensi waktu komputasi serta kualitas hasil penjadwalan. Dengan memanfaatkan algoritma optimasi, tantangan dalam proses manual dapat diatasi, meskipun faktor seperti ukuran data yang besar dapat memengaruhi durasi proses komputasi. Kajian terhadap metode dan algoritma ini memberikan dasar teori yang kuat untuk mengembangkan sistem penjadwalan yang lebih efisien dan efektif di berbagai bidang[10].

Sistem informasi dirancang untuk meningkatkan efisiensi pengelolaan data dan mendukung proses pengambilan keputusan, terutama dalam lingkungan organisasi atau komunitas. Dalam konteks pengelolaan jadwal, teknologi berbasis algoritma seperti *Hybrid Particle Swarm Optimization* (PSO) dan *Genetic Algorithm* (GA) dapat digunakan untuk menghasilkan solusi optimal yang memenuhi berbagai kendala dan kebutuhan. Pendekatan pengembangan sistem seperti model *Waterfall* memungkinkan proses perancangan dilakukan secara terstruktur, mulai dari analisis kebutuhan hingga implementasi dan pemeliharaan[11]. Dengan metode ini, setiap tahap dapat diselesaikan secara sistematis, sehingga menghasilkan sistem yang lebih andal dan sesuai dengan kebutuhan pengguna.

2.2.1 Penjadwalan

Penjadwalan, menurut Kenneth R. Baker (2009), adalah suatu proses yang melibatkan pengalokasian sumber daya yang ada untuk menjalankan sekumpulan tugas dalam periode waktu tertentu yang telah ditetapkan. Proses ini tidak hanya meliputi pembagian waktu untuk setiap tugas, tetapi juga mencakup perencanaan yang matang, pemilihan sumber daya yang tepat, serta penentuan waktu yang optimal untuk penggunaan sumber daya tersebut[12]. Tujuan utama dari penjadwalan adalah untuk

mencapai hasil yang diinginkan dalam kerangka waktu yang telah ditentukan, dengan memaksimalkan efisiensi dan efektivitas dalam penggunaan sumber daya yang tersedia.

Dalam konteks manajerial, penjadwalan sering kali menjadi bagian integral dari perencanaan strategis yang lebih luas. Hal ini memungkinkan organisasi atau individu untuk mengelola waktu, tenaga, dan materi secara lebih terstruktur, sehingga meminimalkan pemborosan dan mengoptimalkan pencapaian tujuan. Penjadwalan juga dapat mencakup proses pemantauan dan penyesuaian, mengingat bahwa situasi dan prioritas dapat berubah selama periode pelaksanaan[13]. Dengan demikian, penjadwalan bukan hanya sekadar menentukan kapan suatu tugas harus diselesaikan, tetapi juga bagaimana cara terbaik untuk menyelesaikan tugas tersebut dengan sumber daya yang terbatas.

2.2.2 Particle Swarm Optimization Algorithm

Particle Swarm Optimization (PSO) adalah algoritma optimisasi berbasis populasi yang terinspirasi oleh perilaku sosial hewan, seperti kawanan burung atau sekolah ikan[14]. Algoritma ini pertama kali diperkenalkan oleh James Kennedy dan Russell Eberhart pada tahun 1995. PSO bekerja dengan menggerakkan partikel-partikel dalam ruang pencarian untuk menemukan solusi optimal. Setiap partikel dalam populasi mewakili solusi potensial dan memiliki dua atribut utama, yaitu posisi dan kecepatan. Posisi menggambarkan solusi yang mungkin, sementara kecepatan menggambarkan arah dan langkah yang diambil partikel untuk bergerak menuju solusi yang lebih baik. Kecepatan partikel diperbarui berdasarkan dua komponen utama, yaitu komponen pribadi dan komponen sosial. Komponen pribadi mengarah ke posisi terbaik yang pernah ditemukan oleh partikel itu sendiri, sedangkan komponen sosial mengarah ke posisi terbaik yang ditemukan oleh seluruh populasi. Pembaruan kecepatan partikel dilakukan dengan rumus:

$$v_i(t + 1) = w \cdot v_i(t) + c_1 \cdot r_1 \cdot (pbest_i - x_i(t)) + c_2 \cdot r_2 \cdot (gbest - x_i(t)) \quad (2.1)$$

Keterangan :

$v_i(t)$: kecepatan partikel ke-i pada iterasi t

w : faktor inersia yang mengontrol pengaruh kecepatan sebelumnya

- c_1 dan c_2 : konstanta yang mengontrol pengaruh komponen pribadi dan sosial
 r_1 dan r_2 : bilangan acak antara 0 dan 1
 $pbest_i$: posisi terbaik yang ditemukan oleh partikel ke- i
 $gbest$: posisi terbaik yang ditemukan oleh seluruh populasi.

Setelah kecepatan diperbarui, posisi partikel juga diperbarui dengan rumus:

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (2.2)$$

Posisi baru ini menggambarkan solusi potensial yang lebih baik. Setiap partikel terus mengevaluasi kualitas solusinya dengan menggunakan fungsi objektif yang mengukur seberapa baik solusi tersebut. Jika posisi baru memberikan nilai fungsi objektif yang lebih baik, maka posisi tersebut akan menjadi posisi terbaik pribadi ($pbest$) untuk partikel tersebut. Posisi terbaik global ($gbest$) diperbarui jika ada partikel yang menemukan posisi yang lebih baik daripada posisi terbaik yang ditemukan oleh seluruh populasi[15].

Proses ini diulang selama sejumlah iterasi atau sampai konvergensi tercapai, yaitu ketika perubahan posisi menjadi sangat kecil atau solusi optimal ditemukan. PSO memiliki keunggulan utama berupa kesederhanaan dan kemampuan untuk melakukan pencarian solusi secara global, tanpa terjebak pada solusi lokal. PSO juga sangat efektif dalam menyelesaikan masalah optimisasi yang kompleks, seperti masalah non-linear dan non-convex. Namun, PSO juga memiliki beberapa kekurangan, seperti kemungkinan konvergensi prematur ke solusi lokal dan ketergantungan pada pemilihan parameter yang tepat, seperti faktor inersia dan konstanta sosial.

PSO telah diterapkan dalam berbagai bidang, termasuk optimisasi fungsi matematis, desain jaringan, pengendalian sistem dinamis, serta dalam pengenalan pola dan machine learning. Dengan kemampuannya untuk mencari solusi dalam ruang pencarian yang besar, PSO menjadi alat yang sangat berguna dalam menyelesaikan masalah yang sulit dijangkau oleh metode optimisasi tradisional[16]. Meskipun demikian, pemilihan parameter yang tepat dan pengaturan algoritma yang hati-hati tetap diperlukan untuk memastikan kinerja PSO yang optimal.

2.2.2 Genetic Algorithm

Algoritma genetika merupakan metode pencarian dan optimasi yang terinspirasi dari proses evolusi biologis dan seleksi alam. Dikembangkan pertama kali oleh John

Holland pada tahun 1975, algoritma ini menerapkan prinsip-prinsip genetika seperti seleksi, *crossover* (penyilangan), dan mutasi untuk mencari solusi optimal dari suatu permasalahan. Dalam konteks komputasi, algoritma genetika bekerja dengan populasi kandidat solusi yang direpresentasikan dalam bentuk kromosom[17].

Fundamental dari algoritma genetika terletak pada teori evolusi Darwin, di mana individu yang paling sesuai dengan lingkungannya memiliki probabilitas lebih tinggi untuk bertahan hidup dan menghasilkan keturunan. Dalam implementasi komputasional, setiap kromosom merepresentasikan satu solusi potensial, dan nilai *fitness function* menentukan seberapa baik solusi tersebut dalam menyelesaikan permasalahan yang diberikan[18].

$$\text{Nilai } fitness = \frac{1}{1+f(x)} \quad (2.3)$$

Keterangan : $f(x)$ adalah fungsi objek dari problem yang dituntaskan

Adapun langkah penyelesaian masalah *genetic algorithm* adalah sebagai berikut:

1. Inisialisasi populasi, inisialisasi populasi merupakan tahap awal dalam algoritma genetika di mana sejumlah kromosom dibangkitkan secara acak sebagai kandidat solusi. Setiap kromosom terdiri dari gen-gen yang merepresentasikan variabel-variabel penyelesaian masalah, dengan panjang kromosom dan ukuran populasi ditentukan berdasarkan kompleksitas permasalahan yang akan diselesaikan.
2. *Fitness value*, merupakan tahap evaluasi setiap kromosom dalam populasi untuk mengukur seberapa baik solusi yang dihasilkan. Proses ini menggunakan fungsi *fitness* yang telah didefinisikan sesuai dengan tujuan optimasi, di mana semakin tinggi nilai *fitness* suatu kromosom, semakin baik solusi yang direpresentasikannya untuk menyelesaikan permasalahan yang ada.
3. Seleksi, merupakan proses pemilihan kromosom yang akan menjadi parent untuk generasi berikutnya berdasarkan nilai *fitness* masing-masing kromosom. Metode seleksi yang umum digunakan adalah *roulette wheel selection*, di mana kromosom dengan nilai *fitness* lebih tinggi memiliki probabilitas lebih besar untuk terpilih sebagai parent dalam proses reproduksi.
4. Pindah silang, atau *crossover* adalah proses mengombinasikan material genetik dari dua kromosom parent untuk menghasilkan kromosom *offspring* baru. Proses ini dilakukan dengan probabilitas *crossover* tertentu dan dapat menggunakan berbagai

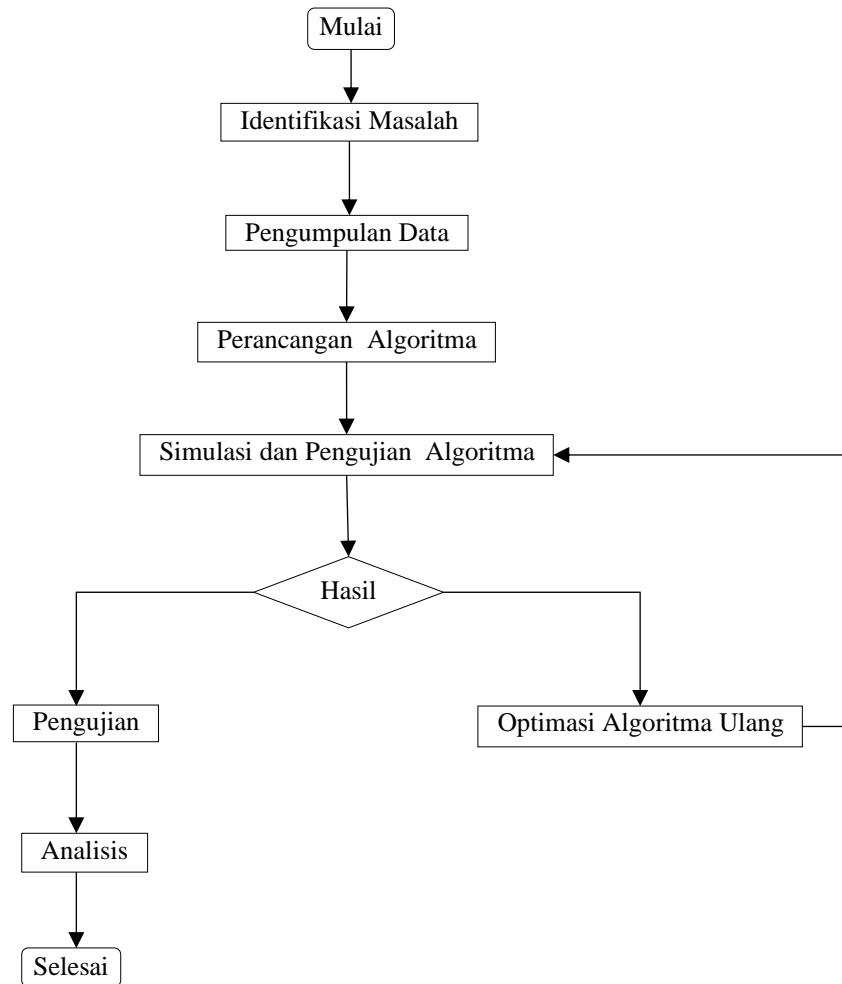
metode seperti *single-point crossover*, *multi-point crossover*, atau *uniform crossover* untuk menghasilkan variasi genetik dalam populasi.

5. Mutasi, adalah proses mengubah nilai gen secara acak pada kromosom dengan probabilitas yang sangat kecil. Tahap ini penting untuk mempertahankan keragaman genetik dalam populasi dan mencegah konvergensi prematur pada solusi lokal, sehingga algoritma dapat terus mengeksplorasi ruang pencarian untuk menemukan solusi yang lebih optimal[19].

BAB III

METODE PENELITIAN

3.1 Desain Penelitian



Gambar 3.1 Diagram Penelitian

3.2 Penjabaran Langkah Penelitian

3.2.1 Identifikasi Masalah

Dalam proses penelitian ini, identifikasi masalah dilakukan melalui metode wawancara langsung dengan pengurus dan petugas ibadah di Gereja HKBP Sabungan Paranginan. Wawancara ini bertujuan untuk menggali informasi mendalam terkait tantangan yang dihadapi dalam penjadwalan petugas ibadah, khususnya dalam memenuhi berbagai kriteria dan kendala operasional. Melalui interaksi langsung

dengan para responden, diperoleh data yang kaya tentang proses manual yang digunakan, permasalahan distribusi tugas, konflik jadwal, serta keterbatasan sumber daya. Pendekatan ini memungkinkan peneliti untuk memahami kebutuhan nyata dari pihak-pihak yang terlibat secara langsung, sehingga dapat merumuskan solusi yang relevan dan sesuai dengan kondisi lapangan.

Beberapa masalah utama yang teridentifikasi melibatkan keterbatasan ketersediaan petugas dan distribusi beban tugas yang tidak merata. Tidak semua jemaat dapat bertugas di semua peran karena adanya kriteria tertentu, seperti pengkotbah yang hanya dapat diisi oleh Pendeta atau Guru Huria, serta pemain musik dan operator yang memerlukan keahlian khusus. Hal ini menyulitkan penyusunan jadwal, terutama ketika jumlah petugas yang memenuhi syarat terbatas. Selain itu, tidak semua petugas tersedia untuk semua sesi (pagi atau siang), sehingga distribusi tugas menjadi lebih kompleks.

Masalah lain yang ditemukan adalah konflik jadwal antara petugas yang memiliki multiple roles. Misalnya, Sintua yang dapat bertugas sebagai pemimpin doa dan penjaga pintu sering menghadapi konflik jika kedua tugas tersebut harus dijalankan di waktu yang bersamaan. Sistem manual sulit untuk melakukan pergantian petugas secara cepat jika ada yang berhalangan hadir.

Rotasi kelompok singer juga menjadi tantangan besar, karena kelompok koor yang sama sering kali bertugas lebih dari sekali dalam satu hari akibat keterbatasan rotasi manual. Hal ini menyebabkan kelelahan pada kelompok tersebut dan kurangnya representasi kelompok koor lain. Pengaturan rotasi kelompok koor yang adil menjadi tantangan, terutama jika kelompok tertentu memiliki preferensi waktu atau keterbatasan jumlah anggota.

Selain itu, pengaturan durasi dan waktu istirahat petugas juga menjadi masalah. Petugas yang bertugas di sesi pagi memerlukan waktu istirahat yang memadai sebelum kembali bertugas di sesi siang, tetapi sistem manual kesulitan memastikan waktu jeda yang cukup, terutama saat petugas yang sama bertugas di kedua sesi. Ketidakefisienan ini dapat mengganggu kualitas pelayanan karena kelelahan petugas.

Keterbatasan jumlah petugas dalam kategori tertentu, seperti pengkotbah dan pemain musik, menyebabkan tugas-tugas tersebut sering diberikan kepada individu yang sama

berulang kali. Untuk peran singer, ada batasan maksimal tiga orang per sesi dengan representasi dari kelompok koor yang berbeda, sehingga koordinasi menjadi tantangan besar.

3.2.2 Pengumpulan Data

Teknik pengumpulan data yang diterapkan adalah melalui wawancara mendalam dengan Pdt. Simajuntak, seorang pendeta dari gereja HKBP Sabungan Paranginan. Wawancara ini bertujuan untuk menggali informasi mengenai situasi dan kondisi gereja, termasuk kebijakan terkait pelayanan serta masalah-masalah yang dihadapi oleh gereja dalam menjalankan berbagai kegiatan. Wawancara ini dilakukan secara langsung untuk mendapatkan informasi yang lebih mendalam dan akurat dari narasumber yang memiliki pengetahuan dan pengalaman di lapangan.

Selain wawancara dengan Pdt. Simajuntak, penulis juga mengumpulkan data terkait ketersediaan petugas di gereja HKBP Sabungan Paranginan. Data ini diperoleh melalui observasi dan pencatatan terhadap jumlah dan peran petugas yang terlibat dalam pelayanan gereja, baik dalam kegiatan ibadah, pengelolaan administrasi gereja, maupun kegiatan sosial. Data ini sangat penting untuk mengetahui sejauh mana gereja memiliki sumber daya manusia yang memadai untuk menjalankan aktivitasnya dengan efisien dan efektif, serta untuk melihat potensi dan tantangan yang ada dalam pengelolaan gereja.

Tabel 3.1 Data Set Ketersediaan Pelayan Gereja

Pdt. L. Simanjuntak	Pendeta
Gr. T. Hutabarat	Guru Huria
St. B. Sitorus	Sintua
St. R. Silalahi	Sintua
St J. Manik	Sintua
St. S. Simbolon	Sintua
St. H. Sihombing	Sintua
St. M. Purba	Sintua
St. N. Napitupulu	Sintua
St. T. Lumban Gaol	Sintua
St. P. Nainggolan	Sintua

St. S. Saragih	Sintua
St. M. Sitompul	Sintua
St. D. Marbun	Sintua
St. H. Pardede	Sintua
St. L. Nababan	Sintua
St. J. Sitorus	Sintua
St. E. Silaban	Sintua
St. F. Hutapea	Sintua
St. G. Manurung	Sintua
St. R. Simanungkalit	Sintua
St. A. Sinaga	Sintua
St. Y. Panjaitan	Sintua
St. K. Sidabutar	Sintua
St. U. Pasaribu	Sintua
St. Z. Hutajulu	Sintua
St. V. Panggabean	Sintua
Joice	Pianis
Yuni	Pianis
Daniel	Terompet
David	Terompet
Jhon	Pianis
James	Terompet

Tabel 3.2 Data Set Operator

Daftar Sintua yang Bisa Operator	
St. L. Nababan	Sintua
St. J. Sitorus	Sintua
St. E. Silaban	Sintua
St. F. Hutapea	Sintua

Selain itu, penulis juga melakukan wawancara dengan beberapa jemaat gereja untuk menggali keluhan dan harapan mereka terhadap pelayanan yang ada di gereja. Wawancara ini dilakukan secara terbuka dan tanpa paksaan, dengan tujuan untuk

mendapatkan gambaran yang objektif mengenai persepsi jemaat terhadap pelayanan gereja dan faktor-faktor yang mempengaruhi kepuasan mereka. Data yang diperoleh dari wawancara ini diharapkan dapat memberikan insight yang berharga dalam upaya perbaikan kualitas pelayanan di gereja HKBP Sabungan Paranginan, baik dari sisi manajerial maupun spiritual.

Tabel 3.3 Tabel Data Keluhan Hasil Wawancara

Nama Petugas	Peran	Tanggal Bertugas	Waktu	Kelompok	Masalah yang Dihadapi
Pdt. L. Simanjuntak	Pengkotbah	22 Oktober 2024	Pagi	Pendeta	Hanya satu pengkotbah tersedia, bertugas 4 kali berturut-turut dalam satu bulan
St. A. Sinaga	Pemimpin Kebaktian	1 Desember 2024	Pagi	Sintua	Konflik dengan jadwal sebagai pengumpul persembahan di sesi yang sama
Joice	Pemain Musik	1 Desember 2024	Siang	Pianis	Bertugas di kedua sesi tanpa jeda waktu yang memadai
Jakkon	Singer	15 Desember 2024	Siang	Koor Ama Exaudi	Koor Ama Exaudi bertugas dua kali di hari yang sama
St. E. Silaban	Pengumpul Persembahan	15 Desember 2024	Pagi	Sintua	Tidak dapat hadir, sulit mencari pengganti mendadak
St. F. Hutapea	Operator Multimedia	15 Desember 2024	Pagi	Operator Multimedia	Tidak mendapat tugas selama dua bulan terakhir

3.2.3 Perancangan Algoritma

Sebelum melakukan perancangan algoritma, langkah pertama yang dilakukan yaitu dengan membuat *constraint* yang akan diterapkan dalam implementasi kode. Berikut adalah *constraint* terkait penjadwalan :

1. Struktur Jadwal Dasar

Sistem menghasilkan jadwal untuk satu bulan penuh, dengan dua kebaktian setiap hari Minggu:

- Kebaktian Pagi (08:00-10:30)
- Kebaktian Siang (11:00-13:30)

2. Kategori dan Peran Personil

a. Peran Utama

1) Kepemimpinan Gereja

- Pendeta
- Guru Huria
- Sintua

2) Pendukung Kebaktian

- Pemusik (Pianis dan Pemain Terompet)
- Operator
- Kelompok Paduan Suara

3. Posisi yang Dibutuhkan Per Kebaktian

Setiap kebaktian harus memiliki:

- 1) 1 Pengkotbah
- 2) 1 Pemimpin Kebaktian
- 3) 1 Pianis
- 4) 1 Pemain Terompet
- 5) 1 Operator
- 6) 2 Penjaga Pintu
- 7) 3 Kelompok Paduan Suara
- 8) 3 Pengumpul Persembahan
- 9) 1 Pembaca Warta
- 10) 1 Pemimpin Doa

4. Kebutuhan Petugas Cadangan

Setiap kebaktian juga harus memiliki petugas cadangan:

- 1) 1 Cadangan Pengkotbah
- 2) 1 Cadangan Pemimpin Kebaktian

- 3) 1 Cadangan Pianis
- 4) 1 Cadangan Pemain Terompet
- 5) 1 Cadangan Operator
- 6) 2 Cadangan Sintua

5. Konstrain Penugasan Personil

1) Kelayakan Peran

- Posisi Pengkotbah hanya dapat diisi oleh Pendeta atau Guru Huria
- Posisi Pemimpin Kebaktian dapat diisi oleh Guru Huria atau Sintua
- Posisi musik harus diisi oleh pemusik yang berkualifikasi (Pianis atau pemain Terompet sesuai peran)
- Posisi operator harus diisi oleh operator yang ditunjuk
- Posisi penjaga pintu, pengumpul persembahan, pembaca warta, dan pemimpin doa harus diisi oleh Sintua

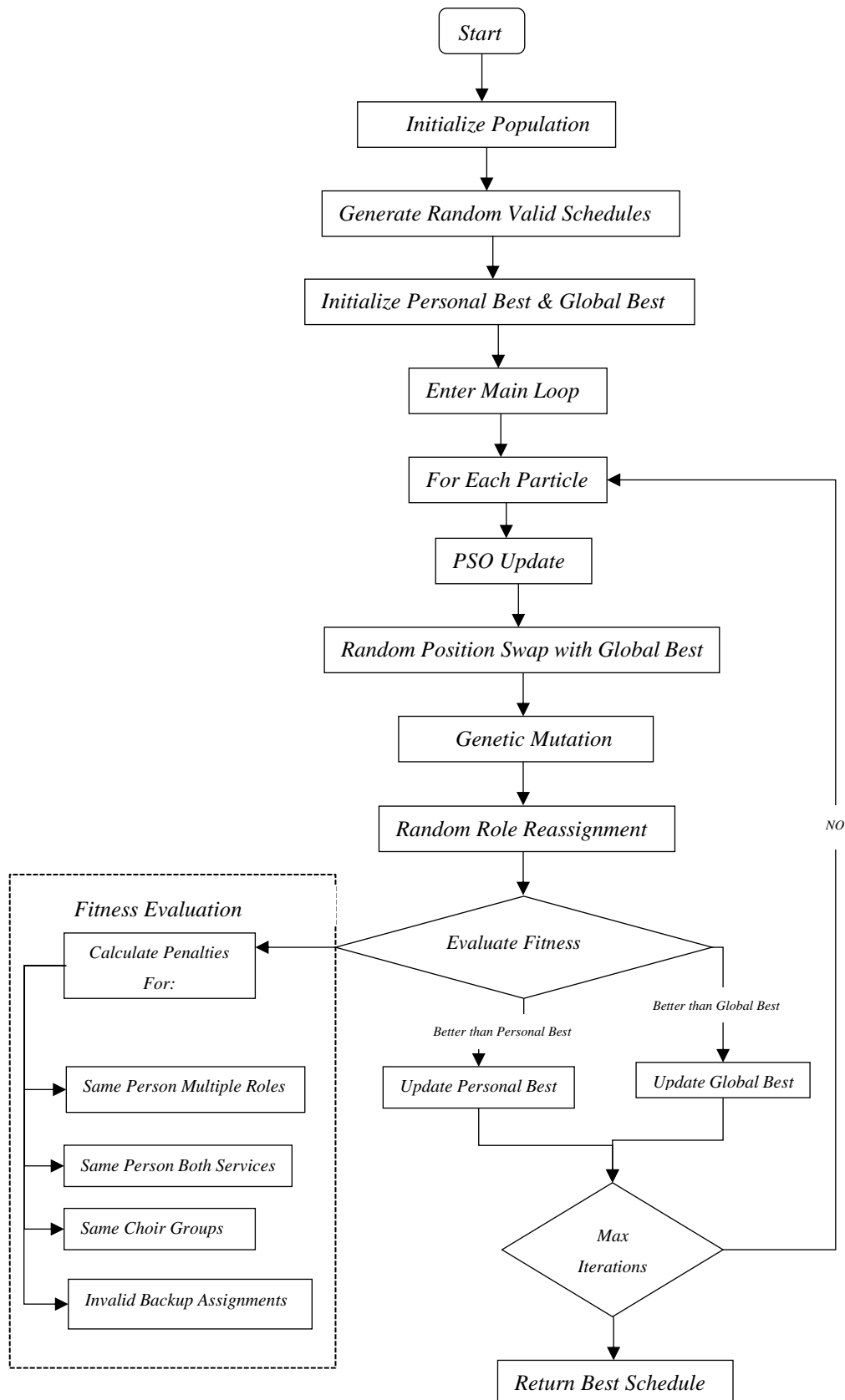
2) Pembatasan Layanan Hari yang Sama

- Orang yang sama tidak boleh bertugas dalam beberapa peran dalam satu kebaktian
- Orang yang sama tidak boleh bertugas di kebaktian pagi dan siang pada hari yang sama
- Kelompok paduan suara yang sama tidak boleh tampil di kebaktian pagi dan siang

3) Pembatasan Petugas Cadangan

- Petugas cadangan tidak boleh sama dengan petugas utama untuk setiap peran
- Petugas cadangan harus memiliki kualifikasi peran yang sama dengan petugas utama

Berikut rancangan Algoritma hybrid PSO-Genetic



Gambar 3.2 Diagram Rancangan Algoritma

1. Inisialisasi populasi dilakukan melalui metode `_initialize_particle` dalam kelas `ChurchService`. Metode ini membuat jadwal acak untuk sejumlah kebaktian tertentu (`num_services`). Setiap kebaktian direpresentasikan sebagai sebuah dictionary yang berisi peran-peran tertentu (misalnya, pengkotbah, pemimpin kebaktian, pianis, dll.), yang diisi secara acak dengan memilih personel yang sesuai dari dictionary `self.personnel` yang sudah ditentukan sebelumnya. Untuk peran yang membutuhkan beberapa orang, seperti penjaga_pintu atau pengumpul_persembahan, digunakan fungsi `random.sample`, sedangkan untuk peran tunggal digunakan `random.choice`. Selain petugas utama, jadwal juga mencakup petugas cadangan untuk setiap peran guna menyediakan alternatif jika diperlukan. Jadwal yang dihasilkan untuk semua kebaktian ini kemudian dikembalikan sebagai sebuah daftar, yang merepresentasikan satu "partikel" dalam algoritma *Particle Swarm Optimization* (PSO). Proses inisialisasi acak ini memastikan keberagaman dalam populasi awal, yang penting untuk keberhasilan proses optimasi.

```
class ChurchService:
    def __init__(self):
        # Initialize all personnel
        self.personnel = {
            'pendeta': [Person('Pdt. L. Simanjuntak', 'Pendeta')],
            'guru_huria': [Person('Gr. T. Hutabarat', 'Guru Huria')],
            'sintua': [
                Person(f"St. {name}", 'Sintua') for name in [
                    'B. Sitorus', 'R. Silalahi', 'J. Manik', 'S. Simbolon',
                    'H. Sihombing', 'M. Purba', 'N. Napitupulu', 'T. Lumban Gaol',
                    'P. Nainggolan', 'S. Saragih', 'M. Sitompul', 'D. Marbun',
                    'H. Pardede', 'L. Nababan', 'J. Sitorus', 'E. Silaban',
                    'F. Hutapea', 'G. Manurung', 'R. Simanungkalit', 'A. Sinaga',
                    'Y. Panjaitan', 'K. Sidabutar', 'U. Pasaribu', 'Z. Hutajulu',
                    'V. Panggabean'
                ]
            ],
        ],
```

Gambar 3.3 Inisialisasi Populasi

2. *Generate Random Valid Schedules* dilakukan oleh metode `_initialize_particle`, yang bertujuan untuk membuat jadwal acak yang valid untuk sejumlah kebaktian tertentu (`num_services`). Metode ini memastikan bahwa setiap kebaktian diisi dengan petugas utama dan cadangan yang dipilih secara acak dari daftar personel yang relevan di `self.personnel`. Setiap jadwal mencakup peran-peran seperti pengkotbah, pemimpin kebaktian, pianis, pemain terompet, operator, penjaga pintu, pengumpul persembahan, singer, pembaca warta, dan pemimpin doa. Untuk memastikan validitas, pemilihan

personel memperhatikan jenis peran (misalnya, hanya pianis untuk peran pianis), dan peran yang membutuhkan lebih dari satu orang (seperti penjaga_pintu atau pengumpul_persembahan) dipilih menggunakan *random.sample*. Jadwal ini juga mencakup petugas cadangan untuk setiap peran, yang juga dipilih secara acak dari kategori personel yang sesuai. Dengan pendekatan ini, metode tersebut menghasilkan jadwal acak yang cukup beragam, sekaligus memenuhi struktur yang diperlukan untuk kebaktian yang valid.

```
def _initialize_particle(self, num_services):
    # Initialize a random valid schedule for the specified number of services
    schedule = []
    for _ in range(num_services):
        service = {
            'pengkotbah': random.choice(self.personnel['pendeta'] + self.personnel['guru_huria']),
            'pemimpin_kebaktian': random.choice(self.personnel['guru_huria'] + self.personnel['sintua']),
            'pianis': random.choice([p for p in self.personnel['musicians'] if p.role == 'Pianis']),
            'terompet': random.choice([p for p in self.personnel['musicians'] if p.role == 'Terompet']),
            'operator': random.choice(self.personnel['operators']),
            'penjaga_pintu': random.sample(self.personnel['sintua'], 2),
            'singer': random.sample(self.personnel['choir_groups'], 3),
            'pengumpul_persembahan': random.sample(self.personnel['sintua'], 3),
            'pembaca_warta': random.choice(self.personnel['sintua']),
            'pemimpin_doa': random.choice(self.personnel['sintua']),
            # Add backup personnel with matching role names
            'cadangan_pengkotbah': random.choice(self.personnel['pendeta'] + self.personnel['guru_huria']),
            'cadangan_pemimpin_kebaktian': random.choice(self.personnel['guru_huria'] + self.personnel['sintua']),
            'cadangan_pianis': random.choice([p for p in self.personnel['musicians'] if p.role == 'Pianis']),
            'cadangan_terompet': random.choice([p for p in self.personnel['musicians'] if p.role == 'Terompet']),
            'cadangan_operator': random.choice(self.personnel['operators']),
            'cadangan_sintua': random.sample(self.personnel['sintua'], 2)
        }
        schedule.append(service)
    return schedule
```

Gambar 3.4 Generate Random Valid Schedules

3. *Initialize Personal Best & Global Best* dilakukan pada bagian awal algoritma *Particle Swarm Optimization (PSO)* di metode *generate_schedule*. Setelah populasi awal (kumpulan partikel) dibuat melalui *_initialize_particle*, setiap partikel dianggap sebagai solusi potensial untuk masalah penjadwalan.

- *Personal Best*: Setiap partikel memiliki *personal best*, yaitu salinan dari jadwal awalnya sendiri. Hal ini disimpan di variabel *personal_best* sebagai representasi solusi terbaik yang dicapai oleh partikel tersebut sejauh ini, berdasarkan nilai *fitness*-nya.
- *Global Best*: Dari semua partikel dalam populasi, dipilih satu partikel dengan nilai *fitness* terbaik (yaitu, dengan penalti terendah). Partikel ini dianggap sebagai solusi terbaik secara global, yang disimpan di variabel *global_best*. Seleksi dilakukan dengan fungsi *min*, yang membandingkan setiap partikel berdasarkan hasil evaluasi *fitness function*.

Proses inisialisasi ini memastikan bahwa algoritma PSO memiliki acuan awal untuk mengevaluasi dan memperbarui solusi, baik di tingkat individu partikel (*personal best*) maupun di tingkat populasi keseluruhan (*global best*). Selanjutnya, global best akan menjadi panduan bagi seluruh partikel selama iterasi untuk mencapai solusi optimal.

```
# Initialize particles with correct number of services
particles = [self._initialize_particle(num_services) for _ in range(num_particles)]
personal_best = copy.deepcopy(particles)
global_best = min(particles, key=lambda x: self._fitness_function(x))

# PSO-GA hybrid optimization
for iteration in range(50):
    for i in range(num_particles):
        new_particle = self._pso_update(particles[i], global_best)
        new_particle = self._genetic_mutation(new_particle)

        if self._fitness_function(new_particle) < self._fitness_function(personal_best[i]):
            personal_best[i] = copy.deepcopy(new_particle)

        if self._fitness_function(new_particle) < self._fitness_function(global_best):
            global_best = copy.deepcopy(new_particle)

    particles[i] = new_particle

return self._format_schedule(global_best, month, year)
```

Gambar 3.5 Initialize Personal Best & Global Best

4. *Enter Main Loop* mengacu pada bagian utama algoritma optimasi PSO-GA *hybrid* di metode *generate_schedule*. Pada bagian ini, proses iterasi dimulai untuk memperbaiki solusi yang dihasilkan dari populasi awal. Berikut adalah penjelasan langkah-langkahnya:

1. Iterasi (*for loop*): Algoritma dijalankan dalam beberapa iterasi (contoh: 50 iterasi), yang merepresentasikan proses eksplorasi dan eksploitasi solusi selama pencarian jadwal terbaik.
2. Perbarui Setiap Partikel: Untuk setiap partikel dalam populasi:
 - Dilakukan pembaruan menggunakan metode *_pso_update*, di mana beberapa bagian jadwal partikel saat ini disesuaikan dengan *global best* (solusi terbaik global). Proses ini mencerminkan mekanisme partikel "bergerak" menuju solusi terbaik.
 - Setelah itu, jadwal yang telah diperbarui akan dimutasi menggunakan metode *_genetic_mutation*. Mutasi ini bertujuan menambahkan variasi acak untuk mencegah algoritma terjebak di solusi lokal.
3. Evaluasi *Fitness*: Setelah pembaruan, nilai *fitness* dari partikel yang baru diperbarui dihitung menggunakan metode *_fitness_function*. *Fitness* diukur

berdasarkan penalti yang diberikan untuk pelanggaran aturan (misalnya, konflik jadwal atau pengurangan personel).

4. Perbarui *Personal Best*: Jika nilai *fitness* partikel yang baru diperbarui lebih baik daripada *personal best*-nya saat ini, maka *personal best* partikel tersebut digantikan dengan jadwal baru.
5. Perbarui *Global Best*: Jika nilai *fitness* partikel yang baru diperbarui lebih baik daripada *global best* (solusi terbaik di seluruh populasi), maka *global best* diperbarui dengan jadwal baru tersebut.
6. *Loop* Berlanjut: Proses ini diulangi untuk setiap partikel dan setiap iterasi hingga seluruh iterasi selesai.

```
def _genetic_mutation(self, particle):
    new_particle = copy.deepcopy(particle)

    if random.random() < 0.2: # 20% chance of mutation
        service_idx = random.randint(0, len(particle) - 1)
        role = random.choice(list(particle[service_idx].keys()))

        if role == 'pengkotbah' or role == 'cadangan_pengkotbah':
            new_particle[service_idx][role] = random.choice(self.personnel['pendeta'] + self.personnel['guru_huria'])
        elif role == 'pemimpin_kebaktian' or role == 'cadangan_pemimpin_kebaktian':
            new_particle[service_idx][role] = random.choice(self.personnel['guru_huria'] + self.personnel['sintua'])
        elif role == 'singer':
            new_particle[service_idx][role] = random.sample(self.personnel['choir_groups'], 3)
        elif role == 'pianis' or role == 'cadangan_pianis':
            new_particle[service_idx][role] = random.choice([p for p in self.personnel['musicians'] if p.role == 'Pianis'])
        elif role == 'terompet' or role == 'cadangan_terompet':
            new_particle[service_idx][role] = random.choice([p for p in self.personnel['musicians'] if p.role == 'Terompet'])
        elif role in ['penjaga_pintu', 'pengumpul_persembahan']:
            new_particle[service_idx][role] = random.sample(self.personnel['sintua'],
                                                            3 if role == 'pengumpul_persembahan' else 2)
        elif role == 'cadangan_sintua':
            new_particle[service_idx][role] = random.sample(self.personnel['sintua'], 2)
        elif role == 'operator' or role == 'cadangan_operator':
            new_particle[service_idx][role] = random.choice(self.personnel['operators'])
        else:
            new_particle[service_idx][role] = random.choice(self.personnel['sintua'])
```

Gambar 3.6 Enter Main Loop

5. *Return Best Schedule* adalah tahap akhir dari metode *generate_schedule*, di mana algoritma menghasilkan jadwal kebaktian terbaik yang telah dioptimalkan. Berikut adalah penjelasan prosesnya:

1. Mengambil *Global Best*: Setelah selesai menjalankan seluruh iterasi di *Main Loop*, algoritma memiliki jadwal terbaik secara global yang disimpan dalam variabel *global_best*. Jadwal ini dipilih karena memiliki nilai *fitness* terendah, yang berarti paling sedikit melanggar aturan atau menghasilkan konflik.
2. Format Jadwal: Jadwal terbaik ini diformat agar mudah dibaca dan dipahami oleh pengguna. Metode *_format_schedule* digunakan untuk mengonversi struktur data mentah dari *global_best* menjadi jadwal yang rapi, lengkap

dengan tanggal, waktu kebaktian (pagi dan sore), serta peran-peran utama dan cadangan yang ditugaskan.

```
def _format_schedule(self, schedule, month, year):
    formatted_schedule = []
    current_date = datetime(year, month, 1)

    sunday_count = 0
    while current_date.month == month:
        if current_date.weekday() == 6: # Sunday
            morning_service = schedule[sunday_count * 2]
            afternoon_service = schedule[sunday_count * 2 + 1]

            formatted_schedule.append({
                'date': current_date.strftime('%Y-%m-%d'),
                'morning': self._format_service(morning_service, "08:00-10:30"),
                'afternoon': self._format_service(afternoon_service, "11:00-13:30")
            })
            sunday_count += 1

        current_date += timedelta(days=1)

    return formatted_schedule
```

Gambar 3.7 *Return Best Schedule*

3.2.4 Simulasi dan Pengujian Algoritma

Simulasi dan pengujian algoritma ini menggunakan kode Python yang melibatkan pengelolaan jadwal pelayanan gereja dengan memanfaatkan algoritma optimasi, seperti algoritma genetika (GA) dan partikel swarm optimization (PSO). Dalam hal ini, data yang dimasukkan adalah daftar petugas gereja yang bertugas pada berbagai posisi, seperti pendeta, guru huria, sintua, musisi, operator, dan kelompok paduan suara. Setelah kode dijalankan, maka tampilan output sebagai berikut

```

Date: 2024-12-01

Morning Service: 08:00-10:30

Petugas Utama:
Pengkotbah: Gr. T. Hutabarat
Pemimpin Kebaktian: St. Z. Hutajulu
Pianis: Jhon
Terompet: James
Operator: St. J. Sitorus
Penjaga Pintu: ['St. U. Pasaribu', 'St. D. Marbun']
Singer: ['Lansia', 'Ina Ari Jumat', 'Pelajar']
Pengumpul Persembahan: ['St. V. Panggabean', 'St. Z. Hutajulu', 'St. D. Marbun']
Pembaca Warta: St. G. Manurung
Pemimpin Doa: St. U. Pasaribu

Petugas Cadangan:
Cadangan Pengkotbah: Pdt. L. Simanjuntak
Cadangan Pemimpin Kebaktian: St. E. Silaban
Cadangan Pianis: Yuni
Cadangan Terompet: David
Cadangan Operator: St. E. Silaban
Cadangan Sintua: ['St. Z. Hutajulu', 'St. B. Sitorus']

Afternoon Service: 11:00-13:30

Petugas Utama:
Pengkotbah: Gr. T. Hutabarat
Pemimpin Kebaktian: St. R. Simanungkalit
Pianis: Jhon
Terompet: David
Operator: St. F. Hutapea
Penjaga Pintu: ['St. E. Silaban', 'St. M. Sitompul']
Singer: ['Ama Jeremia', 'Ama Exaudi', 'Ina Ari Rabu']
Pengumpul Persembahan: ['St. A. Sinaga', 'St. H. Sihombing', 'St. H. Pardede']
Pembaca Warta: St. R. Silalahi
Pemimpin Doa: St. T. Lumban Gaol

Petugas Cadangan:
Cadangan Pengkotbah: Pdt. L. Simanjuntak
Cadangan Pemimpin Kebaktian: St. S. Saragih
Cadangan Pianis: Yuni
Cadangan Terompet: Daniel
Cadangan Operator: St. E. Silaban
Cadangan Sintua: ['St. Y. Panjaitan', 'St. R. Simanungkalit']
-----

```

Gambar 3.8 Tampilan Output Kode Program

Output berupa jadwal mingguan untuk penjadwalan 1 bulan, menampilkan tanggal setiap hari Minggu, waktu ibadah pagi dan sore, serta daftar personel yang bertugas untuk setiap peran, termasuk petugas utama dan cadangan. Peran-peran ini mencakup pengkotbah, pemimpin kebaktian, musisi, operator, penjaga pintu, penyanyi, pengumpul persembahan, pembaca warta, dan pemimpin doa.

Jadwal dibuat menggunakan algoritma genetika dan *Particle Swarm Optimization* (PSO) untuk meminimalkan konflik penjadwalan seperti orang yang sama bertugas di beberapa peran sekaligus atau kelompok paduan suara yang sama bertugas di ibadah pagi dan sore. Output yang dihasilkan merupakan jadwal terbaik yang ditemukan oleh

algoritma, bertujuan untuk memberikan jadwal pelayanan gereja yang terstruktur dan memastikan kelancaran ibadah dengan meminimalkan potensi konflik.

3.2.5 Metode Pengujian

Algorithm Testing adalah metode pengujian yang digunakan untuk memastikan bahwa algoritma yang dikembangkan berfungsi sesuai dengan tujuan dan dapat menghasilkan solusi yang benar dan efisien. Dalam konteks penjadwalan petugas ibadah gereja menggunakan *Hybrid PSO-Genetic Algorithm*, pengujian algoritma akan dimulai dengan pengujian fungsional, yang bertujuan untuk memastikan bahwa algoritma dapat menghasilkan jadwal yang sesuai dengan kebutuhan. Pengujian ini melibatkan pengujian terhadap berbagai skenario, seperti pengaturan jadwal dengan jumlah petugas yang bervariasi, untuk memastikan bahwa algoritma dapat menghasilkan solusi yang tepat dengan memenuhi kriteria yang ditentukan, seperti waktu ibadah, kapasitas, dan ketersediaan petugas.

Selanjutnya, pengujian kinerja dilakukan untuk menilai efisiensi algoritma dalam hal waktu eksekusi dan penggunaan memori. Pengujian ini sangat penting mengingat algoritma yang digunakan adalah *hybrid* dari PSO dan *Genetic*, yang berpotensi memiliki beban komputasi yang lebih tinggi. Pengujian kinerja akan mengukur waktu yang dibutuhkan untuk menyelesaikan penjadwalan dengan berbagai ukuran input, serta membandingkan efisiensi algoritma dengan pendekatan lain yang lebih sederhana. Dengan ini, dapat dinilai apakah algoritma tersebut dapat diterapkan pada skenario yang lebih besar atau lebih kompleks tanpa mengorbankan performa.

Selain itu, pengujian perbandingan dilakukan untuk membandingkan hasil yang diperoleh dari algoritma *hybrid PSO-Genetic* dengan algoritma lain yang mungkin relevan, seperti algoritma PSO atau *Genetic* secara terpisah. Tujuan dari pengujian ini adalah untuk melihat apakah penggabungan kedua algoritma tersebut memberikan peningkatan performa yang signifikan dibandingkan dengan masing-masing algoritma individu. Dengan melakukan pengujian perbandingan, dapat diperoleh wawasan mengenai keunggulan dan kekurangan metode *hybrid* tersebut dalam menghasilkan solusi penjadwalan yang optimal dan efisien dalam konteks penjadwalan petugas ibadah gereja.

DAFTAR PUSTAKA

- [1] R. J. Ondang and S. R. Kalangi, “Pemanfaatan Media Digital dalam Pelayanan Gerejawi,” *TELEIOS J. Teol. dan Pendidik. Agama Kristen*, vol. 3, no. 1, pp. 62–76, 2023, doi: 10.53674/teleios.v3i1.79.
- [2] A. U. Azmi, R. Hidayat, and M. Z. Arif, “Perbandingan Algoritma Particle Swarm Optimization (Pso) Dan Algoritma Glowworm Swarm Optimization (Gso) Dalam Penyelesaian Sistem Persamaan Non Linier,” *Maj. Ilm. Mat. dan Stat.*, vol. 19, no. 1, p. 29, 2019, doi: 10.19184/mims.v19i1.17263.
- [3] D. A. Suprayogi and W. F. Mahmudy, “Penerapan Algoritma Genetika Traveling Salesman Problem with Time Window: Studi Kasus Rute Antar Jemput Laundry,” *J. Buana Inform.*, vol. 6, no. 2, pp. 121–130, 2015, doi: 10.24002/jbi.v6i2.407.
- [4] Y. S. Nugraha, U. Darusalam, and A. Iskandar, “Implementasi Algoritma Genetika pada Perancangan Aplikasi Penjadwalan Instalasi Antivirus Berbasis Website menggunakan Metode Waterfall,” *J. JTIK (Jurnal Teknol. Inf. dan Komunikasi)*, vol. 6, no. 1, pp. 125–137, 2022, doi: 10.35870/jtik.v6i1.417.
- [5] R. Hartono and A. Zein, “Penerapan Algoritma Genetika Dan Jaringan Syaraf Tiruan Dalam Penjadwalan Mata Kuliah,” *6 |Jurnal Ilmu Komput. JIK*, vol. VI, no. 03, pp. 6–7, 2023.
- [6] M. Muhardeny, M. H. Irfani, and J. Alie, “Penjadwalan Mata Pelajaran Menggunakan Algoritma Particle Swarm Optimization (PSO) Pada SMPIT Mufidatul Ilmi,” *J. Softw. Eng. Comput. Intell.*, vol. 1, no. 1, pp. 51–63, 2023, doi: 10.36982/jseci.v1i1.3047.
- [7] H. Kuswanto, P. S. Informatika, F. T. Informasi, J. J. Raya, and K. Makasar, “SISTEM INFORMASI PENJADWALAN KUNJUNGAN TEKNISI,” vol. 8, no. 4, pp. 431–437, 2024.
- [8] N. Yumarnis and S. Dewi Andriana, “the Use of Genetic Algorithm in Determining Employee Schedules At Pln Ulp Office Selatan,” *Int. J. Data Sci. Vis.*, vol. 2, no. 1, p. 37, 2023.
- [9] H. Syahputra, B. Sugito, and Z. Sitorus, “Application Of Particle Swarm Optimization Algorithm In Information Systems Class Scheduling SMK Panca Budi 1 Medan,” *Int. J. Comput. Sci. Math. Eng.*, vol. 2, no. 2, 2023.

- [10] W. A. Puspaningrum, A. Djunaidy, and R. A. Vinarti, "Penjadwalan Mata Kuliah Menggunakan Algoritma Genetika di Jurusan Sistem Informasi ITS," *J. Tek. Pomits*, vol. 2, no. 1, pp. 127–131, 2013.
- [11] T. Handayani, D. H. Fudholi, and S. Rani, "Kajian Algoritma Optimasi Penjadwalan Mata Kuliah," *Petir*, vol. 13, no. 2, pp. 212–222, 2020, doi: 10.33322/petir.v13i2.1027.
- [12] E. Z. Hooker, "Field research.," *SCI Nurs.*, vol. 7, no. 3, pp. 65–67, 1990.
- [13] N. I. Lesmana, "Penjadwalan Produksi Untuk Meminimalkan Waktu Produksi Dengan Menggunakan Metode Branch And Bound," *J. Tek. Ind.*, vol. 17, no. 1, p. 42, 2017, doi: 10.22219/jtiumm.vol17.no1.42-50.
- [14] B. Santosa, "Budi Santosa dan Paul Willy," 2011.
- [15] I. L. Putra, "Implementasi Algoritma Particle Swarm Optimization(Pso) Dan K-Nearest Neighbor(K-Nn) Dalam Memprediksi Keberhasilan Anak Smk Mendapatkan Kerja," *Technol. J. Ilm.*, vol. 13, no. 4, p. 339, 2022, doi: 10.31602/tji.v13i4.8167.
- [16] K. F. Irnanda, A. P. Windarto, and I. S. Damanik, "Optimasi Particle Swarm Optimization Pada Peningkatan Prediksi dengan Metode Backpropagation Menggunakan Software RapidMiner," *JURIKOM (Jurnal Ris. Komputer)*, vol. 9, no. 1, p. 122, 2022, doi: 10.30865/jurikom.v9i1.3836.
- [17] T. Alam, S. Qamar, A. Dixit, and M. Benaida, "Genetic algorithm: Reviews, implementations and applications," *Int. J. Eng. Pedagog.*, vol. 10, no. 6, pp. 57–77, 2021, doi: 10.3991/IJEP.V10I6.14567.
- [18] W. F. Mahmudy, R. M. Marian, and L. H. S. Luong, "Hybrid Genetic Algorithms for Part Type Selection and Machine Loading Problems with Alternative Production Plans in Flexible Manufacturing System," *ECTI Trans. Comput. Inf. Technol.*, vol. 8, no. 1, pp. 80–93, 1970, doi: 10.37936/ecti-cit.201481.54390.
- [19] R. Christian and D. S. Donoriyanto, "Penerapan Algoritma Genetika Dalam Penjadwalan Mata Kuliah Program Studi Teknik Industri Upn "Veteran" Jawa Timur," *Tekmapro J. Ind. Eng. Manag.*, vol. 16, no. 2, pp. 1–12, 2021, doi: 10.33005/tekmapro.v16i2.157.

LAMPIRAN

LAMPIRAN 1 Kode Program

[KodeProgram](#)