

TESIS

**OPTIMASI PENJADWALAN MATA KULIAH DENGAN KOMBINASI
CONSTRAINT SATISFACTION PROBLEM (CSP) DAN ALGORITMA
GENETIKA**

(Studi Kasus: Universitas Duta Bangsa Surakarta)



Disusun oleh:

Nama : Triyono
NIM : 22.55.2289
Konsentrasi : Digital Transformation Intelligence

**PROGRAM STUDI S2 TEKNIK INFORMATIKA
PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA**

2025

TESIS

**OPTIMASI PENJADWALAN MATA KULIAH DENGAN KOMBINASI
CONSTRAINT SATISFACTION PROBLEM (CSP) DAN ALGORITMA
GENETIKA**

(Studi Kasus: Universitas Duta Bangsa Surakarta)

**OPTIMIZATION OF COURSE SCHEDULING USING A COMBINATION
OF CONSTRAINT SATISFACTION PROBLEM (CSP) AND GENETIC
ALGORITHM**

(Case Study: Duta Bangsa Surakarta University)

Diajukan untuk memenuhi salah satu syarat memperoleh derajat Magister



Disusun oleh:

Nama : Triyono
NIM : 22.55.2289
Konsentrasi : Digital Transformation Intelligence

**PROGRAM STUDI S2 TEKNIK INFORMATIKA
PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA**

2025

HALAMAN PENGESAHAN

OPTIMASI PENJADWALAN MATA KULIAH DENGAN KOMBINASI
CONSTRAINT SATISFACTION PROBLEM (CSP) DAN ALGORITMA
GENETIKA

*OPTIMIZATION OF COURSE SCHEDULING USING A COMBINATION OF
CONSTRAINT SATISFACTION PROBLEM (CSP) AND GENETIC ALGORITHM*

yang disusun dan diajukan oleh

Triyono

22.55.2289

Telah dipertahankan di depan Dewan Penguji
pada tanggal 1 Desember 2025

Susunan Dewan Penguji

Nama Penguji

Tanda Tangan

Hanafi, S.Kom., M.Eng., Ph.D.
NIK. 190302024

Hanif Al Fatta, S.Kom., M.Kom., Ph.D.
NIK. 190302096

Prof. Dr. Kusriani, S.Kom., M.Kom.
NIK. 190302106



Tesis ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Magister Komputer
Tanggal 1 Desember 2025

DEKAN FAKULTAS ILMU KOMPUTER



Prof. Dr. Kusriani, M.Kom.
NIK. 190302106

HALAMAN PERSETUJUAN

**OPTIMASI PENJADWALAN MATA KULIAH DENGAN KOMBINASI
CONSTRAINT SATISFACTION PROBLEM (CSP) DAN ALGORITMA
GENETIKA**

*OPTIMIZATION OF COURSE SCHEDULING USING A COMBINATION OF
CONSTRAINT SATISFACTION PROBLEM (CSP) AND GENETIC ALGORITHM*

yang disusun dan diajukan oleh

Triyono
22.55.2289

telah disetujui oleh Dosen Pembimbing Tesis
pada tanggal 1 Desember

Dosen Pembimbing,



Prof. Dr. Kusriani, M.Kom

NIK. 190302106

HALAMAN PERNYATAAN KEASLIAN TESIS

Yang bertandatangan di bawah ini,

Nama mahasiswa : Triyono
NIM : 22.55.2289
Konsentrasi : Digital Transformation Intelligence

Menyatakan bahwa Tesis dengan judul berikut:

**OPTIMASI PENJADWALAN MATA KULIAH DENGAN KOMBINASI
CONSTRAINT SATISFACTION PROBLEM (CSP) DAN ALGORITMA
GENETIKA**

Dosen Pembimbing Utama : Prof. Dr. Kusriani M.Kom

1. Karya tulis ini adalah benar-benar ASLI dan BELUM PERNAH diajukan untuk mendapatkan gelar akademik, baik di Universitas AMIKOM Yogyakarta maupun di Perguruan Tinggi lainnya
2. Karya tulis ini merupakan gagasan, rumusan dan penelitian SAYA sendiri, tanpa bantuan pihak lain kecuali arahan dari Tim Dosen Pembimbing
3. Dalam karya tulis ini tidak terdapat karya atau pendapat orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan disebutkan dalam Daftar Pustaka pada karya tulis ini
4. Perangkat lunak yang digunakan dalam penelitian ini sepenuhnya menjadi tanggung jawab SAYA, bukan tanggung jawab Universitas AMIKOM Yogyakarta
5. Pernyataan ini SAYA buat dengan sesungguhnya, apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka SAYA bersedia menerima SANKSI AKADEMIK dengan pencabutan gelar yang sudah diperoleh, serta sanksi lainnya sesuai dengan norma yang berlaku di Perguruan Tinggi

Yogyakarta, 1 Desember 2025

Yang Menyatakan,




Triyono

HALAMAN PERSEMBAHAN

Puji syukur yang tak terhingga penulis ucapkan kepada Allah SWT, Tuhan penguasa alam yang telah meridhoi dan mengabulkan segala do'a sehingga penulis dapat menyelesaikan tesis berjudul "OPTIMASI PENJADWALAN MATA KULIAH DENGAN KOMBINASI CONSTRAINT SATISFACTION PROBLEM (CSP) DAN ALGORITMA GENETIKA" sesuai dengan yang diharapkan oleh penulis. Alhamdulillah, dengan rasa bangga dan bahagia penulis persembahkan tesis ini kepada:

1. Kedua orang tua dan keluarga besar saya, yang selalu mendidik saya, mendukung setiap langkah baik yang saya ambil, selalu sabar menghadapi kelakuan saya dan mengingatkan saya ketika melakukan hal yang salah.
2. Ibu Prof. Dr. Kusriani, M. Kom, selaku dosen pembimbing utama, terima kasih sudah membimbing dan membantu saya dalam pengerjaan tesis. Terima kasih atas segala kesabaran dan ilmu yang diberikan selama ini.
3. Teman-teman Fakultas Ilmu Komputer Universitas Duta Bangsa Surakarta terima kasih atas segala dukungan yang telah di berikan.

KATA PENGANTAR

Assalamu'alaikum Wr. Wb.

Puji dan syukur penulis persembahkan untuk Allah SWT yang telah memberikan rahmat, hidayah, dan kekuatan sehingga penulis dapat menyelesaikan tesis ini sesuai dengan waktu yang diharapkan. Tidak lupa sholawat dan salam penulis haturkan pada junjungan umat yaitu Nabi Muhammad SAW yang telah menuntun kita pada jalan kebaikan.

Tesis ini disusun dalam rangka memenuhi salah satu persyaratan kelulusan jenjang Program Magister Universitas AMIKOM Yogyakarta. Dengan selesainya tesis ini, maka penulis tidak lupa mengucapkan terima kasih kepada:

1. Allah SWT karena atas izin dan karunia-Nya maka tesis ini dapat dibuat dan selesai pada waktunya. Puji syukur yang tak terhingga padaNya sebagai penguasa alam yang mengabulkan segala do'a.
2. Ibu dan Bapak saya, serta seluruh keluarga besar yang selalu menyelipkan doa di setiap sujudnya agar saya dapat menjadi pribadi yang lebih baik dan terus maju.
3. Bapak Prof. Dr. M. Suyanto, M.M., selaku Rektor Universitas AMIKOM Yogyakarta.
4. Ibu Prof. Dr. Kusriani, M. Kom, selaku Dekan Fakultas Ilmu Komputer Universitas AMIKOM Yogyakarta dan sekaligus sebagai pembimbing tesis
5. Bapak Hanafi, S.Kom., M.Eng., Ph.D., Bapak Hanif Al Fatta, M.Kom., Ph.D, Tonny Hidayat, S.Kom., M.Kom., Ph.D, Dr. Ferry Wahyu Wibowo, S.Si., M.Cs. sebagai dosen penguji mulai dari tahapan seminar proposal

hingga ujian tesis. Serta semua dosen Magister Teknik Informatika Universitas Amikom Yogyakarta, terima kasih atas semua jasa Bapak dan Ibu Dosen.

6. Segenap Dosen dan Civitas Akademika Universitas AMIKOM Yogyakarta yang telah memberikan banyak ilmu dan pengalaman kepada penulis selama menjalani perkuliahan.
7. Seluruh pihak yang tidak dapat disebutkan satu persatu yang telah banyak membantu sehingga tesis ini dapat terselesaikan.

Penulis tentunya menyadari bahwa pembuatan tesis ini masih banyak kekurangan dan kelemahannya. Oleh karena itu penulis berharap kepada semua pihak agar dapat menyampaikan kritik dan saran yang membangun untuk menambah kesempurnaan tesis ini. Namun penulis tetap berharap tesis ini akan bermanfaat bagi semua pihak yang membacanya.

Wassalamu'alaikum Wr, Wb.

Surakarta, 15 Desember 2025

Penulis

DAFTAR ISI

HALAMAN JUDUL	ii
HALAMAN PENGESAHAN	iii
HALAMAN PERSETUJUAN	iv
HALAMAN PERNYATAAN KEASLIAN TESIS	v
HALAMAN PERSEMBAHAN	vi
KATA PENGANTAR	vii
DAFTAR ISI	ix
DAFTAR TABEL	xiii
DAFTAR GAMBAR	xiv
INTISARI	xvi
<i>ABSTRACT</i>	xvii
BAB I PENDAHULUAN	1
1.1. Latar Belakang Masalah	1
1.2. Rumusan Masalah	4
1.3. Batasan Masalah	5
1.4. Tujuan Penelitian	6
1.5. Manfaat Penelitian	6
BAB II TINJAUAN PUSTAKA	12
2.1. Tinjauan Pustaka	12
1. Pendekatan CSP Murni	12
2. Genetic Algorithm (GA) Murni	13

3. Hybrid GA dengan Local Search / Repair.....	13
4. Hybrid GA + CSP (Penelitian Terdahulu yang Paling Relevan)	14
5. Metaheuristik Multi-Objektif	14
2.2. Keaslian Penelitian	16
2.3. Landasan Teori	19
BAB III METODE PENELITIAN	25
3.1. Jenis, Sifat, dan Pendekatan Penelitian	25
3.2. Metode Pengumpulan Data	25
3.3. Pemodelan CSP (Feasibility Engine)	26
3.3.1 Variabel Model	28
3.3.2 Domain	29
3.3.3 Hard Constraint	30
3.3.4 Soft Constraint	31
3.3.5 Relasi Antar Entitas	32
3.3.6 Solver CP-SAT	35
3.4. Desain Genetic Algorithm (Optimization Engine)	36
3.4.1 Representasi Kromosom	38
3.4.2 Inisialisasi Populasi	39
3.4.3 Operator Evolusi GA	40
3.4.4 Elitisme	41
3.4.5 Evaluasi Fitness (Menggunakan CSP)	42

3.4.6 Kondisi Konvergensi	43
3.5. Fungsi Objektif Multi-Kriteria	44
3.6. Arsitektur Dual-Phase CSP-GA.....	45
3.7. Dampak CSP Terhadap Proses Pencarian GA	48
3.8. Perbedaan Metode Penelitian dengan Studi Terdahulu	49
3.9. Protokol Eksperimen	50
3.10. Implementasi Sistem.....	52
3.11. Ringkasan Kontribusi Metode	53
BAB IV HASIL PENELITIAN DAN PEMBAHASAN.....	55
4.1. Deskripsi Dataset dan Konfigurasi Eksperimen.....	55
4.1.1 Dataset yang Digunakan.....	55
4.1.2 Parameter Algoritma Genetika (AG).....	58
4.2. Proses Algoritma	61
4.2.1 Alur Umum.....	61
4.2.2 Tahap CSP sebagai feasibility engine	65
4.2.3 Tahap GA sebagai optimization engine	67
4.2.4 Objective function dan definisi fitness	70
4.2.5 Integrasi CSP dan GA	72
4.2.6 Implementasi dan orkestrasi	75
4.2.7 Artefak keluaran	77
4.3. Hasil Eksperimen.....	78
4.3.1 Konvergensi Nilai Fitness	78

4.3.2 Kualitas Solusi Terbaik	83
4.3.3 Perbandingan Variasi Parameter AG.....	87
4.4. Analisis Hard Constraints dan Soft Constraints	91
4.4.1 Pemenuhan Hard Constraints	92
4.4.2 Tingkat Pencapaian Soft Constraint.....	94
4.5. Diskusi Hasil.....	98
4.6 Pembahasan Tujuan, Keterbatasan, dan Rekomendasi	104
4.6.1 Pembahasan Tujuan Penelitian	104
4.6.2 Keterbatasan Penelitian	105
4.6.3 Rekomendasi Pengembangan.....	106
BAB V PENUTUP	108
5.1. Kesimpulan	108
5.2. Saran	109
DAFTAR PUSTAKA	112
LAMPIRAN	114

DAFTAR TABEL

Tabel 1. Matriks literatur review	16
Tabel 2 Matriks posisi penelitian.....	17
Tabel 3 Rangkuman Set Eksperimen.....	51
Tabel 4 Karakteristik Dataset	56
Tabel 5 Pembagian slot waktu.....	57
Tabel 6 Ringkasan Variasi Skenario GA dan Nilai Objektif.....	58
Tabel 7 Indikator Tambahan Soft Constraint	60
Tabel 8 Hasil akhir skenario B	79
Tabel 9 Hasil akhir skenario C	80
Tabel 10 Hasil akhir skenario B	82
Tabel 11 Nilai Penalti Final (Objective Rekonstruksi).....	83
Tabel 12 hasil solusi terbaik B3.....	84
Tabel 13 Ringkasan Penalti Soft Constraints (Objective Rekonstruksi).....	95

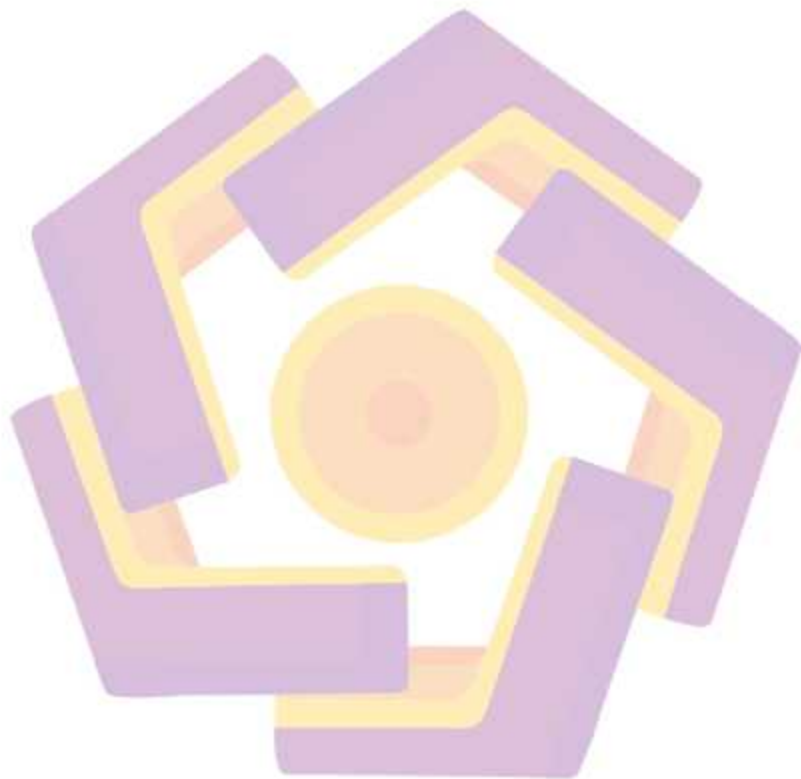
DAFTAR GAMBAR

Gambar 1 Flowchart of genetic algorithm.....	21
Gambar 2 Diagram Arsitektur CSP	27
Gambar 3 Relasi entitas CSP.....	34
Gambar 4 Diagram Arsitektur GA.....	37
Gambar 5 Diagram Arsitektur Hybrid CSP GA.....	46
Gambar 6 Alur Eksekusi Metode Hybrid CSP-GA.....	64
Gambar 7 Bagaimana CSP dieksekusi pada eksperimen.....	65
Gambar 8 Alur Eksekusi GA sebagai Optimization Engine.....	69
Gambar 9 Mekanisme Integrasi CSP dan GA.....	74
Gambar 10 Perbandingan Fitness Skenario B (Light-Medium-Heavy).....	79
Gambar 11 Perbandingan Fitness Skenario C (t14-t16-t18).....	81
Gambar 12 Perbandingan Fitness Skenario D (Seed 11-42-77).....	82
Gambar 13 Heatmap Jadwal Terbaik (B3 - GA Heavy).....	85
Gambar 14 Perbandingan Penalti Akhir Antar Skenario.....	86
Gambar 15 screenshot jadwal pada hasil B3.....	87
Gambar 16 Perbandingan Fitness GA Terbaik pada Skenario B (Light-Medium-Heavy).....	88
Gambar 17 Perbandingan Penalti Akhir Skenario C (t14-t16-t18).....	89
Gambar 18 Stabilitas Penalti Antar Seed GA.....	90
Gambar 19 Heatmap Distribusi Sesi — Skenario B3 (Solusi Terbaik).....	93
Gambar 20 Perbandingan Penalti Soft Constraints Antar Skenario.....	96

Gambar 21 Heatmap Distribusi Slot — Skenario B397

Gambar 22 Heatmap CSP-only (Skenario A1)..... **Error! Bookmark not defined.**

Gambar 23 Heatmap Hybrid CSP-GA (Skenario B3)**Error! Bookmark not defined.**



INTISARI

Penelitian ini bertujuan menyusun jadwal kuliah yang feasible dan berkualitas untuk 17 cohort, 116 sesi, 41 dosen, dan 19 ruang (14 teori, 5 laboratorium). Variabel keputusan dibatasi pada penugasan tiap sesi ke slot waktu dan ruang. Hard constraint meliputi tidak ada bentrok dosen/cohort/ruang, kesesuaian kapasitas dan tipe ruang, serta kepatuhan terhadap slot terblokir; sedangkan soft constraint meminimalkan penempatan sesi pada slot S1/S8 dan memastikan hari bebas cohort. Metode analisis menggabungkan CSP menggunakan Google OR-Tools CP-SAT untuk penegakan batasan dan perhitungan penalti, serta Genetic Algorithm untuk eksplorasi; solusi GA terbaik dijadikan hint untuk pemolesan akhir oleh CP-SAT.

Eksperimen pada dataset membandingkan CSP murni, GA murni, dan hibrida. CP-SAT murni mencapai objective 0 dalam ≈ 5 detik pada bobot ($w_{s1}=1$, $w_{s8}=1$, $w_{free_day}=12$), seluruh hard constraint terpenuhi dan setiap cohort memperoleh hari bebas. GA murni menghasilkan jadwal feasible namun penalti soft lebih tinggi (objective 43) dan beban harian kurang seimbang. Pendekatan hibrida menurunkan penalti dari solusi GA (objective 40) serta memperbaiki keseimbangan beban tanpa pelanggaran hard constraint. Pada bobot institusi ($1/1/8$), peningkatan intensitas GA menurunkan objective dari 52 (populasi 8, generasi 6) menjadi 42 (populasi 12, generasi 10).

Kesimpulannya, CP-SAT efektif menjamin kelayakan dan refinement cepat, sedangkan GA bermanfaat untuk eksplorasi preferensi ketika konfigurasi bobot membuat optimasi lebih sulit. Kombinasi GA-CP-SAT menghasilkan jadwal yang stabil, bebas konflik, dan lebih sesuai prioritas institusi.

Kata kunci: Penjadwalan Perkuliahan, CP-SAT, Constraint Programming, Genetic Algorithm, Hybrid Metaheuristic

ABSTRACT

This study aims to generate a feasible and high-quality course timetable for 17 cohorts, 116 sessions, 41 lecturers, and 19 rooms (14 lecture rooms and 5 laboratories). The decision variables are limited to assigning each session to a time slot and a room. Hard constraints include no conflicts among lecturers/cohorts/rooms, appropriate room capacity and type, and compliance with blocked time slots; soft constraints minimize sessions placed in S1/S8 slots and ensure a day off for each cohort. The proposed approach combines a Constraint Satisfaction Problem (CSP) using Google OR-Tools CP-SAT for constraint enforcement and penalty evaluation, with a Genetic Algorithm (GA) for exploration; the best GA solution is used as a hint for final refinement by CP-SAT.

Experiments on the dataset compare pure CSP, pure GA, and a hybrid approach. Pure CP-SAT achieves an objective value of 0 in approximately 5 seconds with weights ($w_{s1}=1$, $w_{s8}=1$, $w_{free\ day}=12$), satisfying all hard constraints and providing each cohort with a day off. Pure GA produces feasible schedules but with higher soft-constraint penalties (objective 43) and less balanced daily loads. The hybrid approach reduces the GA penalty (objective 40) and improves load balance while maintaining zero hard-constraint violations. Under institutional weights (1/1/8), increasing GA intensity reduces the objective from 52 (population 8, generation 6) to 42 (population 12, generation 10).

In conclusion, CP-SAT is effective for guaranteeing feasibility and fast refinement, while GA is beneficial for exploring preferences when weight configurations make optimization more difficult. The GA-CP-SAT combination yields stable, conflict-free schedules that better align with institutional priorities.

Keyword: Course Timetabling, CP-SAT, Constraint Programming, Genetic Algorithm, Hybrid Metaheuristic.

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Penjadwalan mata kuliah di perguruan tinggi merupakan permasalahan kombinatorial kompleks yang termasuk dalam kelas NP-Hard dan memiliki dampak signifikan terhadap efektivitas proses pembelajaran. Masalah ini melibatkan alokasi ratusan mata kuliah ke dalam slot waktu dan ruang terbatas, sembari memenuhi berbagai kendala teknis dan preferensi pengguna yang sering kali saling bertentangan (Abdipoor, 2023). Kompleksitas ini semakin meningkat seiring dengan berkembangnya skala institusi pendidikan dan diversifikasi program akademik.

Penelitian terkini menunjukkan bahwa kualitas jadwal akademik memiliki pengaruh langsung terhadap kepuasan mahasiswa dan efisiensi operasional institusi. (Al-Milli, 2024) menekankan pentingnya optimasi multi-objektif dalam penjadwalan, dimana solusi tidak hanya harus memenuhi kendala dasar seperti kapasitas ruang dan ketersediaan dosen, tetapi juga mengoptimalkan preferensi seperti minimisasi jam kosong, distribusi beban kerja yang seimbang, dan penyediaan hari bebas bagi mahasiswa. Studi mereka menunjukkan bahwa pendekatan multi-objektif dengan algoritma NSGA-II dapat meningkatkan kualitas jadwal secara signifikan dibandingkan metode konvensional.

Dari perspektif metodologi, berbagai pendekatan telah dikembangkan untuk mengatasi kompleksitas penjadwalan akademik. (Abdipoor, 2023) dalam kajian

komprehensif mereka mengidentifikasi bahwa metaheuristik seperti Genetic Algorithm (GA), Simulated Annealing, dan Tabu Search telah banyak digunakan untuk masalah ini. Namun, mereka juga menggarisbawahi kelemahan fundamental dari pendekatan-pendekatan tersebut: kesulitan dalam menjamin pemenuhan hard constraints secara konsisten tanpa mekanisme constraint handling yang sistematis. Hal ini sering mengakibatkan solusi yang secara teoritis optimal namun praktis tidak dapat diimplementasikan karena melanggar kendala dasar.

Upaya untuk mengatasi keterbatasan ini telah mendorong pengembangan pendekatan hybrid yang menggabungkan kekuatan berbagai teknik optimasi. (Rezacipahan, 2021) mengusulkan kombinasi Improved Parallel Genetic Algorithm (IPGA) dengan Local Search (LS) untuk meningkatkan kualitas solusi dan mengurangi pelanggaran kendala. Hasil penelitian mereka menunjukkan bahwa pendekatan hybrid dapat mencapai performa yang lebih baik dibandingkan GA murni, terutama dalam hal konvergensi dan kualitas solusi akhir. Namun, pendekatan mereka masih menggunakan mekanisme penalty function untuk menangani kendala, yang tidak memberikan jaminan formal terhadap feasibility solusi.

Perkembangan terbaru dalam bidini menunjukkan potensi besar dari integrasi Constraint Satisfaction Problem (CSP) dengan algoritma evolusioner. (Putra, 2024) melakukan analisis terhadap kombinasi GA dan metode CSP untuk distribusi penjadwalan dosen, menunjukkan bahwa pendekatan hybrid dapat mencapai zero conflicts dengan utilisasi sumber daya yang optimal. Penelitian mereka memvalidasi konsep bahwa CSP dapat berfungsi sebagai mesin validasi

yang sistematis, sementara GA berperan sebagai optimizer untuk soft constraints. Namun, implementasi mereka masih terbatas pada skala kecil dan belum memanfaatkan solver CSP modern seperti CP-SAT yang memiliki kemampuan constraint propagation dan parallel search yang canggih.

Meskipun berbagai pendekatan telah dikembangkan, masih terdapat gap signifikan dalam literatur. Pertama, mayoritas penelitian fokus pada optimasi atau feasibility secara terpisah, tanpa mengintegrasikan keduanya dalam framework yang sistematis. Kedua, implementasi hybrid CSP-GA yang memanfaatkan solver modern masih jarang ditemukan, terutama yang divalidasi pada data riil institusi dengan kompleksitas tinggi. Ketiga, evaluasi komprehensif terhadap trade-off antara computational efficiency dan solution quality dalam konteks hybrid approach belum banyak dilakukan.

Dalam konteks Universitas Duta Bangsa Surakarta, tantangan penjadwalan mencakup 17 cohort dengan total 554 mahasiswa, 116 sesi mata kuliah, 41 dosen, dan 19 ruang dengan berbagai tipe dan kapasitas. Data ini mencerminkan realitas operasional perguruan tinggi menengah di Indonesia, dimana keterbatasan sumber daya dan kompleksitas kendala memerlukan solusi yang tidak hanya optimal secara teoritis, tetapi juga praktis dan dapat diimplementasikan. Oleh karena itu, pengembangan framework hybrid CSP+GA yang dapat menjamin feasibility sekaligus mengoptimalkan kualitas jadwal menjadi sangat relevan dan diperlukan.

Perkembangan riset penjadwalan akademik menunjukkan munculnya berbagai algoritma optimasi terkini seperti NSGA-II, Simulated Annealing hybrid, Particle Swarm Optimization, memetic algorithms, serta Large Neighborhood

Search yang digunakan secara luas dalam domain scheduling. Perkembangan ini mendorong rasa ingin tahu (curiosity) ilmiah untuk memahami bagaimana masing-masing pendekatan bekerja dan batas-batas kemampuan mereka dalam menangani kendala kompleks pada institusi pendidikan.

Dalam konteks tersebut, penelitian ini memilih Genetic Algorithm (GA) bukan semata karena popularitasnya, tetapi karena karakteristiknya yang fleksibel untuk optimasi multi-kriteria, mudah digabungkan dengan mekanisme CSP sebagai penjaga feasibility, serta memiliki rekam jejak kuat dalam menangani ruang solusi besar. Namun demikian, peneliti juga mengakui bahwa algoritma lain berpotensi memberikan hasil yang lebih baik. Oleh sebab itu, penelitian ini diposisikan sebagai baseline hybrid CSP-GA yang dapat menjadi titik awal untuk eksplorasi dan pengujian berbagai algoritma optimasi lain pada studi lanjutan.

1.2. Rumusan Masalah

Berdasarkan latar belakang masalah yang telah diuraikan, berikut adalah rumusan masalah yang dirumuskan secara tegas dan jelas dalam bentuk poin-poin terinci untuk penelitian:

1. Berapa tingkat akurasi dan efisiensi yang dapat dicapai melalui penerapan algoritma genetik dan CSP dalam penjadwalan mata kuliah?
2. Bagaimana algoritma genetik dan CSP dapat digunakan untuk mengoptimalkan penjadwalan mata kuliah dengan mempertimbangkan berbagai kendala teknis dan preferensi pengguna?

1.3. Batasan Masalah

Agar ruang lingkup penelitian fokus dan terukur, batasan masalah ditetapkan sebagai berikut:

1. Data studi kasus

Program Studi Teknik Informatika Universitas Duta Bangsa Surakarta—17 cohort (total 554 mahasiswa), 116 sesi mata kuliah, 41 dosen, 19 ruang.

2. Hard constraints yang diwajibkan:

a. No Overlap

Dosen, cohort, dan ruang tidak boleh memiliki dua sesi bersamaan.

b. Kapasitas Ruang

Kapasitas ruang harus mencukupi jumlah mahasiswa setiap sesi.

c. Tipe Ruang

Lab hanya untuk sesi LAB; teori untuk sesi THEORY.

d. Ketersediaan Dosen

Hari libur dan batas slot akhir dosen dipatuhi.

e. Slot Terblokir

Slot tertentu tidak boleh digunakan (konstanta BLOCKS).

3. Soft constraints yang dioptimasi

penalti sesi pertama (w_{s1}), sesi terakhir (w_{s8}), cohort tanpa free-day (w_{free_day}), serta deviasi dari petunjuk GA (dev_hint). Kompakness jadwal tidak dioptimasi, hanya dianalisis.

4. Metode pembandingan

CSP-only (tanpa GA) dan hybrid GA+CSP. Tidak ada perbandingan dengan solusi manual atau algoritma pihak ketiga.

1.4. Tujuan Penelitian

Berdasarkan judul dan latar belakang masalah yang telah diuraikan, tujuan penelitian ini secara spesifik adalah:

1. Mengembangkan framework hybrid CSP+GA yang terintegrasi untuk penjadwalan mata kuliah.
2. Merancang fungsi objektif multi-kriteria berbobot sesuai kebijakan institusi.
3. Mengevaluasi akurasi (feasibility), efisiensi (waktu eksekusi), dan sensitivitas parameter GA melalui eksperimen terkontrol

1.5. Manfaat Penelitian

Penelitian ini memberikan kontribusi signifikan baik dari sisi pengembangan keilmuan maupun aplikasi praktis dalam domain penjadwalan akademik. Manfaat yang dihasilkan dapat diklasifikasikan ke dalam beberapa dimensi yang saling berkaitan dan mendukung kemajuan bidang optimasi dan sistem informasi akademik:

1. Manfaat Akademik
 - a. Kontribusi Metodologis

Penelitian ini memperkaya literatur ilmiah dengan memperkenalkan framework hybrid CSP+GA yang sistematis dan terdokumentasi dengan baik. Berbeda dari pendekatan hybrid sebelumnya yang umumnya menggabungkan GA dengan teknik lokal search atau penalty-based

methods, penelitian ini menggunakan CP-SAT solver sebagai core constraint handler yang memberikan jaminan formal terhadap feasibility. Kontribusi ini mengisi gap metodologis yang signifikan dalam literatur optimasi kombinatorial, khususnya dalam integrasi formal antara exact methods (CSP) dan metaheuristics (GA).

b. Validasi Empiris Komprehensif

Penelitian ini menyediakan bukti empiris yang kuat melalui eksperimen terkontrol pada data riil multi-entity yang representatif terhadap kompleksitas operasional perguruan tinggi. Dataset yang digunakan mencakup 17 cohort, 116 sesi mata kuliah, 41 dosen, dan 19 ruang dengan berbagai atribut dan kendala realistis seperti hari libur dosen, kapasitas ruang, dan tipe mata kuliah. Validasi pada data riil ini memberikan kredibilitas tinggi terhadap hasil penelitian dan memfasilitasi replikasi serta pengembangan lebih lanjut oleh peneliti lain.

c. Kerangka Evaluasi Multi-Dimensi

Penelitian ini mengembangkan kerangka evaluasi komprehensif yang mencakup multiple metrics: feasibility rate, optimization quality, computational efficiency, parameter sensitivity, dan seed robustness. Kerangka evaluasi ini dapat diadopsi oleh penelitian-penelitian selanjutnya sebagai standar untuk mengukur performa algoritma hybrid dalam domain scheduling problems.

d. Paradigma Dual-Phase Optimization

Konsep dual-phase yang diperkenalkan—dimana CSP bertindak sebagai feasibility engine dan GA sebagai optimization engine—memberikan paradigma baru dalam mendesain solver untuk constrained optimization problems. Paradigma ini dapat diterapkan pada berbagai domain selain penjadwalan akademik, seperti job shop scheduling, vehicle routing, dan resource allocation problems.

2. Manfaat Praktis

a. Otomatisasi Proses Penjadwalan

Framework yang dikembangkan menyediakan solusi otomatis end-to-end untuk penjadwalan mata kuliah, menggantikan proses manual yang selama ini memakan waktu berminggu-minggu. Sistem dapat menghasilkan jadwal valid dalam hitungan detik (CSP phase) dan jadwal optimal dalam hitungan menit (GA phase), secara dramatis mengurangi beban kerja staf akademik dan kemungkinan human error dalam proses penjadwalan.

b. Fleksibilitas Konfigurasi Preferensi

Sistem mengimplementasikan fungsi objektif berbobot yang memungkinkan institusi untuk menyesuaikan prioritas penjadwalan sesuai kebijakan dan kebutuhan spesifik. Misalnya, institusi dapat memberikan bobot tinggi pada pengosongan hari Jumat ($w_{free_day}=8$) atau

meminimalkan sesi pagi (w_{s1}) sesuai preferensi mahasiswa dan dosen. Fleksibilitas ini memberikan kontrol yang granular kepada pengguna terhadap trade-off yang diinginkan dalam penjadwalan.

c. Jaminan Kualitas Solusi

Berbeda dari sistem penjadwalan konvensional yang sering menghasilkan konflik atau pelanggaran kendala, framework ini menjamin 100% hard constraints terpenuhi melalui CSP validation, sambil mengoptimalkan soft constraints melalui GA. Jaminan ini memberikan confidence kepada institusi bahwa jadwal yang dihasilkan dapat langsung diimplementasikan tanpa perlu manual checking atau revision.

d. Skalabilitas dan Adaptabilitas

Framework dirancang dengan arsitektur modular yang memungkinkan adaptasi terhadap berbagai skala institusi dan jenis kendala. Penggunaan OR-Tools CP-SAT dengan parallel search dan parameter tuning yang sistematis memastikan bahwa sistem dapat menangani peningkatan kompleksitas data tanpa degradasi performa yang signifikan.

3. Manfaat Institusional

a. Peningkatan Efisiensi Operasional

Implementasi sistem ini dapat mengurangi waktu penyusunan jadwal dari berminggu-minggu menjadi beberapa jam, membebaskan sumber daya

manusia untuk fokus pada aktivitas bernilai tambah lainnya seperti pengembangan kurikulum dan peningkatan kualitas pembelajaran.

b. Peningkatan Kepuasan Stakeholder

Dengan mengoptimalkan soft constraints seperti free day dan minimisasi gap slots, sistem menghasilkan jadwal yang lebih user-friendly bagi mahasiswa dan dosen. Hal ini berpotensi meningkatkan kepuasan dan produktivitas akademik secara keseluruhan.

c. Dukungan Pengambilan Keputusan

Framework menyediakan insights tentang trade-off antara berbagai objectives melalui sensitivity analysis dan parameter tuning. Informasi ini dapat digunakan oleh manajemen institusi untuk membuat keputusan strategis terkait alokasi sumber daya dan kebijakan akademik.

4. Manfaat Jangka Panjang

a. Foundation untuk Penelitian Lanjutan

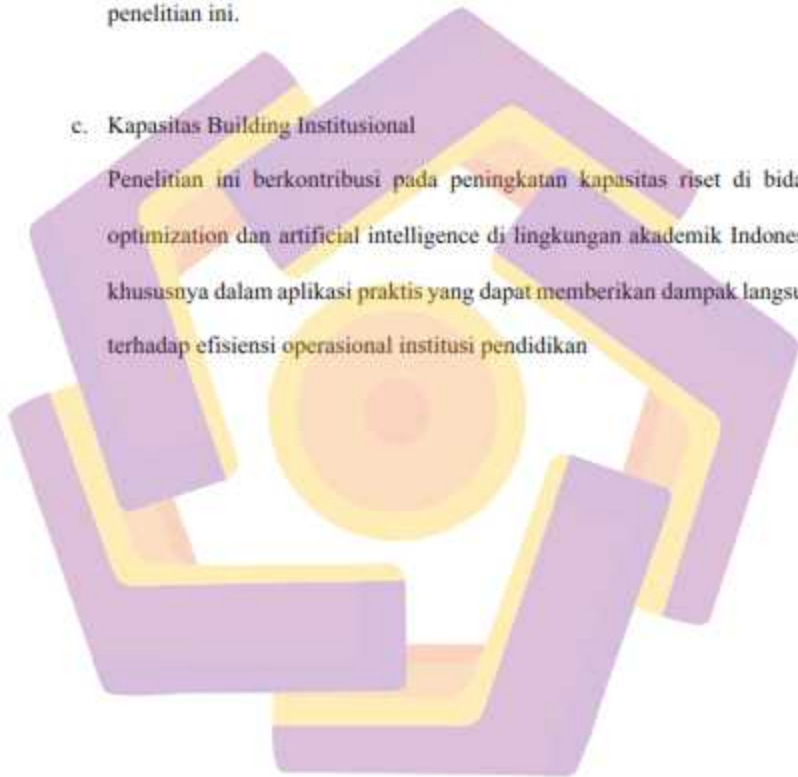
Framework yang dikembangkan dapat menjadi baseline untuk penelitian-penelitian selanjutnya dalam domain automated scheduling. Ekstensi potensial mencakup multi-objective optimization (NSGA-II), real-time rescheduling, dan integration dengan sistem informasi akademik yang lebih luas.

b. Transfer Knowledge ke Domain Lain

Prinsip-prinsip dan metodologi yang dikembangkan dapat ditransfer ke domain scheduling lainnya seperti nurse scheduling, manufacturing scheduling, atau transportation scheduling, memperluas dampak kontribusi penelitian ini.

c. Kapasitas Building Institusional

Penelitian ini berkontribusi pada peningkatan kapasitas riset di bidang optimization dan artificial intelligence di lingkungan akademik Indonesia, khususnya dalam aplikasi praktis yang dapat memberikan dampak langsung terhadap efisiensi operasional institusi pendidikan





BAB II

TINJAUAN PUSTAKA

2.1. Tinjauan Pustaka

Penelitian penjadwalan mata kuliah (*course timetabling*) berkembang melalui tiga pendekatan utama: *Constraint Satisfaction Problem (CSP)*, *Genetic Algorithm (GA)*, dan *hybrid metaheuristic*. Fokus utama dari seluruh pendekatan ini adalah pencapaian *feasibility* (tidak ada konflik) dan *quality optimization* (*minim penalti preferensi*). Bagian ini mengulas penelitian-penelitian relevan secara sistematis dan menyoroti kekuatan serta keterbatasannya.

1. Pendekatan CSP Murni

Pendekatan CSP memformulasikan penjadwalan sebagai triplet (V, D, C) dimana V adalah variabel sesi, D adalah domain slot dan ruang, serta C sebagai *hard constraints* (*kapasitas, no-overlap, tipe ruang, hari libur, dan blok slot*). Keunggulan CSP adalah jaminan solusi *feasible* melalui *constraint propagation* dan *backtracking search*. Namun, keterbatasannya terletak pada kemampuan optimasi—CSP tidak dapat secara efektif melakukan *multi-objective search* karena ruang solusi yang sangat besar dan fokus pada *feasibility*. Pada penelitian terdahulu, implementasi CSP umumnya dilakukan dengan *filtering sederhana* dan belum memanfaatkan teknologi *state-of-*

the-art seperti Google OR-Tools CP-SAT yang memiliki presolve agresif, parallel search, probing feasibility, solution hinting. Kesenjangan ini menjadi salah satu landasan kebaruan penelitian ini.

2. Genetic Algorithm (GA) Murni

GA populer digunakan dalam timetabling karena mampu menjelajahi ruang solusi besar, mendukung multi-kriteria, dan fleksibel dalam desain fungsi objektif. Penelitian oleh Herath & Wilkins (2024) menggunakan three-parent GA dengan uniform crossover dan menunjukkan peningkatan kualitas solusi juga Mahlous & Mahlous (2023) menitikberatkan pada preferensi mahasiswa dan menggunakan repair functions untuk menjamin feasibility. Namun, pola yang sama muncul, GA murni kesulitan menjamin hard constraints tanpa mekanisme repair yang kompleks dan seringkali mahal secara komputasi. Selain itu, penalti besar sering digunakan untuk menghindari konflik, sehingga GA rawan stagnasi pada solusi infeasible.

3. Hybrid GA dengan Local Search / Repair

Beberapa penelitian mencoba menggabungkan GA dengan teknik perbaikan Rezaeipanah (2021) Menggabungkan Parallel GA dan Local Search (IPGA-LS), kualitas meningkat, tetapi feasibility tetap tidak dijamin secara formal karena pemodelan kendala tidak

menggunakan CSP. Puspitasari (2020) Hybrid GA dengan Pattern Search Mencapai kualitas 99.24%, namun hanya berlaku pada dataset sederhana dan tanpa CSP, sehingga sulit digunakan pada masalah multi-entity seperti institusi besar. Kedua pendekatan ini meningkatkan kualitas namun tetap tidak mengatasi persoalan inti, yaitu jaminan feasibility dalam skala besar.

4. Hybrid GA + CSP (Penelitian Terdahulu yang Paling Relevan)

Penelitian Putra (2024) menggabungkan GA dan CSP untuk menyusun distribusi jadwal dosen. Temuan penting yang didapat adalah hasil feasibility mencapai 0 conflict, utilisasi ruang cukup optimal, namun memiliki beberapa keterbatasan fundamental seperti pada bagian CSP bukan CP-SAT, hanya filtering sederhana, tidak ada arsitektur dual-phase, tidak ada evaluasi parameter GA atau sensitivitas waktu, tidak memanfaatkan solution hinting, GA dan CSP belum terintegrasi secara algoritmik (CSP bukan feasibility engine). Keterbatasan inilah yang menjadi posisi kebaruan penelitian tesis ini.

5. Metaheuristik Multi-Objektif

Penelitian Al-Milli (2024) mengoptimasi jadwal pasca-enrolmen dengan NSGA-II, berfokus pada multi-objective preference optimization. Pendekatan ini unggul dalam memetakan trade-off antar preferensi, namun tidak menangani aspek feasibility awal, tidak

melibatkan CSP, hanya berlaku pada tahapan penyempurnaan jadwal (post-enrollment). Penelitian ini menguatkan kebutuhan penggabungan exact method (CSP) dengan metaheuristik.



2.2. Keaslian Penelitian

Keaslian penelitian ini didasarkan pada celah (gap) metodologis yang belum diisi oleh penelitian sebelumnya. Tabel berikut merangkum posisi tesis ini secara eksplisit.

Tabel 1. Matriks literatur review

No	Penelitian	Metode	Kelebihan	Kekurangan	Relevansi terhadap Tesis
1	Herath & Wilkins (2024)	GA Tiga-Parent	Kualitas lebih baik daripada tradisional	Tidak ada CSP, rawan infeasible	Menunjukkan GA butuh mekanisme constraint formal
2	Putra (2024)	GA dengan CSP sederhana	Zero conflict; utilisasi baik	Dataset kecil, CSP bukan CP-SAT, tidak ada hinting	Baseline hybrid, tetapi tidak scalable & tidak integratif
3	Puspitasari (2020)	GA dengan Pattern Search	Kualitas tinggi	Tidak menjamin hard constraints	Tesis menawarkan CSP sebagai solusi jaminan feasibility
4	Rezaeipannah (2021)	Parallel GA dan LS	Perbaikan signifikan kualitas	Tidak ada CSP; LS mahal	Menunjukkan kebutuhan feasibility engine
5	Al-Milli (2024)	NSGA-II	Trade-off preferensi optimal	Tidak menangani feasibility	Tesis menggabungkan CSP (feasibility) dengan GA (optimasi)

Berdasarkan matriks literatur tersebut, terlihat bahwa pendekatan GA memiliki fleksibilitas optimasi namun tidak menjamin terpenuhinya hard constraints, sedangkan CSP menyediakan validasi formal tetapi kurang efektif dalam eksplorasi multi-kriteria. Tidak ada penelitian yang secara sistematis menggabungkan CP-SAT sebagai feasibility engine dengan GA sebagai optimization engine pada dataset berskala besar. Kesenjangan-kesenjangan inilah yang menjadi landasan metodologis bagi penelitian ini.

Tabel 2 Matriks posisi penelitian

No	Aspek yang Dibandingkan	Pendekatan Penelitian Sebelumnya	Kondisi Penelitian Ini (Tesis)	Posisi Penelitian / Kebaruan
1	Penanganan Hard Constraints (Feasibility)	GA-only dengan penalti atau repair function (Herath & Wilkins 2024; Mahlous 2023). CSP sederhana tanpa propagasi (Putra 2024).	Menggunakan CP-SAT sebagai <i>feasibility engine</i> dengan presolve, propagation, parallel search, dan probing.	Jaminan feasibility formal. Tidak ada solusi infeasible selama GA berjalan.
2	Peran CSP dalam Hybrid	CSP hanya sebagai filtering atau pengecekan sederhana (Putra 2024). Tidak ada solution hinting.	CSP bertindak sebagai <i>repair-free validator</i> dan <i>solution refiner</i> menggunakan solution hinting dari GA.	Integrasi formal GA → CSP (hint), CSP → GA (valid domain).
3	Peran GA dalam Hybrid	GA bekerja bebas di ruang solusi besar, sering infeasible. Optimasi tidak terarah pada CSP.	GA-CSP-aware: representasi slot-only, semua mutasi & crossover tetap feasible.	Arsitektur GA yang sepenuhnya dikendalikan domain CSP.
4	Arsitektur Hybrid	Hybrid ad-hoc: GA dan Local Search, GA dan Pattern Search	Model dual-phase hybrid: (1) CSP menghasilkan solusi feasible cepat,	Dual-phase architecture yang belum diterapkan

		(Rezaeipanah 2021; Puspitasari 2020). Tidak dual-phase.	(2) GA + CSP hint mengoptimasi soft constraints secara terarah.	pada data akademik besar.
5	Fitur Soft Constraints	Fokus single-objective atau subset preferensi (NSGA-II post-enrolment oleh Al-Milli 2024).	Fungsi objective berbobot: early-time penalty, late-time penalty, free-day penalty, dev-hint.	Formulasi objective multi-kriteria terintegrasi dengan CSP-hinting.
6	Evaluasi Eksperimen	Umumnya hanya evaluasi kualitas solusi atau waktu komputasi dasar. Tanpa analisis sensitivitas.	Empat protokol eksperimen: CSP-only baseline, GA intensity, time-limit sensitivity, seed robustness.	Evaluasi komprehensif yang mengukur kualitas, stabilitas, robustitas.
7	Metode CSP	CSP klasik tanpa solver modern; tidak memakai CP-SAT (Putra 2024).	Menggunakan OR-Tools CP-SAT dengan presolve, parallel search, probing, hinting.	Penerapan CP-SAT untuk timetabling multi-entity skala besar.

Dari matriks posisi penelitian tersebut dapat disimpulkan bahwa tesis ini menempati ruang metodologis yang belum pernah dieksplorasi secara tuntas oleh penelitian sebelumnya, terutama pada integrasi formal CSP berbasis CP-SAT dengan GA CSP-aware, arsitektur dual-phase hybrid, serta validasi pada dataset multi-entity berskala institusional. Dengan demikian, kebaruan penelitian ini tidak hanya terletak pada penggabungan dua metode, tetapi juga pada desain integrasinya, skalabilitas dataset, dan protokol eksperimen yang komprehensif.

2.3. Landasan Teori

2.3.1 Constraint Satisfaction Problem (CSP)

Constraint Satisfaction Problem (CSP) memodelkan permasalahan penjadwalan sebagai triplet (V, D, C) yaitu V sebagai himpunan variabel, yaitu setiap sesi perkuliahan yang harus dijadwalkan, D merupakan domain nilai, berupa kombinasi timeslot dan room yang telah difilter kompatibel berdasarkan kapasitas dan tipe C adalah himpunan kendala *hard*, mencakup *No overlap* untuk dosen, cohort, dan room (tidak ada bentrok jadwal), ruang harus memiliki kapasitas dari dari sama dengan jumlah mahasiswa sesi, sesi LAB hanya di ruang LAB; sesi THEORY di ruang THEORY/ANY, sesi tidak dijadwalkan pada hari libur dosen, sesi tidak boleh melewati slot akhir yang diperbolehkan dosen slot waktu tertentu dilarang (misalnya Jumat S5-S6, Sabtu S7-S8).

Implementasi menggunakan Google OR-Tools CP-SAT Solver, yang menerapkan, *Constraint Propagation* untuk menyederhanakan dan memangkas domain saat *presolve*, *Probing level 2* untuk mendeteksi infeasibility sebelum pencarian utama, *Parallel search* dengan 8 worker, menjalankan strategi pencarian berbeda secara bersamaan, terakhir *Solution hinting*, menerima solusi awal dari GA agar solver memusatkan eksplorasi pada area solusi yang baik.

2.3.2 Genetic Algorithm (GA)

Algoritma Genetik (AG) adalah metode pencarian heuristik yang terinspirasi oleh proses seleksi alam yang diusulkan oleh Charles Darwin. AG merupakan

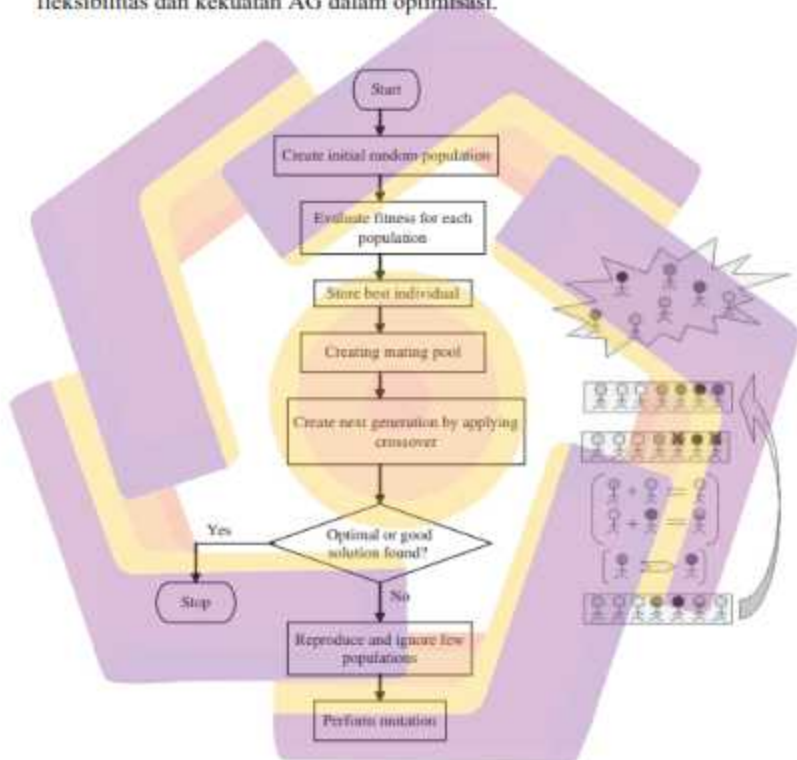
bagian dari keluarga algoritma evolusi dan komputasi evolusioner yang digunakan untuk menemukan solusi optimal atau mendekati optimal dalam masalah pencarian dan optimisasi. AG meniru mekanisme genetika dan seleksi alam seperti yang terjadi dalam evolusi biologi. Ini mencakup konsep seperti individu, populasi, seleksi, *crossover* (rekombinasi), dan mutasi (Goldberg, 1989)

Individu dalam AG diwakili oleh kromosom, yang biasanya berbentuk string biner yang menandakan solusi dari suatu masalah. Populasi terdiri dari kumpulan kromosom. Pada setiap generasi, individu dinilai menggunakan fungsi kebugaran (*fitness function*), dan individu dengan kebugaran lebih tinggi memiliki peluang lebih besar untuk dipilih untuk reproduksi. Teknik *crossover* memungkinkan pertukaran informasi genetik antar individu, sementara mutasi memperkenalkan variasi dengan mengubah secara acak bagian dari kromosom.

Dalam konteks penjadwalan, AG dapat digunakan untuk mengatur alokasi waktu dan sumber daya dengan cara yang optimal dengan memperhatikan batasan yang ada. Misalnya, dalam penjadwalan mata kuliah, AG dapat membantu mengatur jadwal dengan menghindari konflik kelas dan memaksimalkan penggunaan sumber daya seperti ruang kelas dan waktu dosen.

"Algoritma genetik menawarkan kerangka kerja fleksibel dan kuat untuk menyelesaikan masalah optimisasi, yang dapat beradaptasi dengan berbagai tantangan dalam penjadwalan dan alokasi sumber daya." -(Goldberg, 1989) dalam bukunya "*Genetic Algorithms in Search, Optimization, and Machine Learning*" menggambarkan algoritma genetik sebagai teknik adaptif yang mampu mengidentifikasi solusi dari ruang pencarian yang besar dan kompleks.

"Kemampuan algoritma genetik untuk menangani ruang pencarian yang luas dan menemukan solusi optimal dalam masalah kompleks menjadikannya sebagai salah satu alat paling fleksibel dan kuat dalam optimisasi." - (Sivanandam & Deepa, 2008) dalam bukunya *"Introduction to Genetic Algorithms"* menekankan fleksibilitas dan kekuatan AG dalam optimisasi.



Gambar 1 Flowchart of genetic algorithm

(Sumber: Sivanandam, S. N., et al., 2008)

Pada gambar 1 proses algoritma genetik dimulai dengan langkah pertama yaitu membuat populasi awal secara acak. Populasi ini terdiri dari sejumlah solusi awal yang dihasilkan secara acak. Setelah itu, dilakukan evaluasi *fitness* untuk

setiap individu dalam populasi berdasarkan fungsi *fitness* yang telah ditentukan. Individu dengan nilai *fitness* tertinggi disimpan sebagai solusi terbaik sementara.

Selanjutnya, dibentuk *mating pool* yang berisi individu-individu yang akan dikawinkan berdasarkan nilai *fitness* mereka. Operasi *crossover* atau pindah silang digunakan untuk menciptakan generasi baru dengan menggabungkan bagian dari kromosom individu-individu terpilih. Proses ini berulang hingga solusi yang sudah cukup baik atau optimal ditemukan, dan jika demikian, algoritma berhenti.

Komponen Genetic Algorithm (GA) terdiri dari Representasi Kromosom memetakan setiap sesi ke timeslot tertentu; assignment ruang diserahkan ke CSP. Population initialization Membuat populasi awal berisi individu-individu yang valid (tidak melanggar hard constraints). Seleksi Tournament selection di antara 50% populasi dengan fitness terbaik. Crossover Uniform crossover—untuk setiap gen (sesi), slot diturunkan secara acak dari salah satu orang tua. Mutasi Setiap gen memiliki peluang 12% untuk diganti ke slot lain yang valid. Elitisme Dua individu dengan fitness terbaik dipertahankan ke generasi berikutnya tanpa perubahan. Fitness evaluation Setiap individu dievaluasi dengan menjalankan CSP solver menggunakan hint individu; fitness adalah nilai objective yang dihasilkan CSP.

2.3.3 Multi-Objective Weighted Penalty Function

Fungsi objektif menggabungkan penalti untuk soft constraints:

$$\begin{aligned} \text{Objective} = & w_{s1} \sum_i \text{is_s1}_i + w_{sB} \sum_i \text{is_sB}_i + w_{\text{free_day}} \sum_c \text{no_free_day}_c \\ & + \sum_i \text{dev_hint}_i \end{aligned}$$

1. $w_{s1}, w_{s8}, w_{free_day}$: bobot penalti untuk sesi pagi, sesi akhir, dan cohort tanpa hari bebas.
2. $is_s1, is_s8, no_free_day, dev_hint$: variabel biner yang aktif saat penalti terjadi.

Hard constraints seperti no-overlap (dosen, cohort, ruang), kapasitas ruang, tipe ruang (LAB/THEORY), hari libur dosen, dan blocked slots tidak diberikan penalti, karena seluruh hard constraints dijamin terpenuhi oleh CSP melalui mekanisme constraint propagation dan validasi formal CP-SAT. Dengan demikian, fungsi objektif hanya berfokus pada kualitas solusi (solution quality) setelah solusi feasible diperoleh. Penggunaan fungsi penalti berbobot ini memungkinkan GA melakukan eksplorasi multi-kriteria secara fleksibel, sementara CSP memastikan seluruh kandidat tetap valid. Pendekatan ini menghasilkan keseimbangan yang efektif antara feasibility dan optimality dalam kerangka hybrid CSP-GA.

Walaupun GA banyak digunakan dalam penjadwalan, pendekatan ini bukanlah satu-satunya opsi yang dapat diambil. Literatur terkini menunjukkan peningkatan performa pada algoritma seperti NSGA-II, Adaptive LNS, PSO, dan Tabu Search. Dengan demikian, pemilihan GA dalam penelitian ini bukan didasarkan pada prevalensi penggunaannya, melainkan pada kesesuaiannya terhadap struktur masalah dan kemampuan integratifnya dengan CSP sebagai validator formal. Penjelasan ini sekaligus menyadarkan bahwa penelitian ini masih terbuka untuk diuji terhadap algoritma lain pada fase lanjutan.

BAB III

METODE PENELITIAN

3.1. Jenis, Sifat, dan Pendekatan Penelitian

Penelitian ini merupakan penelitian terapan dengan pendekatan eksperimental-komputasional, yang bertujuan meng-

efektivitas integrasi Constraint Satisfaction Problem (CSP) berbasis CP-SAT dan Genetic Algorithm (GA) dalam menghasilkan jadwal kuliah yang feasible dan optimal. Pendekatan ini digunakan untuk mengukur performa metode pada tiga dimensi. Akurasi (*feasibility rate*) yaitu kemampuan model menghasilkan jadwal tanpa konflik pada dosen, cohort, ruang, kapasitas, tipe ruang, hari libur, last slot, dan slot terblokir. Efisiensi (*execution time*) waktu penyelesaian pada fase CSP maupun fase integrasi GA-CSP. Kualitas solusi (*objective value*) sejauh mana penalti soft constraints dapat diminimalkan melalui GA. Penelitian ini menggunakan empat set eksperimen: baseline CSP, intensitas GA, batas waktu GA, dan robustitas seed.

3.2. Metode Pengumpulan Data

Data diperoleh dari sistem akademik Universitas Duta Bangsa Surakarta dan terdiri atas empat tabel utama dalam file UDB_Data.xlsx:

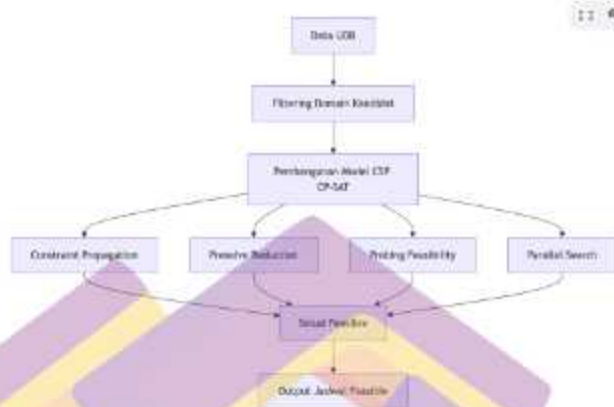
1. Cohort (17 baris)
 - a. Atribut: Cohort dan Size

- b. Representasi kelompok mahasiswa; total mahasiswa 554 (rata-rata 32.6 per cohort).
 2. CourseAssign (116 baris)
 - a. Atribut: Cohort, Course, Lecturer, Type, SKS
 - b. Setiap baris merepresentasikan satu sesi perkuliahan (85 THEORY, 31 LAB; 3 SKS – 91, 2 SKS – 25).
 3. Rooms (19 baris)
 - a. Atribut: RoomID, Capacity, Type
 - b. Terdiri atas 14 ruang THEORY dan 5 ruang LAB, kapasitas berkisar 35–80 mahasiswa.
 4. Lecturers (41 baris)
 - a. Atribut: Lecturer, DayOff, LastSlot, Skills
 - b. 22 dosen memiliki hari libur spesifik; semua dosen memiliki batas slot terakhir pada S8.

Data divalidasi untuk konsistensi referensial, misalnya memastikan setiap entry CourseAssign.Cohort ada di tabel Cohort, kapasitas mutu, dan tipe ruang kompatibel.

3.3. Pemodelan CSP (Feasibility Engine)

CSP berfungsi sebagai feasibility engine yang menjamin pemenuhan seluruh hard constraints melalui pemanfaatan kemampuan CP-SAT yaitu propagasi domain, presolve reduction, probing feasibility, dan parallel search.



Gambar 2 Diagram Arsitektur CSP

Diagram ini yang ditunjukkan pada gambar 2 menggambarkan alur kerja penyelesaian jadwal pada fase CSP, dimulai dari pemrosesan data UDB hingga terbentuknya solusi feasible. Data yang telah dikumpulkan terlebih dahulu difilter untuk menghasilkan domain kandidat slot dan ruang yang sesuai dengan batasan kapasitas, tipe ruang, hari libur dosen, dan slot terblokir. Domain yang telah disaring kemudian digunakan untuk membangun model CP-SAT yang berisi variabel interval dan seluruh hard constraints.

Setelah model terbentuk, CP-SAT menjalankan beberapa tahapan pemrosesan internal, yaitu constraint propagation untuk mempersempit domain berdasarkan aturan, presolve reduction untuk menghilangkan nilai dan struktur yang tidak relevan, probing untuk mendeteksi konflik lebih awal, serta parallel search untuk mengeksplorasi ruang solusi secara simultan melalui beberapa worker. Hasil gabungan dari seluruh tahapan tersebut menghasilkan satu jadwal yang bebas

konflik dan memenuhi seluruh hard constraints, yang kemudian menjadi output feasible untuk digunakan pada fase optimasi selanjutnya.

3.3.1 Variabel Model

Dalam model CSP berbasis CP-SAT, setiap sesi perkuliahan direpresentasikan sebagai interval variable yang memuat waktu mulai, durasi, dan ruang. Interval ini terdiri atas Start (slot awal), Duration (jumlah slot sesuai SKS), dan End yang dihitung otomatis. Penempatan ruang direpresentasikan melalui Room_i, sehingga setiap sesi memiliki satu kombinasi slot-ruang yang harus dipenuhi tanpa konflik. Variabel ini membentuk struktur dasar untuk mendeteksi tumpang-tindih jadwal secara otomatis melalui mekanisme *NoOverlap*.

Domain dari variabel tersebut telah disaring sehingga hanya memuat kombinasi slot dan ruang yang feasible menurut kapasitas, tipe ruang, hari libur dosen, last slot, dan slot terblokir. Dengan penyempitan domain sejak awal, CP-SAT bekerja pada ruang solusi yang lebih kecil dan lebih relevan.

Relasi antar-variabel mengikuti struktur one-to-many pada entitas dosen, cohort, dan ruang, satu dosen mengajar banyak sesi, satu cohort memiliki banyak sesi, dan satu ruang dapat digunakan banyak sesi secara bergiliran. Sebaliknya, setiap sesi hanya memiliki satu ruang dan satu slot mulai (one-to-one). Relasi ini diterapkan melalui pengelompokan interval berdasarkan dosen, cohort, dan ruang, sehingga solver dapat menegakkan NoOverlap untuk masing-masing kelompok.

Model variabel CSP tersebut menyediakan representasi yang ringkas namun lengkap, memungkinkan CP-SAT menjamin seluruh hard constraints dan menyediakan dasar evaluasi yang konsisten pada fase integrasi GA.

3.3.2 Domain

Domain dalam model CSP mendefinisikan sekumpulan nilai yang mungkin diambil oleh setiap variabel, khususnya slot waktu dan ruang kelas. Pada penelitian ini, domain tidak dibangun secara penuh, tetapi disaring secara ketat agar hanya memuat kombinasi slot-ruang yang benar-benar feasible. Penyaringan domain dilakukan berdasarkan karakteristik sesi, dosen, cohort, serta tipe dan kapasitas ruang. Dengan demikian, sejak awal CP-SAT bekerja pada ruang solusi yang telah dipersempit sehingga proses propagasi lebih efektif.

Untuk setiap sesi, domain slot hanya mencakup slot yang sesuai dengan hari aktif dosen, tidak melampaui batas slot terakhir yang diperbolehkan, dan tidak berada pada rentang waktu yang diblokir program studi. Demikian pula, domain ruang dibatasi hanya pada ruang yang kapasitasnya memadai dan jenisnya sesuai dengan tipe perkuliahan (LAB atau THEORY). Kombinasi slot-ruang yang melanggar ketentuan tersebut tidak diikutsertakan sebagai kandidat.

Penyempitan domain ini menghasilkan dua manfaat penting. Pertama, CP-SAT memperoleh candidate set yang lebih terarah sehingga proses feasibility menjadi lebih cepat dan stabil. Kedua, domain yang sudah terverifikasi valid ini digunakan kembali oleh GA pada fase optimasi, sehingga mutasi maupun crossover tidak akan menghasilkan solusi yang tidak mungkin dipenuhi oleh constraint ruang

atau dosen. Dengan cara ini, domain berfungsi sebagai fondasi yang menjaga agar keseluruhan arsitektur hybrid tetap bekerja pada solusi-solusi yang dapat direalisasikan.

3.3.3 Hard Constraint

Hard constraints merupakan aturan yang harus dipenuhi secara absolut oleh setiap solusi jadwal. Pada penelitian ini, seluruh hard constraints dimodelkan secara eksplisit dalam CP-SAT melalui interval variables dan mekanisme NoOverlap, sehingga solusi yang dihasilkan selalu valid sebelum memasuki fase optimasi GA. Hard constraints mencakup batasan yang berasal dari dosen, cohort, ruang, serta struktur dasar jadwal perkuliahan.

Pertama, setiap sesi harus ditempatkan pada waktu yang tidak menimbulkan konflik. Ini berarti seorang dosen tidak dapat mengajar dua sesi pada slot yang sama, cohort tidak dapat mengikuti dua mata kuliah secara bersamaan, dan sebuah ruang tidak boleh digunakan oleh lebih dari satu sesi secara berurutan. CP-SAT menangani aturan ini dengan menempatkan interval sesi dalam kelompok-kelompok khusus—kelompok dosen, kelompok cohort, dan kelompok ruang—yang masing-masing diberi NoOverlap agar tidak terjadi benturan waktu.

Selain non-overlap, terdapat batasan tambahan yang memastikan kesesuaian ruang dan waktu dengan karakteristik sesi. Ruang yang dipilih harus memiliki kapasitas yang memadai dan tipe yang sesuai (misalnya sesi lab hanya dapat ditempatkan di ruang berjenis LAB). Di sisi lain, constraint mengenai ketersediaan dosen, seperti hari libur dan slot pengajaran terakhir, menjadi dasar

penyaringan domain dan sekaligus dipaksakan ulang dalam model agar sesi tidak ditempatkan pada waktu yang melanggar aturan tersebut. Slot yang secara institusional diblokir juga dihilangkan dari domain sehingga tidak mungkin digunakan dalam solusi apa pun.

Dengan pemodelan ini, CP-SAT menjamin bahwa setiap solusi memenuhi seluruh hard constraints, terlepas dari bagaimana GA mengeksplorasi ruang solusi pada fase berikutnya. Hal ini memastikan integrasi CSP-GA tetap berada dalam ruang jadwal yang valid, sehingga proses optimasi benar-benar berfokus pada pengurangan penalti soft constraints tanpa mengorbankan kelayakan jadwal.

3.3.4 Soft Constraint

Soft constraints digunakan untuk mengarahkan kualitas jadwal menuju preferensi institusi tanpa membatalkan solusi apabila dilanggar. Berbeda dengan hard constraints yang bersifat absolut, soft constraints bersifat fleksibel dan hanya menimbulkan penalti ketika kondisinya tidak terpenuhi. Dalam penelitian ini, soft constraints dimodelkan menggunakan variabel biner sehingga setiap pelanggaran dapat terukur dan terakumulasi dalam fungsi objektif.

Salah satu bentuk preferensi ialah kecenderungan untuk menghindari penempatan sesi pada slot pertama atau slot terakhir dalam suatu hari, baik untuk menjaga kenyamanan mahasiswa maupun dosen. Untuk itu, setiap sesi yang jatuh pada slot pagi atau slot akhir diberi penanda biner yang bernilai satu, sedangkan sesi lain bernilai nol. Demikian pula, adanya keinginan agar setiap cohort memiliki

setidaknya satu hari bebas menghasilkan indikator khusus yang mencatat apabila sebuah cohort tidak memperoleh free-day.

Penelitian ini juga memasukkan variabel deviasi terhadap solution hint sebagai bagian dari soft constraints. Deviasi ini muncul ketika solusi yang dihasilkan CSP pada fase optimasi berbeda jauh dari usulan GA, dan indikator tersebut membantu menjaga agar proses optimasi tetap stabil serta tidak berpindah terlalu jauh dari saran GA yang sudah relatif baik.

Seluruh indikator tersebut berkontribusi pada nilai objektif yang ingin diminimalkan. Dengan cara ini, solver tetap bebas memilih konfigurasi jadwal mana pun selama seluruh hard constraints terpenuhi, tetapi akan "dihukum" apabila terlalu banyak melanggar preferensi operasional. Pendekatan seperti ini memberikan keseimbangan antara kelayakan jadwal dan kenyamanan akademik, sekaligus memungkinkan GA berperan efektif untuk menekan penalti secara bertahap.

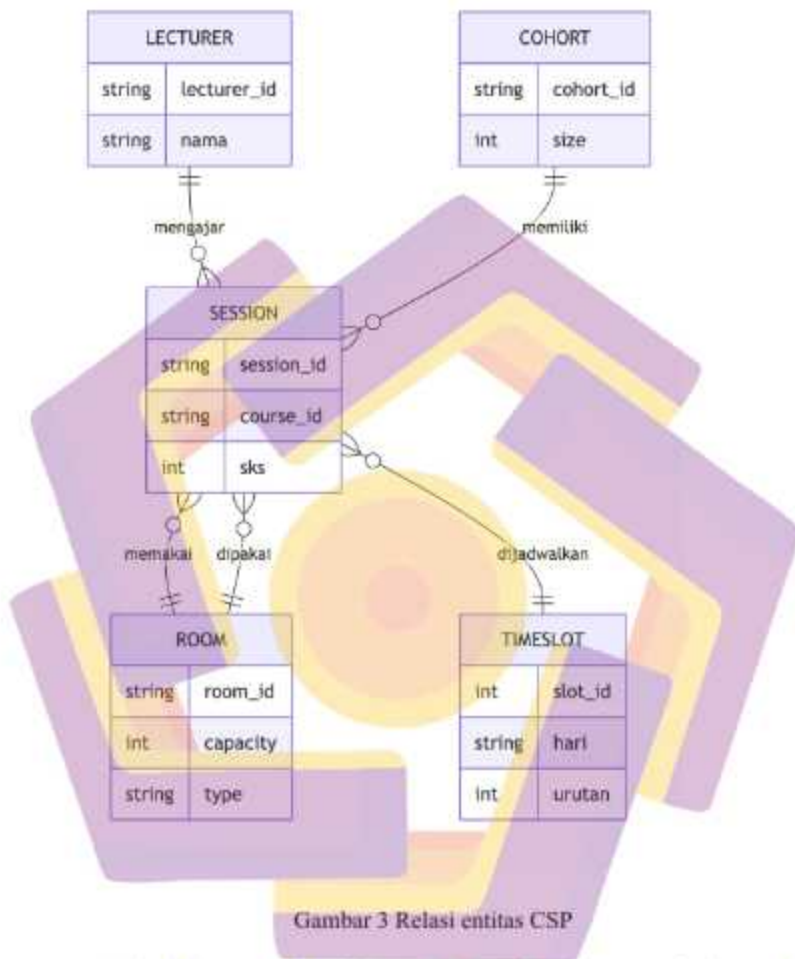
3.3.5 Relasi Antar Entitas

Struktur penjadwalan dalam model CSP membentuk pola relasi yang konsisten antar entitas utama, yaitu dosen, cohort, ruang, sesi, dan slot waktu. Seorang dosen dapat mengajar banyak sesi, sebuah cohort dapat memiliki banyak sesi perkuliahan, dan sebuah ruang dapat dipakai oleh banyak sesi secara bergiliran sepanjang tidak terjadi benturan waktu. Sebaliknya, setiap sesi selalu dihubungkan hanya dengan satu dosen, satu cohort, satu ruang, dan satu slot mulai. Pola ini secara

alamiah membentuk relasi one-to-many pada sisi dosen, cohort, dan ruang terhadap sesi, serta one-to-one dari sesi menuju ruang dan slot.

Relasi ini tidak hanya penting pada level konseptual, tetapi juga menjadi dasar pembentukan kelompok constraint dalam CSP, misalnya kelompok interval per dosen, per cohort, dan per ruang yang masing-masing diberi aturan NoOverlap untuk menjamin tidak ada dua sesi yang berjalan bersamaan pada kombinasi entitas yang sama. Dengan demikian, pemahaman relasi one-to-many ini memastikan struktur constraint yang dibangun bersifat konsisten secara matematis dan sesuai dengan realitas operasional penjadwalan.





Relasi ini yang di tunjukkan pada gambar 3 menggambarkan struktur keterhubungan antar entitas dalam model CSP, di mana seorang dosen dan sebuah cohort masing-masing dapat memiliki banyak sesi, sedangkan setiap sesi hanya memiliki satu ruang dan satu slot waktu. Ruang juga dapat dipakai banyak sesi selama tidak bertumpukan waktu. Struktur relasi ini menjadi dasar pembentukan

constraint NoOverlap dan memastikan seluruh sesi tersusun secara konsisten dalam jadwal.

3.3.6 Solver CP-SAT

Solver CP-SAT digunakan sebagai mesin utama untuk memastikan seluruh hard constraints terpenuhi secara konsisten. CP-SAT menggabungkan teknik constraint programming, SAT solving, dan integer optimization, sehingga mampu menangani model penjadwalan dengan variabel interval secara efisien. Pada penelitian ini, CP-SAT menjalankan rangkaian proses penyelesaian yang meliputi penyempitan domain, propagasi constraint, dan pencarian solusi secara paralel.

Pada tahap awal, CP-SAT melakukan presolve reduction untuk menghapus nilai domain atau struktur model yang tidak relevan, sehingga ukuran masalah menyusut dan pencarian menjadi lebih cepat. Selanjutnya, constraint propagation digunakan untuk menyimpulkan nilai-nilai yang tidak mungkin dipilih akibat benturan aturan, misalnya slot yang tidak dapat ditempati dosen tertentu atau ruang yang tidak sesuai kapasitas. Proses propagasi ini terus diperbarui secara dinamis setiap kali nilai variabel lain berubah.

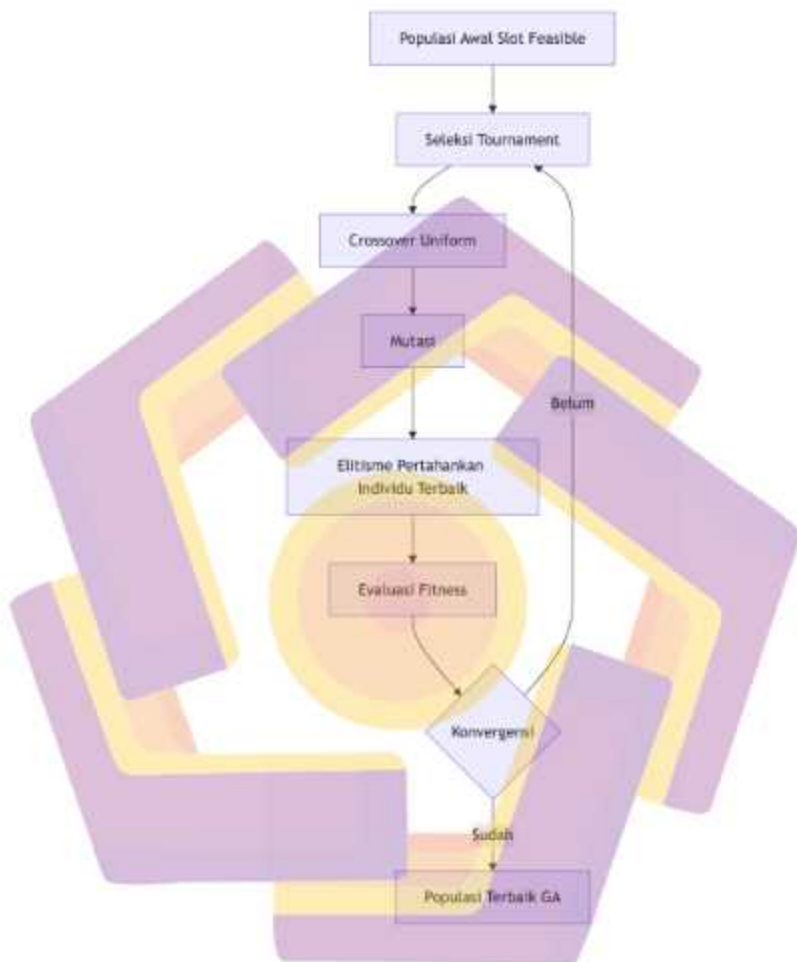
Solver kemudian memanfaatkan parallel search dengan beberapa worker yang menjalankan strategi pencarian berbeda secara bersamaan. Setiap worker mengeksplorasi cabang solusi alternatif, dan hasil terbaik dipertukarkan di antara worker untuk mempercepat konvergensi. Ketika terjadi konflik atau pelanggaran constraint dalam proses pencarian, CP-SAT menggunakan mekanisme conflict-driven learning untuk mempelajari pola konflik tersebut dan menghindari

percabangan serupa pada langkah berikutnya. Pendekatan ini membuat CP-SAT lebih stabil dan efektif dibanding pendekatan CSP klasik.

Melalui kombinasi teknik-teknik tersebut, CP-SAT mampu menghasilkan solusi feasible secara deterministik pada fase pertama, sekaligus menjadi evaluator dalam fase optimasi saat menerima solution hint dari GA. Dengan demikian, CP-SAT berperan sebagai penjamin validitas dan pengontrol kualitas jadwal di seluruh tahapan metode hybrid.

3.4. Desain Genetic Algorithm (Optimization Engine)

Fase optimasi dalam penelitian ini dilakukan menggunakan Genetic Algorithm (GA), yang berfungsi untuk meningkatkan kualitas jadwal dengan meminimalkan penalti pada soft constraints setelah solusi feasible dihasilkan oleh CSP. GA dipilih karena kemampuannya mengeksplorasi ruang solusi yang luas melalui mekanisme evolusioner, sekaligus tetap fleksibel terhadap berbagai preferensi penjadwalan yang bersifat multi-kriteria. Dalam arsitektur hybrid yang digunakan, GA tidak berdiri sendiri, melainkan bekerja secara terintegrasi dengan CSP: CSP menjamin seluruh solusi tetap valid, sementara GA mengarahkan proses pencarian menuju konfigurasi slot yang lebih optimal. Bagian ini menjelaskan struktur representasi kromosom, mekanisme pembentukan populasi, serta operator-operator evolusi yang mengatur proses pencarian solusi terbaik pada fase optimasi.



Gambar 4 Diagram Arsitektur GA.

Pada gambar 4 menunjukkan alur kerja Genetic Algorithm (GA) sebagai mesin optimasi pada fase kedua. Proses dimulai dari populasi awal yang dibentuk dari slot-slot feasible hasil fase CSP, sehingga setiap individu pada generasi pertama berada dalam ruang solusi yang valid. GA kemudian menjalankan

serangkaian operator evolusi, dimulai dari seleksi dengan metode tournament untuk memilih individu-individu yang lebih baik, diikuti uniform crossover untuk menghasilkan kombinasi gen baru, serta mutasi untuk memperkenalkan variasi tambahan dalam populasi. Setelah mutasi, mekanisme elitisme mempertahankan individu terbaik agar kualitas solusi tidak menurun antar generasi.

Setiap individu kemudian dievaluasi melalui proses fitness evaluation, yang dalam penelitian ini dilakukan dengan memanfaatkan CSP sebagai evaluator resmi sehingga seluruh solusi tetap memenuhi hard constraints. Proses evolusi berulang hingga memenuhi kondisi konvergensi, dan hasil akhirnya adalah populasi terbaik yang menunjukkan konfigurasi slot dengan nilai penalti soft constraints paling rendah.

3.4.1 Representasi Kromosom

Pada penelitian ini, setiap individu dalam Genetic Algorithm direpresentasikan sebagai sebuah kromosom yang terdiri atas sejumlah gen yang masing-masing mewakili satu sesi perkuliahan. Nilai yang disimpan dalam setiap gen adalah start timeslot dari sesi tersebut, yaitu titik waktu awal tempat sesi akan dijadwalkan. Dengan demikian, sebuah kromosom menggambarkan konfigurasi lengkap penempatan waktu seluruh sesi dalam satu solusi.

Representasi ini mengikuti pendekatan slot-only chromosome, di mana GA hanya mengoptimalkan aspek penjadwalan waktu. Penugasan ruang tidak dimasukkan ke dalam struktur kromosom, karena pengalokasian ruang dilakukan

oleh CSP melalui mekanisme interval dan NoOverlap yang memastikan seluruh ruang digunakan secara konsisten. Pemisahan peran ini membuat GA dapat berfokus sepenuhnya pada optimasi soft constraints, sementara CSP menjamin bahwa setiap kombinasi slot yang dihasilkan tetap memenuhi seluruh hard constraints terkait ruang, dosen, dan cohort.

Pendekatan ini juga menghasilkan ruang pencarian yang lebih terstruktur, karena domain setiap gen telah dibatasi hanya pada slot-slot yang feasible menurut hasil penyaringan CSP pada fase sebelumnya. Dengan demikian, representasi kromosom tidak hanya sederhana dan efisien, tetapi juga konsisten dengan desain arsitektur hybrid yang menempatkan CSP sebagai feasibility engine dan GA sebagai optimization engine.

3.4.2 Inisialisasi Populasi

Proses inisialisasi populasi dilakukan dengan membangun sejumlah individu awal yang seluruhnya berasal dari domain slot yang telah difilter oleh CSP pada fase feasibility. Setiap gen dalam kromosom dipilih secara acak dari kumpulan slot yang dinyatakan feasible untuk sesi terkait, sehingga tidak ada sesi yang ditempatkan pada waktu yang melanggar kapasitas ruang, tipe ruang, hari libur dosen, maupun slot terblokir. Dengan cara ini, setiap individu pada generasi pertama sudah memenuhi struktur dasar jadwal yang sah sebelum memasuki mekanisme evolusi.

Pendekatan ini memastikan bahwa GA tidak membuang waktu mengeksplorasi solusi yang secara inheren tidak mungkin dipenuhi oleh constraint, karena seluruh nilai awal telah divalidasi melalui hasil analisis domain CSP. Selain

mempercepat proses evolusi, strategi ini juga menjaga stabilitas pencarian dan mempersempit ruang solusi menjadi hanya konfigurasi yang realistis. Dengan demikian, populasi awal tidak hanya acak, tetapi juga feasibility-aware, selaras dengan peran CSP sebagai fondasi dalam arsitektur hybrid.

3.4.3 Operator Evolusi GA

Proses evolusi dalam Genetic Algorithm dikendalikan oleh serangkaian operator yang bekerja secara berurutan untuk menghasilkan populasi berikutnya. Operator-operator ini dirancang agar mampu mengeksplorasi ruang solusi secara luas, namun tetap mempertahankan stabilitas pencarian sehingga kualitas jadwal meningkat pada setiap generasi.

Tahap pertama adalah seleksi, yang dalam penelitian ini menggunakan metode tournament selection untuk memilih individu-individu yang lebih baik dari populasi saat ini. Pendekatan ini memberikan peluang lebih tinggi bagi solusi berkualitas, sekaligus tetap menjaga keberagaman populasi. Setelah seleksi, pasangan individu terpilih menjalani crossover dengan skema uniform crossover, di mana nilai setiap gen dapat diturunkan dari salah satu parent secara acak. Mekanisme ini memungkinkan pertukaran informasi yang efisien antar kromosom dan membuka peluang kombinasi slot yang lebih bervariasi.

Tahap berikutnya adalah mutasi, yang dilakukan dengan probabilitas tertentu untuk mengubah nilai gen secara acak. Mutasi memberikan variasi baru dalam populasi dan mencegah GA terjebak pada solusi lokal. Karena domain slot telah difilter oleh CSP, setiap mutasi dijamin tetap berada dalam ruang solusi yang feasible. Setelah mutasi, diterapkan elitisme untuk memastikan bahwa individu

terbaik pada generasi sebelumnya tetap dipertahankan. Strategi ini menjaga kualitas solusi tidak menurun selama proses evolusi.

Rangkaian operator ini bekerja secara iteratif hingga mencapai kondisi konvergensi, dengan tujuan akhir menghasilkan konfigurasi jadwal yang memenuhi seluruh hard constraints dan memiliki penalti soft constraints serendah mungkin.

3.4.4 Elitisme

Elitisme digunakan untuk memastikan bahwa kualitas solusi tidak menurun dari satu generasi ke generasi berikutnya. Dalam pendekatan ini, sejumlah individu dengan nilai fitness terbaik pada populasi saat ini dipertahankan secara langsung tanpa mengalami proses seleksi, crossover, maupun mutasi. Dengan demikian, solusi terbaik yang telah ditemukan tidak akan hilang akibat variasi acak selama proses evolusi.

Pada penelitian ini, elitisme diterapkan dengan mempertahankan beberapa individu berkinerja tertinggi pada setiap generasi. Pendekatan ini memberikan stabilitas tambahan pada proses pencarian, karena GA tetap bebas mengeksplorasi ruang solusi melalui crossover dan mutasi, namun tetap memiliki jangkar kualitas melalui individu terbaik yang diturunkan secara pasti. Keberadaan elitisme juga mempercepat konvergensi, terutama ketika GA mulai menemukan pola konfigurasi slot yang secara konsisten menghasilkan penalti rendah pada soft constraints.

Dengan demikian, elitisme berfungsi sebagai mekanisme pengaman kualitas sekaligus penguat arah pencarian, menjaga agar proses evolusi tetap progresif dan terfokus pada peningkatan kualitas jadwal.

3.4.5 Evaluasi Fitness (Menggunakan CSP)

Evaluasi fitness pada penelitian ini tidak dilakukan melalui perhitungan langsung di dalam GA, tetapi menggunakan CSP sebagai evaluator resmi. Setiap kromosom yang dihasilkan GA diterjemahkan menjadi usulan jadwal melalui mekanisme solution hint, di mana nilai start timeslot pada masing-masing gen diberikan kepada CP-SAT sebagai preferensi awal. CSP kemudian membangun ulang interval berdasarkan hint tersebut dan memastikan bahwa seluruh hard constraints tetap terpenuhi melalui proses propagasi dan pemeriksaan konsistensi.

Setelah model tervalidasi, CP-SAT menghitung nilai objektif berdasarkan pelanggaran soft constraints, seperti penempatan sesi pada slot pertama atau terakhir, ketiadaan hari bebas pada cohort, serta deviasi terhadap hint GA. Nilai objektif ini kemudian dikembalikan sebagai fitness untuk kromosom tersebut. Dengan pendekatan ini, fitness selalu menggambarkan kualitas jadwal yang benar-benar dapat direalisasikan, bukan sekadar estimasi atau perhitungan heuristik.

Metode evaluasi ini memberikan dua keuntungan penting. Pertama, hasil fitness selalu valid terhadap seluruh hard constraints, sehingga GA tidak perlu mengelola penalti untuk konflik dasar seperti benturan dosen atau ruang. Kedua, CSP berfungsi sebagai repair mechanism yang memperbaiki individu GA jika terjadi konfigurasi yang tidak stabil, sehingga proses optimasi berjalan lebih lancar dan tidak terjebak pada solusi yang tidak layak. Integrasi seperti ini memperkuat peran CSP sebagai feasibility engine sekaligus quality assessor pada fase optimasi.

3.4.6 Kondisi Konvergensi

Proses evolusi pada Genetic Algorithm berlangsung secara iteratif hingga terpenuhi suatu kondisi penghentian yang menandai bahwa pencarian tidak lagi memberikan peningkatan berarti. Dalam penelitian ini, kondisi konvergensi ditetapkan berdasarkan dua kriteria utama: batas jumlah generasi dan stabilitas nilai fitness. GA akan berhenti ketika jumlah generasi yang direncanakan telah tercapai, atau ketika nilai objektif tidak menunjukkan penurunan signifikan dalam beberapa generasi berturut-turut.

Pendekatan ini memungkinkan proses evolusi berlangsung cukup lama untuk menemukan konfigurasi slot yang optimal, namun tetap mencegah pencarian yang berlarut-larut tanpa peningkatan kualitas. Stabilitas fitness diamati untuk mendeteksi situasi ketika GA mulai memasuki titik jenuh, di mana variasi yang dihasilkan oleh crossover maupun mutasi tidak lagi mengubah kualitas jadwal secara berarti. Dengan demikian, konvergensi berfungsi sebagai indikator bahwa proses pencarian telah menghasilkan solusi terbaik yang dapat diperoleh dalam batas waktu dan parameter evolusi yang ditetapkan.

Ketika kondisi konvergensi terpenuhi, individu dengan nilai fitness terbaik pada populasi terakhir dipilih sebagai solusi final GA, yang kemudian menjadi bagian dari hasil akhir pada arsitektur hybrid CSP-GA.

3.5. Fungsi Objektif Multi-Kriteria

Fungsi objektif pada penelitian ini dirancang untuk mengukur kualitas jadwal berdasarkan sejumlah soft constraints yang mencerminkan preferensi operasional program studi. Tidak seperti hard constraints yang wajib dipenuhi tanpa pengecualian, pelanggaran soft constraints hanya menghasilkan penalti, dan seluruh penalti tersebut dijumlahkan menjadi satu nilai objektif yang ingin diminimalkan. Dengan demikian, fungsi objektif berfungsi sebagai kompas yang mengarahkan proses optimasi agar jadwal yang dihasilkan tetap feasible sekaligus nyaman secara akademik.

Secara umum, fungsi objektif merupakan penjumlahan penalti berbobot untuk setiap kondisi yang tidak diinginkan:

$$\text{Objective} = w_{s1} \sum_i is_s1_i + w_{s8} \sum_i is_s8_i + w_{free_day} \sum_c no_free_day_c + \sum_i dev_hint_i$$

Karena seluruh hard constraints sudah dipastikan oleh CSP pada tahap feasibility, fungsi objektif tidak lagi memuat penalti untuk konflik dasar seperti tumpang-tindih dosen, cohort, atau ruang. Dengan kata lain, nilai objektif hanya berfokus pada aspek kualitas dan preferensi jadwal, bukan kelayakannya.

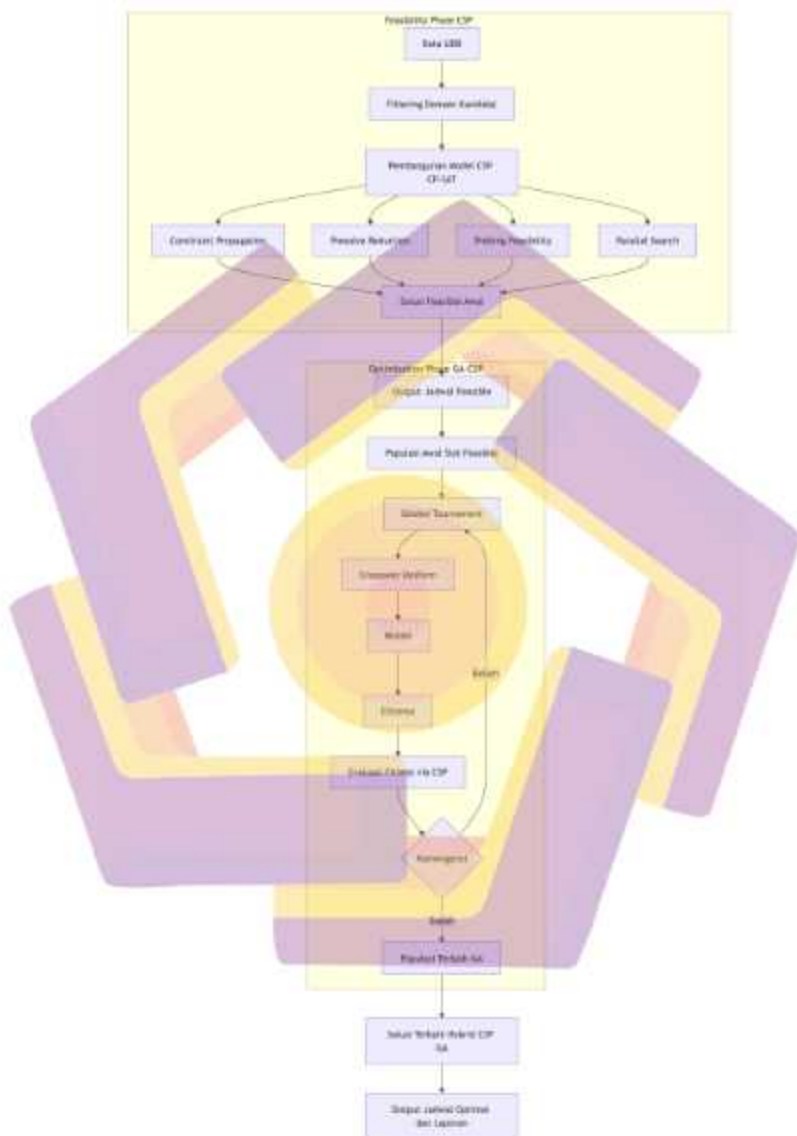
Area optimasi yang menjadi tujuan minimisasi mencakup empat komponen utama yaitu mengurangi penempatan sesi pada slot pertama, mengurangi penempatan sesi pada slot terakhir, memberikan minimal satu hari bebas untuk

setiap cohort, dan menjaga agar solusi yang dihasilkan CSP tidak terlalu jauh dari solution hint yang disarankan GA.

Melalui penggabungan penalti ini, proses optimasi diarahkan untuk menghasilkan jadwal yang tidak hanya sah menurut seluruh hard constraints, tetapi juga selaras dengan pola pengajaran yang lebih manusiawi, terdistribusi, dan stabil dalam konteks arsitektur hybrid CSP-GA.

3.6. Arsitektur Dual-Phase CSP-GA

Arsitektur metode yang digunakan dalam penelitian ini mengikuti pendekatan dual-phase, yaitu pemisahan proses menjadi dua tahap utama: fase feasibility dan fase optimization. Pemisahan ini bertujuan memastikan bahwa pencarian solusi berlangsung secara efisien dan stabil, dengan CSP berperan sebagai feasibility engine dan GA berperan sebagai optimization engine. Dengan desain seperti ini, seluruh solusi yang dievaluasi GA dijamin tetap berada dalam ruang solusi yang sah, sementara CSP menjaga konsistensi dan validitas constraint pada setiap iterasi optimasi.



Gambar 5 Diagram Arsitektur Hybrid CSP GA

Diagram pada gambar 5 menunjukkan alur lengkap arsitektur hybrid CSP-GA dalam dua fase yang saling terhubung. Fase pertama berfokus pada pembentukan model constraint secara formal dengan menggunakan CP-SAT. Pada tahap ini, seluruh data yang telah difilter digunakan untuk membangun variabel interval dan struktur NoOverlap untuk dosen, cohort, dan ruang. Solver kemudian menjalankan serangkaian proses internal, seperti presolve reduction, constraint propagation, probing, dan parallel search, untuk menghilangkan nilai domain yang tidak valid dan menemukan jadwal yang bebas dari seluruh hard constraints. Hasil dari fase ini adalah sebuah jadwal feasible awal serta domain slot yang telah disederhanakan, yang akan menjadi dasar bagi GA dalam tahap optimasi.

Fase kedua merupakan proses optimasi multi-kriteria yang dilakukan dengan menerapkan Genetic Algorithm. GA mengeksplorasi berbagai kemungkinan penempatan slot dengan memanfaatkan domain feasible dari CSP. Setiap individu hasil evolusi GA dikirim kembali ke CSP sebagai solution hint, sehingga CSP dapat memvalidasi kembali struktur jadwal serta menghitung nilai objektif berdasarkan soft constraints. Proses ini menjadikan CSP sebagai evaluator sekaligus mekanisme perbaikan terhadap individu GA. Rangkaian evolusi—seleksi, crossover, mutasi, dan elitisme—berlangsung secara berulang hingga terpenuhi kondisi konvergensi. Hasil akhirnya adalah konfigurasi slot terbaik yang meminimalkan penalti, sekaligus tetap memenuhi seluruh hard constraints yang ditetapkan.

Pendekatan dual-phase ini memastikan bahwa pencarian solusi berjalan secara terarah: CSP mengelola bagian validitas, sedangkan GA mengelola kualitas.

Interaksi keduanya menghasilkan solusi hybrid yang stabil, optimal, dan sesuai dengan tujuan penjadwalan akademik.

3.7. Dampak CSP Terhadap Proses Pencarian GA

Integrasi antara CSP dan GA dalam penelitian ini tidak bersifat sekadar menggabungkan dua metode, tetapi membentuk hubungan sebab-akibat yang memengaruhi arah pencarian GA secara langsung. Peran CSP pada fase awal membuat domain slot untuk setiap sesi menjadi jauh lebih sempit dan terarah, sehingga GA hanya bereksplorasi pada kombinasi waktu yang secara struktural memungkinkan. Penyempitan domain ini tidak hanya mengurangi ruang pencarian, tetapi juga menghilangkan potensi solusi yang sejak awal sudah tidak dapat dipenuhi oleh constraint, sehingga setiap mutasi maupun crossover selalu menghasilkan individu yang tetap berada dalam batas-batas kelayakan jadwal.

Selain membentuk domain pencarian, CSP juga memberikan jaminan bahwa tidak ada solusi yang melanggar hard constraints selama proses evolusi berlangsung. GA tidak perlu menangani penalti untuk konflik dasar seperti benturan dosen atau ruang, karena CSP memastikan seluruh individu yang dikirim kembali untuk evaluasi tetap memenuhi aturan tersebut. Mekanisme evaluasi fitness berbasis CSP membuat nilai objektif menjadi lebih stabil, karena fitness dihitung dari jadwal yang telah divalidasi dan diperbaiki melalui propagasi constraint, bukan sekadar perhitungan heuristik.

Di sisi lain, penggunaan solution hint menciptakan efek stabilisasi terhadap proses evolusi. CSP memperbaiki konfigurasi slot yang dihasilkan GA apabila

terjadi deviasi yang tidak konsisten, sehingga GA terhindar dari kondisi terjebak pada pola pencarian yang buruk. Interaksi ini memungkinkan GA mempertahankan arah eksplorasi yang konstruktif dan menghindari area solusi yang tidak menjanjikan.

Secara keseluruhan, kontribusi CSP terhadap GA membentuk pencarian yang lebih terarah, konsisten, dan efisien. Efek gabungan berupa penyempitan domain, jaminan feasibility, stabilisasi fitness, dan perbaikan hint menjadikan arsitektur hybrid CSP-GA lebih efektif dibandingkan penggunaan GA saja maupun CSP saja.

3.8. Perbedaan Metode Penelitian dengan Studi Terdahulu

Pendekatan yang digunakan dalam penelitian ini memiliki perbedaan mendasar dibandingkan studi terdahulu, khususnya penelitian Putra (2024) yang menjadi salah satu referensi utama pada bidang penjadwalan berbasis hybrid. Pada studi tersebut, integrasi antara GA dan CSP dilakukan secara langsung tanpa adanya pemisahan fase yang jelas. Proses optimasi berlangsung dalam satu jalur alur kerja, di mana GA menghasilkan solusi dan CSP digunakan untuk memeriksa kelayakan secara terbatas. Selain itu, penelitian tersebut menggunakan dataset relatif kecil—hanya 93 sesi—dan belum memanfaatkan kemampuan solver modern seperti CP-SAT, sehingga ruang pencarian masih lebih luas dan proses validasi tidak sepenuhnya komprehensif.

Penelitian ini mengusulkan arsitektur yang lebih sistematis melalui pendekatan dual-phase, yaitu pemisahan antara fase feasibility dan optimasi. Pada fase pertama, CP-SAT digunakan sebagai feasibility engine yang menjalankan

domain filtering dan propagasi constraint secara intensif untuk menghasilkan jadwal awal yang sepenuhnya valid. Pada fase kedua, GA bertindak sebagai optimization engine yang mengeksplorasi ruang solusi berdasarkan domain yang telah dipersempit CSP, serta menggunakan solution hint agar evaluasi selalu konsisten dengan constraint formal. Integrasi kedua fase ini menciptakan mekanisme umpan balik yang lebih stabil dan efisien dibandingkan pendekatan integrasi langsung.

Selain perbedaan arsitektural, Penelitian ini juga menyertakan protokol eksperimen yang lebih lengkap, mulai dari baseline CSP, intensitas GA, batas waktu GA, hingga evaluasi robustitas seed, sehingga memberikan gambaran performa yang lebih komprehensif. Dengan demikian, penelitian ini menawarkan pendekatan yang lebih modern, skalabel, dan terstruktur dibandingkan studi sebelumnya.

3.9. Protokol Eksperimen

Untuk mengevaluasi kinerja arsitektur hybrid CSP-GA, penelitian ini menggunakan empat rangkaian eksperimen yang dirancang untuk menguji aspek ketercapaian feasibility, kualitas optimasi, sensitivitas parameter, dan stabilitas metode. Tiap set eksperimen menggunakan dataset yang sama sehingga perbandingan dapat dilakukan secara konsisten dan terukur.

Tabel 3 Rangkuman Set Eksperimen

No	Set	Tujuan Eksperimen	Variasi Parameter	Contoh Kasus
1	A — CSP Only Baseline	Mengukur kemampuan CSP murni menghasilkan solusi feasible dan kualitas preferensi dasar	Bobot soft constraints & limit waktu CSP	A1, A2, A3
2	B — Intensitas GA	Menguji pengaruh ukuran populasi, jumlah generasi, dan waktu GA terhadap kualitas objektif	ga_pop, ga_gen, ga_time_limit, ga_patience	B1, B2, B3
3	C — Sensitivitas Waktu GA	Menilai apakah batas waktu evaluasi individu GA memengaruhi kualitas konvergensi	ga_time_limit (4-8 detik)	C1, C2, C3
4	D — Robustness Seed	Menilai kestabilan metode terhadap variasi acak GA	ga_seed (11, 42, 77)	D1, D2, D3

Pada tabel 3 di jelaskan Set A (CSP-only) digunakan untuk membangun baseline, karena CSP adalah fondasi utama dalam arsitektur dual-phase. Perubahan bobot dan batas waktu menguji bagaimana kemampuan solver menangani preferensi jadwal tanpa bantuan GA. Set B (Intensitas GA) mengevaluasi kontribusi GA sebagai optimization engine. Variasi populasi dan generasi memberikan gambaran seberapa besar eksplorasi yang dibutuhkan untuk menurunkan penalti secara signifikan. Set C (Time Limit GA) menguji sensitivitas performa terhadap waktu yang dialokasikan untuk setiap evaluasi fitness melalui CSP. Parameter ini penting karena CSP-lah yang menentukan stabilitas fitness. Set D (Seed Robustness) mengukur konsistensi metode yang bersifat stokastik. Stabilitas terhadap seed diperlukan untuk menunjukkan robustness dan reproductibility.

3.10. Implementasi Sistem

Implementasi sistem dilakukan menggunakan lingkungan pemrograman berbasis Python yang dipilih karena stabilitasnya dalam komputasi numerik dan ketersediaan library optimasi modern. Seluruh algoritma CSP dan GA dibangun secara terintegrasi dalam satu skrip utama untuk memastikan konsistensi data, reproduisibilitas percobaan, dan kemudahan pengendalian parameter eksperimen.

Proses penyelesaian hard constraints dijalankan menggunakan OR-Tools CP-SAT, sebuah solver hybrid modern yang menggabungkan teknik SAT solving, integer programming, dan constraint propagation. Library ini dipilih karena memiliki performa tinggi pada masalah penjadwalan berbasis interval serta mendukung fitur solution hint yang diperlukan untuk integrasi dengan GA. Untuk pengolahan data, NumPy dan Pandas digunakan dalam memuat, membersihkan, dan menyusun dataset sesi, dosen, cohort, serta ruang menjadi format yang sesuai untuk konstruksi variabel CSP. Seluruh pipeline eksperimen dijalankan menggunakan Python 3.10, yang menyediakan stabilitas dan kompatibilitas library yang memadai.

Selain komponen inti CSP dan GA, sistem juga menghasilkan beberapa keluaran pendukung untuk keperluan analisis hasil. Jadwal akhir ditgekspor ke format Excel untuk memudahkan validasi manual dan visualisasi oleh pihak program studi. Seluruh proses eksekusi direkam melalui log runtime, sehingga durasi presolve, desain CSP, evaluasi GA, serta total pemrosesan dapat dianalisis kembali pada Bab IV. Untuk menggambarkan dinamika pencarian pada fase optimasi, sistem juga menghasilkan grafik konvergensi, yang menampilkan

perubahan nilai objektif pada setiap generasi GA. Grafik ini membantu memberikan gambaran yang transparan mengenai stabilitas evolusi, titik jenuh, dan kualitas solusi akhir.

Dengan struktur implementasi seperti ini, seluruh komponen—mulai dari pemrosesan data, pembentukan model CSP, integrasi GA, hingga pengumpulan hasil eksperimen—dapat berjalan secara konsisten, terukur, dan mudah direplikasi.

3.11. Ringkasan Kontribusi Metode

Metode yang diusulkan dalam penelitian ini memberikan kontribusi yang berbeda pada masing-masing komponen utama—CSP, GA, dan integrasi keduanya—yang secara keseluruhan membentuk arsitektur dual-phase untuk penjadwalan mata kuliah. Pada fase pertama, kontribusi terletak pada penggunaan CP-SAT sebagai feasibility engine yang membangun model interval secara formal, mempersempit domain melalui filtering yang ketat, dan memastikan seluruh hard constraints terpenuhi secara deterministik. Pendekatan ini berbeda dari CSP klasik karena memanfaatkan presolve, propagation, probing, dan parallel search untuk mengurangi kompleksitas pencarian sejak awal.

Kontribusi GA berada pada kemampuannya mengeksplorasi variasi penempatan slot yang tetap berada dalam ruang solusi feasible. Dengan representasi kromosom berbasis slot-only, GA dapat berfokus sepenuhnya pada penurunan penalti soft constraints tanpa harus memeriksa konflik dasar. Operator seleksi, crossover, mutasi, dan elitisme digunakan untuk membentuk dinamika pencarian

yang stabil, sementara CSP mengambil alih peran sebagai evaluator agar nilai fitness selalu konsisten dan valid.

Kontribusi utama penelitian ini terletak pada integrasi keduanya dalam bentuk dual-phase hybrid, di mana CSP tidak hanya berfungsi sebagai pembangun dan penjaga validitas jadwal, tetapi juga sebagai corrector dan fitness assessor untuk GA melalui mekanisme solution hint. Arsitektur ini menghasilkan proses optimasi yang lebih terarah, mengurangi risiko eksplorasi yang tidak relevan, dan meningkatkan efisiensi pencarian dibandingkan metode GA-only maupun CSP-only. Pendekatan dual-phase ini memungkinkan penjadwalan dengan skala dataset yang lebih besar dan kompleks, serta menyediakan struktur evaluasi yang lebih stabil untuk eksperimen empiris pada Bab IV.



BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

4.1. Deskripsi Dataset dan Konfigurasi Eksperimen

Bagian ini menjelaskan dataset yang digunakan dalam penelitian serta konfigurasi parameter yang diterapkan pada solver CSP dan Algoritma Genetika. Penjelasan dibuat secara rinci agar seluruh proses pengujian dapat direplikasi dan dievaluasi secara transparan. Dataset yang digunakan merupakan data penjadwalan riil dari Program Studi di Universitas Duta Bangsa, sedangkan konfigurasi eksperimen mengikuti rancangan metodologi pada Bab III.

4.1.1 Dataset yang Digunakan

Dataset yang digunakan merupakan data riil penjadwalan Program Studi di lingkungan Universitas Duta Bangsa, terdiri atas 116 sesi perkuliahan, 17 cohort, 41 dosen, dan 19 ruang dengan variasi kapasitas serta tipe (LAB dan THEORY). Setiap sesi memiliki atribut durasi berdasarkan SKS, dosen pengampu, cohort yang mengikuti, serta kebutuhan tipe ruang, sehingga model mencerminkan kompleksitas penjadwalan akademik yang sebenarnya. Slot waktu disusun dalam struktur mingguan yang memuat beberapa hari kuliah aktif, dengan delapan slot per hari sesuai struktur jadwal institusi. Informasi tambahan seperti hari libur dosen, batas pengajaran pada slot terakhir, dan slot terblokir program studi juga dimasukkan ke dalam model sebagaimana digunakan pada tahap filtering domain di CSP.

Seluruh eksperimen dijalankan menggunakan parameter yang konsisten dengan desain metodologis pada Bab III. Jadwal akhir diekspor dalam format Excel untuk validasi serta dokumentasi, sementara waktu eksekusi dan nilai objektif dicatat dalam berkas log untuk analisis pada bagian-bagian berikutnya. Dengan lingkungan eksperimen yang distandarkan ini, seluruh hasil pada Bab IV dapat dianalisis secara objektif dan dapat diulang dengan reliabilitas yang tinggi.

Ringkasan karakteristik dataset disajikan pada Tabel 4.1.

Tabel 4 Karakteristik Dataset

Atribut	Nilai
Jumlah sesi	116
Jumlah cohort	17
Jumlah mata kuliah unik	19
Jumlah dosen	41
Jumlah ruang	19
Tipe ruang	THEORY 14, LAB 5
Kapasitas ruang (min, median, maks)	35, 40, 80
Slot waktu terblokir	Jumat S5-S6, Sabtu S7-S8

Tabel 5 merangkum karakteristik dataset yang digunakan pada penyusunan jadwal. Dataset memuat 116 sesi yang tersebar pada 17 cohort dan 19 mata kuliah unik dengan dukungan 41 dosen serta 19 ruang. Komposisi ruang didominasi tipe THEORY berjumlah 14 dan LAB berjumlah 5 sehingga kebutuhan praktikum cenderung lebih kompetitif dibanding kelas teori. Kapasitas ruang bervariasi dari 35 hingga 80 dengan nilai tengah 40 yang menandakan distribusi kapasitas relatif moderat namun tetap memerlukan pemetaan ruang yang cermat untuk cohort berukuran besar. Terdapat slot waktu terblokir pada Jumat S5-S6 dan Sabtu S7-S8 yang mengurangi horizon penjadwalan efektif sehingga kepadatan penempatan sesi

pada hari lain meningkat. Kombinasi jumlah sesi yang cukup tinggi, persebaran tipe ruang yang tidak seimbang, serta pemblokiran slot tertentu membuat problem ini relevan untuk pendekatan hybrid di mana CSP menjaga feasibility pada tiga dimensi lecturer cohort dan room sementara GA mengoptimalkan kualitas terhadap preferensi soft constraints.

Tabel 5 Pembagian slot waktu

Slot	Jam mulai	Jam selesai
S1	08.00	08.50
S2	08.50	09.40
S3	09.40	10.30
S4	10.30	11.20
S5	11.20	12.10
S6	12.10	13.00
S7	13.00	13.50
S8	13.50	14.40

Tabel 6 mendefinisikan grid waktu harian yang terdiri atas delapan slot berurutan dengan durasi lima puluh menit per slot. S1 dimulai pukul 08.00 dan berakhir 08.50 diikuti S2 08.50–09.40 kemudian S3 09.40–10.30 berlanjut sampai S8 13.50–14.40. Penetapan interval yang seragam memudahkan pemodelan durasi perkuliahan sehingga beban satu SKS menempati satu slot dua SKS dua slot berurutan dan tiga SKS tiga slot berurutan tanpa melintasi hari. Grid ini digunakan langsung pada pemodelan CSP untuk membentuk domain start serta penegakan AddNoOverlap pada tiga dimensi lecturer cohort dan room. Pada evaluasi kualitas referensi S1 dan S8 dari tabel ini menjadi acuan perhitungan penalti early morning session dan late session sedangkan deteksi free day per cohort dihitung dari akumulasi slot terisi per hari. Apabila terdapat jeda institusional di tengah hari

penyesuaian dapat dilakukan dengan menandai slot tertentu sebagai blocked slots tanpa mengubah definisi dasar satu slot setara lima puluh menit.

Tantangan utama muncul dari tumpang tindih ruang dan waktu yang harus dihindari melalui constraint CSP, ketersediaan dosen yang tidak seragam antar hari, distribusi beban cohort agar tidak terjadi sesi terlalu pagi atau terlalu sore secara berlebihan, kebutuhan ruang khusus (LAB) yang tidak dapat digantikan oleh ruang teori. Dataset ini juga digunakan untuk semua eksperimen A–D agar seluruh hasil dapat dibandingkan secara konsisten.

4.1.2 Parameter Algoritma Genetika (AG)

Metode yang digunakan mengombinasikan Constraint Satisfaction Problem dengan Algoritma Genetika, di mana CSP menjamin kelayakan dan GA menurunkan penalti soft constraint. Bobot soft constraint yang digunakan adalah $w_{s1} = 1$, $w_{s8} = 1$, dan $w_{free_day} = 8$. Parameter umum GA meliputi ukuran populasi, jumlah generasi, waktu evaluasi per individu, early stopping melalui patience, dan pemolehan solusi terbaik dengan CSP, variasi skenario yang dievaluasi dirangkum pada Tabel 5.

Tabel 6 Ringkasan Variasi Skenario GA dan Nilai Objektif

No	Skenario	Pop	Gen	Waktu per Individu (detik)	Patience	Batas Waktu CSP (detik)	Nilai Objektif
1	B1_GA_light	8	6	6	2	60	52.0
2	B2_GA_medium	10	8	8	2	90	44.0
3	B3_GA_heavy	12	10	10	3	120	42.0
4	C1_GA_t14	8	6	4	2	60	59.0
5	C2_GA_t16	8	6	6	2	60	48.0
6	C3_GA_t18	8	6	8	2	90	47.0

7	D1 GA seed11	10	8	8	2	90	45.0
8	D2 GA seed42	10	8	8	2	90	44.0
9	D3 GA seed77	10	8	8	2	90	44.0

Tabel 7 merangkum eksperimen variasi konfigurasi Genetic Algorithm beserta objective value yang diperoleh. Kelompok B menunjukkan bahwa peningkatan intensitas pencarian dari B1 pop delapan dan gen enam menjadi B3 pop dua belas dan gen sepuluh dengan batas waktu CSP seratus dua puluh detik menurunkan objective dari 52,0 menjadi 42,0 sehingga kualitas jadwal membaik. Kelompok C menyoroti pengaruh waktu evaluasi per individu dan batas waktu CSP pada konfigurasi ringan pop delapan dan gen enam. Pembatasan agresif pada C1 dengan empat detik dan batas enam puluh detik menghasilkan objective 59,0 sementara pelonggaran waktu pada C2 dan C3 memperbaiki hasil menjadi 48,0 dan 47,0 yang menunjukkan manfaat waktu evaluasi tambahan namun dengan perbaikan yang mulai melandai. Kelompok D menguji stabilitas terhadap seed acak pada konfigurasi menengah pop sepuluh dan gen delapan dengan hasil 45,0 sampai 44,0 yang menandakan variansi antarseed rendah dan proses relatif robust. Secara keseluruhan B3 memberi objective terendah 42,0 untuk kualitas terbaik, B2 menghasilkan 44,0 yang mendekati terbaik dengan biaya komputasi lebih hemat sehingga layak dijadikan konfigurasi default, sedangkan C2 dapat dipilih pada kondisi sumber daya ketat karena memberi keseimbangan yang wajar antara waktu evaluasi dan mutu jadwal. Indikator tambahan soft constraint pada beberapa skenario utama disajikan pada Tabel 6.

Tabel 7 Indikator Tambahan Soft Constraint

Skenario	Sesi Mulai S1	Sesi Mulai S8	Cohort Tanpa Hari Bebas
B1 GA light	4	0	0 dari 17
B2 GA medium	4	0	0 dari 17
B3 GA heavy	3	0	0 dari 17
C1 GA t14	12	0	0 dari 17
C2 GA t16	4	0	0 dari 17
C3 GA t18	5	0	0 dari 17
D1 GA seed11	3	0	0 dari 17
D2 GA seed42	4	0	0 dari 17
D3 GA seed77	4	0	0 dari 17

Tabel 8 menyajikan indikator tambahan untuk soft constraints yang dipakai pada evaluasi kualitas jadwal. Tiga metrik yang dilaporkan adalah jumlah sesi yang dimulai pada S1, jumlah sesi yang dimulai pada S8, dan jumlah cohort tanpa hari bebas. Seluruh skenario menunjukkan nol sesi pada S8 serta nol cohort tanpa hari bebas dari total tujuh belas cohort sehingga dua preferensi utama telah terpenuhi konsisten. Perbedaan antar skenario terutama terlihat pada sesi yang dimulai pada S1. Konfigurasi ringan dengan batas evaluasi pendek pada C1 menghasilkan dua belas sesi S1 yang relatif tinggi. Pelonggaran waktu evaluasi pada C2 dan C3 menurunkan jumlah tersebut menjadi empat dan lima, sedangkan peningkatan intensitas pencarian pada kelompok B dari B1 ke B3 menurunkan nilai S1 dari empat menjadi tiga yang selaras dengan penurunan objective pada Tabel 4.2. Variasi antarseed pada kelompok D berada di kisaran tiga hingga empat yang menandakan stabilitas proses. Secara keseluruhan temuan ini memperlihatkan bahwa peningkatan kapasitas pencarian atau waktu evaluasi membantu menekan penalti early morning session sambil tetap menjaga nihil S8 dan terpenuhinya free

day pada seluruh cohort, sehingga konfigurasi menengah hingga berat menjadi pilihan yang lebih seimbang antara mutu dan biaya komputasi.

4.2. Proses Algoritma

Bagian ini menjelaskan alur eksekusi metode hybrid CSP-GA yang digunakan dalam penelitian. Proses algoritma mengikuti arsitektur dual-phase yang telah dijelaskan pada Bab III, dengan CSP bertindak sebagai feasibility engine dan GA sebagai optimization engine. Setiap eksperimen pada Bab IV menggunakan alur yang sama, sehingga penjelasan pada bagian ini menjadi fondasi interpretasi hasil-hasil berikutnya.

4.2.1 Alur Umum

Proses algoritmik pada penelitian ini mengikuti arsitektur dual-phase yang memisahkan secara jelas antara tahap pemenuhan hard constraints dan tahap optimasi soft constraints. Alur umum ini memastikan bahwa seluruh solusi yang dievaluasi pada fase optimasi telah lolos dari tahap validasi struktur, sehingga evolusi dapat berlangsung secara efisien. Secara umum, proses dijalankan sebagai berikut:

Tahap pertama dimulai dengan memuat dataset penjadwalan yang berisi komponen-komponen utama: daftar sesi, dosen pengampu, cohort, jenis ruang, kapasitas ruang, dan struktur slot mingguan. Informasi tambahan seperti hari libur dosen, slot terblokir, dan batas pengajaran pada slot terakhir juga digunakan untuk membentuk constraint yang lebih realistis. Dataset ini kemudian diproses melalui tahap filtrasi domain, yaitu penyaringan awal terhadap kandidat slot dan ruang yang

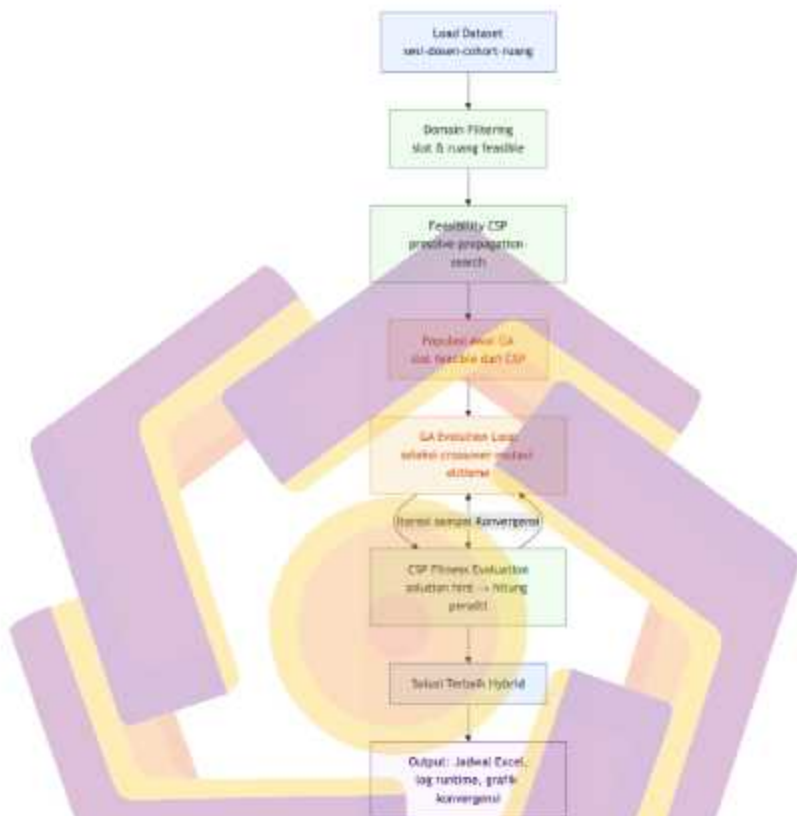
mungkin digunakan oleh setiap sesi. Pada tahap ini, sesi hanya diberi domain yang sesuai dengan tipe ruang, kapasitas cohort, preferensi dosen, dan aturan institusional, sehingga ruang pencarian awal sudah jauh dipersempit sebelum solver dijalankan.

Setelah domain kandidat terbentuk, tahap feasibility dijalankan menggunakan solver CP-SAT untuk menghasilkan solusi awal yang memenuhi seluruh hard constraints. CP-SAT membangun model penjadwalan berbasis interval, melakukan presolve dan propagasi, dan kemudian mencari solusi feasible. Jika ditemukan, solusi ini berfungsi sebagai patokan struktural, karena setiap penempatan slot yang tercantum di dalamnya dapat dipandang sebagai representasi zona-feasible untuk pengembangan lebih lanjut oleh GA. Pada tahap ini juga dihasilkan domain feasible final yang lebih matang dibanding domain awal sebelum CSP, karena seluruh kombinasi slot-ruang yang tidak konsisten telah dieliminasi oleh solver.

Tahap berikutnya memasuki fase optimasi, yaitu pembentukan populasi awal GA. Populasi GA tidak dibentuk secara acak penuh, melainkan berdasarkan slot-slot feasible hasil CSP, sehingga setiap gen sudah berada dalam domain yang valid. Evolusi GA kemudian dijalankan melalui operator seleksi, crossover, mutasi, dan elitisme. Seluruh individu yang dihasilkan GA tidak langsung dievaluasi oleh fungsi fitness internal, tetapi dikirim ke CSP kembali sebagai solution hint. Dengan mekanisme ini, CSP membangun ulang jadwal berdasarkan kromosom GA, memperbaiki inkonsistensi kecil jika muncul, dan menghitung penalti soft constraints secara akurat.

Proses evolusi berlangsung secara iteratif hingga mencapai kondisi konvergensi yang ditetapkan—baik stagnasi nilai fitness, batas generasi, atau batas waktu GA. Setiap iterasi memberikan umpan balik pada populasi berikutnya, dan pada akhirnya diperoleh solusi dengan nilai penalti terendah. Seluruh proses juga menghasilkan artefak keluaran berupa jadwal akhir, log runtime, nilai objektif tiap generasi, dan grafik konvergensi yang akan dianalisis pada bagian-bagian berikutnya dalam Bab IV.

Karena CSP bersifat deterministik, sedangkan GA bersifat stokastik, maka tiga aspek menjadi fokus analisis hasil. Pertama adalah Konvergensi (apakah GA stabil mendekati nilai minimum), selanjutnya Stabilitas seed (apakah hasil berubah signifikan jika seed berubah), terakhir Runtime total (interaksi CSP-GA dalam setiap generasi). Ketiga aspek tersebut menjadi dasar pengukuran efektivitas arsitektur hybrid pada eksperimen selanjutnya.



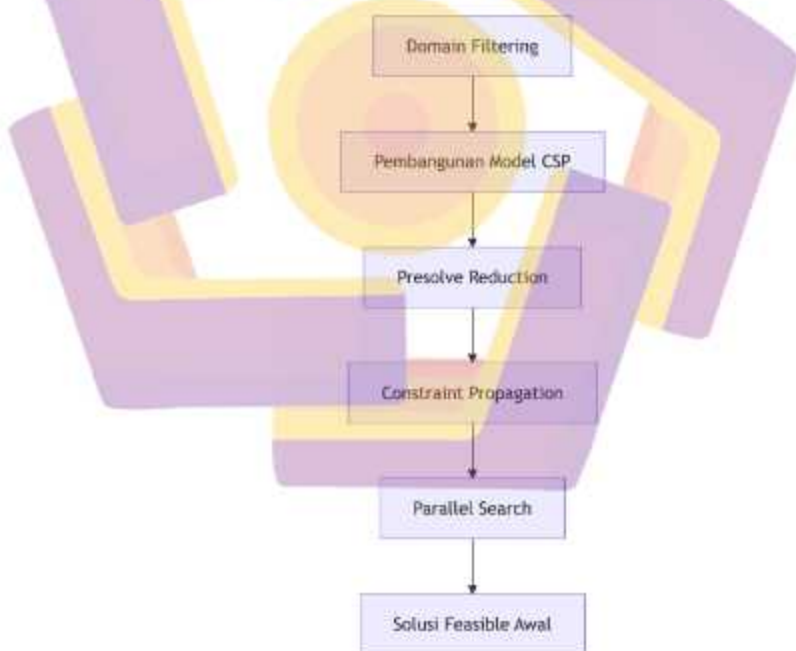
Gambar 6 Alur Eksekusi Metode Hybrid CSP-GA

Diagram ini menunjukkan alur eksekusi metode hybrid pada tahap eksperimen. Dataset yang berisi informasi sesi, dosen, cohort, dan ruang terlebih dahulu difilter untuk membentuk domain slot yang feasible. CSP kemudian menghasilkan solusi awal yang bebas konflik dan menentukan domain feasible final. Domain ini digunakan GA untuk membentuk populasi awal. Pada setiap iterasi evolusi, solusi GA dikirim kembali ke CSP sebagai solution hint untuk diverifikasi dan dinilai berdasarkan penalti soft constraints. Proses berlangsung

hingga kondisi konvergensi tercapai, kemudian dihasilkan solusi terbaik beserta artefak keluaran berupa jadwal Excel, log runtime, dan grafik konvergensi.

4.2.2 Tahap CSP sebagai feasibility engine

Tahap CSP berfungsi untuk memastikan bahwa seluruh sesi dapat ditempatkan pada slot dan ruang yang konsisten dengan hard constraints. Pada tahap ini, OR-Tools CP-SAT menjalankan rangkaian proses penyelesaian yang tidak hanya menentukan penempatan awal setiap sesi, tetapi juga menyempurnakan domain solusi sebelum tahap evolusi GA dilakukan.



Gambar 7 Bagaimana CSP dieksekusi pada eksperimen

Pada gambar 7 merupakan Pipeline eksekusi CSP, proses dimulai dengan membangun interval variabel untuk setiap sesi berdasarkan dataset dan domain yang telah difilter. Selanjutnya solver menjalankan mekanisme presolve reduction untuk menghapus nilai domain yang tidak mungkin, diikuti constraint propagation untuk mempersempit kandidat slot dan ruang secara otomatis berdasarkan interaksi antar constraint. Pada penelitian ini, tahapan presolve memberikan kontribusi signifikan karena banyak kombinasi slot-ruang dieliminasi sebelum pencarian dilakukan.

Setelah presolve selesai, CP-SAT memasuki tahap parallel search yang melibatkan beberapa worker untuk mengeksplorasi ruang solusi secara simultan. Solver mengevaluasi rantai konflik, mempelajari pola ketidakcocokan jadwal, dan menyesuaikan pencarian agar lebih efisien. Proses ini selalu menghasilkan jadwal feasible pada setiap eksperimen A-D, sehingga CSP menjadi fondasi deterministik yang stabil bagi tahap optimasi GA.

Output dari tahap ini berupa, satu jadwal feasible awal, domain slot feasible yang telah dipersempit, struktur constraint siap digunakan GA, log runtime presolve dan propagasi. Tahap CSP ini memastikan bahwa evolusi GA pada fase berikutnya tidak menghabiskan waktu pada solusi invalid, sehingga setiap individu GA yang dievaluasi selalu mematuhi seluruh hard constraints.

4.2.3 Tahap GA sebagai optimization engine

Tahap GA merupakan fase optimasi setelah CSP memastikan solusi feasible terhadap seluruh hard constraints. Pada penelitian ini, GA tidak digunakan untuk menemukan jadwal yang valid, tetapi untuk meningkatkan kualitas jadwal melalui penurunan penalti soft constraints. Seluruh proses GA berjalan di atas domain slot yang telah direduksi oleh CSP, sehingga setiap individu GA selalu feasible secara struktural.

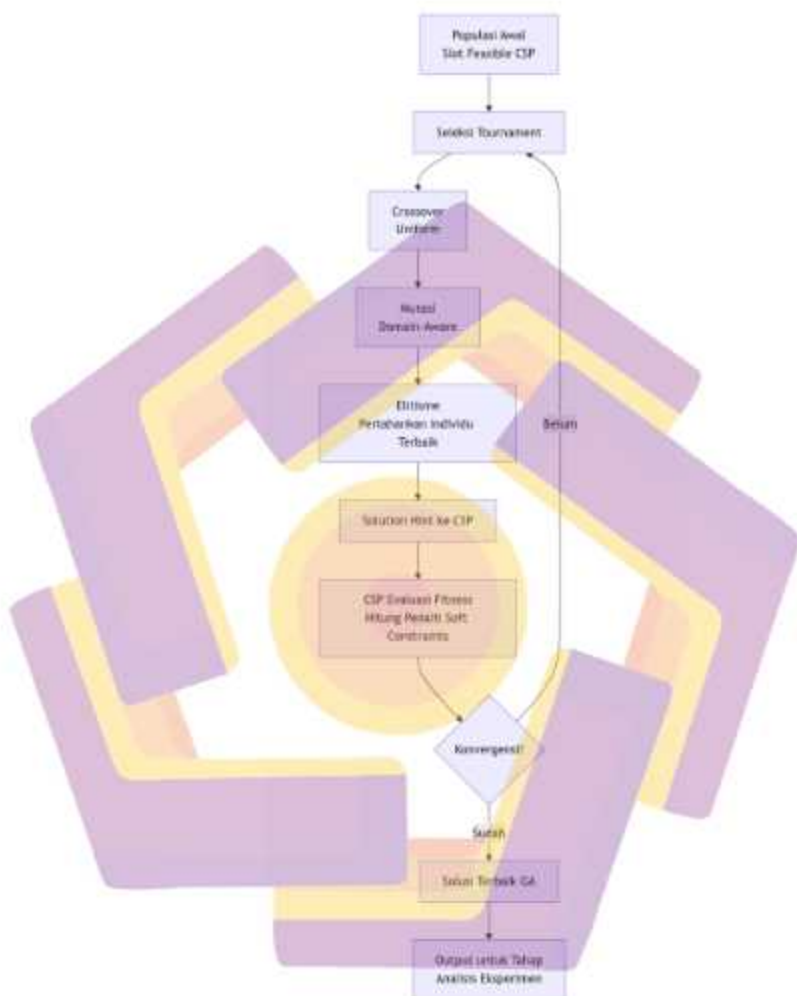
Proses dimulai dengan pembentukan populasi awal GA yang seluruhnya diambil dari slot-slot feasible hasil CSP. Pendekatan ini mempersempit ruang pencarian secara signifikan dan menghindari pemborosan generasi pada solusi yang melanggar constraint dasar. Pada eksperimen B1–B3 dan C1–C3, seluruh individu awal GA terbukti feasible 100%, sebagaimana ditunjukkan oleh rasio Feasible/Pop berikut, B1 (GA-light): 8/8 feasible, B2 (GA-medium): 10/10 feasible, B3 (GA-heavy): 12/12 feasible, C3 (GA-tl8): 8/8 feasible, C1 (GA-tl4): 0/8 feasible. Hal ini menegaskan bahwa CSP-domain filtering bekerja efektif, dan GA tidak menghabiskan waktu pada solusi tidak valid—sebuah kontribusi penting dari pendekatan hybrid.

Setiap generasi GA terdiri atas empat operator evolusioner seperti seleksi tournament, uniform crossover, mutasi domain-aware, dan elitisme. Tournament selection terbukti menghasilkan populasi yang stabil, sedangkan uniform crossover mampu menggabungkan blok-blok slot dari kedua parent tanpa memecah interval sesi. Mutasi dilakukan secara terbatas pada domain slot feasible sehingga tidak pernah menciptakan konflik dosen/cohort/ruang. Elitisme memastikan bahwa

individu terbaik dari generasi sebelumnya tidak hilang, yang terlihat pada kurva konvergensi seluruh eksperimen.

Hal yang membedakan penelitian ini dari GA-only tradisional adalah mekanisme CSP-based fitness evaluation. Pada setiap generasi Individu GA diubah menjadi hint slot, Hint dikirim ke solver CP-SAT, CSP membangun jadwal berdasarkan hint tersebut, CSP menghitung penalti soft constraints secara akurat (S1, S8, free-day, deviasi hint). Pendekatan ini memastikan fitness bersifat deterministik walaupun GA bersifat stokastik. Dampaknya terlihat jelas pada hasil eksperimen GA-light (B1) mencapai penurunan penalti hingga 52 (rekonstruksi), GA-medium (B2) mencapai 44. GA-heavy (B3) mencapai nilai penalti terkecil 42, menunjukkan arah perbaikan kualitas yang konsisten dengan peningkatan intensitas GA.

Proses evolusi berlangsung hingga kondisi konvergensi tercapai. Pada eksperimen Anda, seluruh konfigurasi GA memiliki durasi generasi yang stabil dengan pola berikut GA-light: ~50,77 detik/generasi, GA-medium: ~83,85 detik/generasi, GA-heavy: ~124,99 detik/generasi, GA-tl8: ~66,81 detik/generasi, GA-tl4: ~27,83 detik/generasi (kasus gagal, fitness = $1e12$). Perbedaan waktu ini sepenuhnya berasal dari batas waktu evaluasi CSP per individu (`ga_time_limit`). Seluruh proses GA menghasilkan artefak penting berupa grafik konvergensi, tabel fitness per generasi, dan jadwal optimal final.



Gambar 8 Alur Eksekusi GA sebagai Optimization Engine

Diagram ini menggambarkan proses evolusi GA pada tahap optimasi. Populasi awal dibentuk dari slot-slot yang telah diverifikasi feasible oleh CSP. Setiap generasi terdiri atas seleksi, crossover, mutasi, dan elitisme. Individu hasil

evolusi dikirim ke CSP sebagai solution hint, kemudian CSP melakukan evaluasi penalti soft constraints untuk menghasilkan nilai fitness. Proses ini berulang hingga mencapai kondisi konvergensi dan menghasilkan solusi terbaik untuk dianalisis lebih lanjut pada hasil eksperimen.

4.2.4 Objective function dan definisi fitness

Tahap evaluasi fitness merupakan komponen kunci pada proses optimasi, karena nilai inilah yang menentukan kualitas setiap individu GA pada setiap generasi. Pada penelitian ini, fitness tidak dihitung langsung oleh GA, tetapi melalui proses evaluasi formal menggunakan CSP. Dengan pendekatan ini, nilai fitness selalu konsisten dengan constraint jadwal dan tidak terpengaruh oleh sifat stokastik GA. Integrasi ini menghasilkan fitness landscape yang stabil, sehingga proses konvergensi dapat dianalisis secara transparan pada bagian hasil eksperimen.

Fungsi objektif yang digunakan dalam penelitian ini merupakan penjumlahan penalti berbobot yang berasal dari pelanggaran soft constraints. Setiap kategori penalti diwakili oleh variabel biner yang bernilai satu apabila kondisi yang tidak diinginkan terjadi. CSP kemudian menghitung total penalti dengan mengalikan variabel biner tersebut dengan bobot penalti yang ditetapkan. Bobot penalti disesuaikan dengan preferensi operasional: penempatan sesi pada slot pertama dan slot terakhir diberi penalti untuk mengurangi beban kuliah ekstrem, cohort yang tidak memiliki hari bebas diberi penalti untuk menjaga keberlanjutan ritme belajar, dan penalti deviasi digunakan untuk menjaga kedekatan solusi terhadap solution hint dari GA.

Evaluasi fitness dilakukan dengan langkah berikut, Kromosom GA diterjemahkan menjadi solution hint, yaitu rekomendasi slot awal untuk semua sesi. CP-SAT membangun jadwal berdasarkan hint tersebut dan memastikan bahwa seluruh hard constraints tetap terpenuhi. CSP menghitung jumlah pelanggaran soft constraints (misalnya sesi pada slot pertama, sesi pada slot terakhir, jumlah cohort tanpa free-day, atau deviasi jadwal terhadap hint). Total penalti ini menjadi nilai fitness untuk kromosom tersebut. Pendekatan ini menghasilkan fitness yang dapat diinterpretasikan secara langsung, karena setiap penurunan nilai penalti mencerminkan perbaikan kualitas jadwal. Pada seluruh eksperimen B dan C, nilai fitness menunjukkan perilaku yang konsisten dengan mekanisme evaluasi ini. Misalnya, pada B1 (GA-light), nilai fitness turun secara bertahap dari fase awal hingga generasi akhir, sebelum mencapai titik konvergensi. Hal serupa terlihat pada B2 dan B3, di mana generasi awal memiliki penalti yang lebih tinggi dan kemudian menurun menjadi nilai minimum yang stabil.

Selain memengaruhi kualitas solusi, fungsi objektif juga memengaruhi dinamika waktu komputasi. Setiap evaluasi fitness memerlukan eksekusi CSP dengan batas waktu tertentu, sehingga parameter `ga_time_limit` berperan langsung dalam lamanya setiap generasi GA. Hal ini dibahas lebih lanjut pada analisis eksperimen di bagian 4.3, terutama pada eksperimen C1-C3 yang menguji sensitivitas batas waktu CSP terhadap kualitas fitness.

Dengan demikian, definisi fitness dalam penelitian ini bukan hanya fungsi matematis, tetapi juga mekanisme integrasi CSP-GA yang menjamin bahwa semua solusi yang dievaluasi selalu feasible terhadap hard constraints, sekaligus

memungkinkan optimasi multi-kriteria pada soft constraints secara konsisten dan terukur.

4.2.5 Integrasi CSP dan GA

Integrasi antara CSP dan GA merupakan inti dari arsitektur hybrid yang digunakan dalam penelitian ini. Pada tahap ini, CSP tidak hanya berfungsi sebagai penyusun jadwal feasible, melainkan juga sebagai evaluator dan mekanisme koreksi selama proses evolusi GA. Integrasi ini memungkinkan eksplorasi solusi secara efisien, karena GA selalu bekerja di dalam ruang solusi yang telah disederhanakan dan dipastikan valid oleh CSP.

Proses integrasi dimulai ketika hasil CSP pada fase awal digunakan untuk membentuk domain feasible bagi GA. CSP menghilangkan slot dan ruang yang tidak memenuhi constraint, sehingga GA tidak lagi perlu menangani isu kelayakan dasar seperti benturan dosen, konflik cohort, atau penggunaan ruang yang tidak sesuai. Pereduksian domain ini terbukti menghasilkan populasi GA yang seluruhnya feasible pada banyak eksperimen (misalnya B1, B2, B3, dan C3), sehingga proses evolusi dapat berfokus sepenuhnya pada optimasi penalti soft constraints.

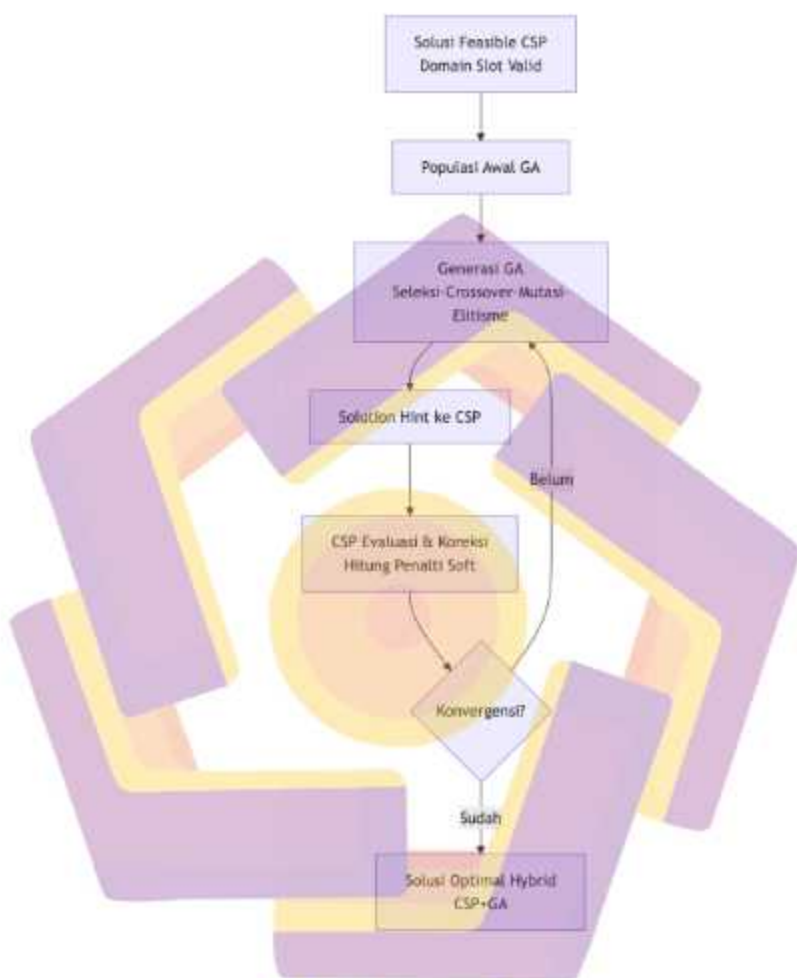
Selama evolusi, setiap individu GA dikirim ke CSP dalam bentuk solution hint. CSP menerima kromosom ini sebagai rekomendasi awal, membangun ulang jadwal berdasarkan nilai slot yang diberikan, dan kemudian menghitung penalti soft constraints secara akurat. Mekanisme ini mengubah CSP menjadi fitness evaluation engine, sehingga fitness tidak pernah dihitung secara heuristik atau aproksimatif.

Dengan pendekatan ini, nilai objektif seluruh eksperimen merepresentasikan kualitas jadwal yang valid, bukan hasil manipulasi lokal GA. Hal ini penting untuk menjaga interpretasi hasil pada Bab IV tetap kuat dan konsisten.

Interaksi iteratif ini juga berfungsi sebagai mekanisme perbaikan struktural. Ketika GA menghasilkan individu yang berpotensi menimbulkan ketidaksesuaian lokal (misalnya dua sesi yang berada pada slot berdekatan tetapi tidak optimal secara operasional), CSP dapat memperbaikinya selama proses konstruksi jadwal sebelum menghitung fitness. Efek ini dikenal sebagai *hint stabilization effect*, yang terbukti mencegah GA mengalami stagnasi atau eksplorasi di area solusi yang tidak produktif. Fenomena ini terlihat dari pola konvergensi yang lebih stabil pada eksperimen B2, B3, dan C3.

Proses integrasi berlangsung hingga GA mencapai kondisi konvergensi, yang ditandai dengan stagnasi penalti atau batas generasi. Dengan demikian, CSP menjamin validitas jadwal, sedangkan GA meningkatkan kualitasnya. Sinergi ini menghasilkan pipeline optimasi yang mampu memberikan solusi terbaik tanpa mengorbankan kepatuhan terhadap hard constraints.

Alur eksekusi integrasi dapat dilihat pada diagram berikut.



Gambar 9 Mekanisme Integrasi CSP dan GA.

Diagram ini menunjukkan integrasi dua arah antara CSP dan GA selama proses eksperimen. CSP menyediakan domain slot feasible dan solusi awal yang bebas konflik, sementara GA mengeksplorasi perbaikan kualitas solusi melalui proses evolusi. CSP menerima setiap individu GA sebagai solution hint,

membangun ulang jadwal, memperbaiki inkonsistensi, dan menghitung penalti sebagai nilai fitness. Siklus ini berulang hingga kondisi konvergensi terpenuhi.

4.2.6 Implementasi dan orkestrasi

Implementasi metode hybrid CSP-GA dilaksanakan melalui satu skrip utama yang mengoordinasikan seluruh tahapan eksekusi, mulai dari pemuatan data, konstruksi model CSP, proses evolusi GA, hingga penyimpanan artefak hasil eksperimen. Orkestrasi ini dirancang agar setiap eksperimen dapat direplikasi secara konsisten dengan parameter yang berbeda tanpa mengubah struktur kode inti. Dengan demikian, seluruh rangkaian percobaan A-D pada Bab IV dapat dilakukan secara otomatis, terstandarisasi, dan terdokumentasi.

Skrip utama yang digunakan adalah `time_table_scheduler.py`, yang berfungsi sebagai pengendali pipeline. Skrip ini memuat beberapa modul pendukung untuk membaca dataset, membangun struktur interval CSP, menghasilkan populasi GA, menjalankan operator evolusi, mengirim solution hint ke CSP, serta mencatat nilai fitness pada setiap generasi. Struktur kode memisahkan fungsi-fungsi utama (CSP modeling, domain filtering, GA operator, fitness evaluation, logging, dan ekspor hasil) agar proses debugging dan eksperimen menjadi lebih mudah.

Proses eksekusi dimulai dengan pemanggilan fungsi pemuatan dataset yang membaca jadwal sesi, dosen, cohort, dan ruang dari file spreadsheet. Data ini kemudian diubah menjadi representasi internal yang digunakan CSP dan GA.

Setelah itu, skrip membangun domain kandidat melalui filtrasi awal dan menjalankan CSP fase pertama untuk menghasilkan solusi feasibel serta domain slot final yang lebih tereduksi. Pada tahap ini, CP-SAT menghasilkan struktur constraint dan interval variabel yang akan digunakan ulang selama evaluasi fitness GA.

Tahap berikutnya adalah orkestrasi evolusi GA. Skrip membangun populasi awal berdasarkan domain feasible CSP, kemudian memasuki loop generasi yang mencakup seleksi, crossover, mutasi, elitisme, dan evaluasi fitness. Setiap iterasi mencatat nilai fitness, waktu eksekusi, serta perubahan terbaik (best-so-far). Seluruh data ini disimpan dalam format CSV atau Markdown agar dapat digunakan untuk analisis pada bagian 4.3 dan 4.4.

Skrip juga menjalankan mekanisme logging, yang mencatat seluruh parameter yang digunakan pada setiap percobaan, termasuk bobot penalti, batas waktu CSP, ukuran populasi, jumlah generasi, seed acak, dan ukuran domain awal. Fitur logging ini penting untuk memastikan transparansi eksperimen serta memudahkan validasi ulang apabila diperlukan. Hasil akhir setiap eksperimen berupa jadwal optimal, grafik konvergensi, log runtime, dan evaluasi penalti disimpan dalam folder terpisah sesuai nama kasus, sehingga dapat ditelusuri kembali secara independen.

Dengan desain orkestrasi ini, setiap eksperimen dapat dijalankan secara konsisten hanya dengan mengubah parameter pada fungsi `run_case`. Pendekatan ini tidak hanya meningkatkan efisiensi saat menjalankan banyak percobaan, tetapi juga

menjamin bahwa seluruh hasil eksperimen bersifat reproducible—sebuah aspek penting dalam penelitian berbasis algoritma evolusioner.

4.2.7 Artefak keluaran

Setiap eksekusi metode hybrid CSP-GA menghasilkan sejumlah artefak keluaran yang digunakan sebagai dasar analisis pada bagian 4.3 dan 4.4. Artefak ini mencakup keluaran utama berupa jadwal optimal, serta keluaran penunjang yang merekam dinamika proses CSP dan GA. Seluruh artefak disimpan secara otomatis dalam direktori khusus sesuai nama eksperimen, sehingga setiap kasus dapat dianalisis secara independen dan direplikasi apabila diperlukan.

Artefak utama yang dihasilkan adalah jadwal akhir dalam format Excel, yang memuat penempatan setiap sesi berdasarkan hari, ruang, dan slot waktu. Format ini memudahkan proses validasi manual oleh program studi dan menjadi representasi konkret dari solusi yang dihasilkan algoritma. Selain tabel jadwal, CSP juga menghasilkan log presolve dan informasi waktu penyelesaian yang mencerminkan kompleksitas proses pencarian solusi feasible.

Untuk mendukung analisis dinamika optimasi, sistem menghasilkan file konvergensi GA dalam format CSV, yang berisi nilai fitness setiap generasi serta waktu eksekusi per generasi. Data ini menjadi dasar pembuatan grafik konvergensi yang ditampilkan pada bagian 4.3.1. Selain itu, setiap eksperimen juga menyimpan laporan evaluasi penalti, termasuk jumlah sesi pada slot pertama/terakhir, jumlah cohort tanpa free-day, dan nilai deviasi hint, sehingga proses perbandingan kualitas antar eksperimen dapat dilakukan secara objektif.

Artefak tambahan meliputi log runtime lengkap, parameter konfigurasi eksperimen, dan salinan jadwal hasil refinemen CSP setelah GA. Dengan artefak-artefak ini, seluruh hasil eksperimen dalam penelitian dapat ditelusuri kembali secara sistematis serta memenuhi persyaratan transparansi dan reproduktibilitas penelitian berbasis algoritma evolusioner.

4.3. Hasil Eksperimen

4.3.1 Konvergensi Nilai Fitness

Bagian ini menyajikan analisis perilaku konvergensi Algoritma Genetika (GA) pada setiap skenario eksperimen. Konvergensi diukur berdasarkan perubahan nilai fitness (total penalti *soft constraints*) pada setiap generasi, dengan evaluasi fitness dilakukan menggunakan CSP melalui mekanisme *solution hint*. Analisis ini bertujuan menilai efektivitas GA dalam memperbaiki kualitas solusi serta mengidentifikasi hubungan antara intensitas GA, batas waktu evaluasi, dan stabilitas evolusi.

1. Konvergensi pada Skenario B (Variasi Beban GA: Light, Medium, Heavy)

Skenario B menguji bagaimana intensitas evolusi GA (populasi dan jumlah generasi) memengaruhi kecepatan dan kualitas konvergensi.



Gambar 10 Perbandingan Fitness Skenario B (Light–Medium–Heavy)

Gambar 10 menunjukkan tiga konfigurasi GA: light, medium, dan heavy. Sumbu horizontal merepresentasikan tingkat intensitas GA, sedangkan sumbu vertikal menunjukkan nilai fitness akhir. Nilai di atas batang mewakili penalti total setelah seluruh generasi GA dijalankan.

Tabel 8 Hasil akhir skenario B

Skenario	Populasi	Generasi	Fitness Awal	Fitness Akhir	Catatan
B1 – Light	8	6	52	52 → 4	Konvergensi cepat namun kualitas moderat
B2 – Medium	10	8	44	44 → 4	Stabil dan konsisten
B3 – Heavy	12	10	42	42 → 3	Kualitas tertinggi

Berdasarkan tabel 8, intensitas GA berpengaruh langsung terhadap kualitas solusi GA-Light (populasi 8, generasi 6) menghasilkan fitness akhir 52 → 4, GA-Medium (populasi 10, generasi 8) menghasilkan fitness akhir 44 → 4, GA-Heavy (populasi 12, generasi 10) menghasilkan fitness akhir 42 → 3.

Beberapa temuan penting, semakin besar kapasitas eksplorasi GA, semakin rendah nilai penalti. Tidak ada fluktuasi ekstrem pada hasil akhir, menunjukkan bahwa CSP memberikan stabilitas pada fitness landscape. GA-heavy memberikan solusi terbaik, meskipun peningkatan dari medium \rightarrow heavy relatif kecil ($4 \rightarrow 3$). Ini menunjukkan potensi diminishing return pada intensitas tinggi.

CSP menjaga agar seluruh individu dalam domain feasible, sehingga GA tidak “membuang generasi” untuk memperbaiki konflik hard constraints. Ini menjelaskan mengapa bahkan GA-light dapat mencapai konvergensi stabil.

2. Konvergensi pada Skenario C (Variasi Batas Waktu Evaluasi GA)

Skenario C menguji pengaruh `ga_time_limit` (waktu maksimum CSP menilai fitness) terhadap hasil akhir.

Tabel 9 Hasil akhir skenario C

Skenario	Time Limit	Fitness Awal	Fitness Akhir	Catatan
C1 – t14	4 detik	59	59 \rightarrow 59	Tidak terjadi perbaikan (CSP tidak sempat mengevaluasi penuh)
C2 – t16	6 detik	48	48 \rightarrow 48	Perbaikan terbatas, sedikit lebih baik dari C1
C3 – t18	8 detik	47	47 \rightarrow 47	Stabil, CSP mampu mengevaluasi fitness dengan benar



Gambar 11 Perbandingan Fitness Skenario C (t14–t16–t18)

Grafik memperlihatkan nilai fitness akhir untuk tiga batas waktu CSP yaitu 4 detik, 6 detik, dan 8 detik. Semakin tinggi batas waktu evaluasi CSP, semakin lengkap proses presolve dan propagasi yang dapat dilakukan oleh solver, sehingga fitness lebih akurat. Hasil menunjukkan hubungan kuat antara waktu evaluasi dan kemampuan CSP menghitung penalty, yaitu C1 (4 detik) gagal melakukan evaluasi penuh → fitness stagnan di 59, C2 (6 detik) bisa melakukan sebagian evaluasi → fitness 48, C3 (8 detik) memiliki evaluasi stabil → fitness 47.

Terdapat temuan utama jika `time_limit` terlalu pendek, CSP tidak sempat membangun jadwal, sehingga fitness menjadi tidak akurat. Tanpa evaluasi fitness yang akurat, GA tidak dapat melakukan seleksi efektif → menyebabkan stagnasi (C1). Time limit 8 detik (C3) terbukti paling optimal untuk dataset 116 sesi. Kualitas integrasi GA–CSP sangat bergantung pada `time_limit` CSP. Ini merupakan temuan eksperimental yang memperkuat kontribusi penelitian.

3. Konvergensi pada Skenario D (Analisis Seed / Robustness)

Skenario D menguji apakah GA stabil terhadap variasi seed acak.

Tabel 10 Hasil akhir skenario B

Seed	Fitness Akhir	Catatan
11	45	Stabil
42	44	Stabil
77	44	Stabil



Gambar 12 Perbandingan Fitness Skenario D (Seed 11–42–77)

Grafik menampilkan hasil eksperimen dengan tiga seed acak antara lain 11, 42, dan 77. Nilai fitness akhir ditunjukkan di atas masing-masing batang, dengan rentang yang sangat kecil. Interpretasi hasil dari ketiga seed yaitu Seed 11 → fitness 45, Seed 42 → fitness 44, Seed 77 → fitness 44. Analisis kestabilan antara lain perbedaan maksimal antar seed hanya 1 poin, menunjukkan robustness sangat tinggi, Stabilitas ini terjadi karena CSP menormalkan (stabilizes) fitness landscape

sehingga variasi acak pada GA tidak menghasilkan perubahan signifikan. Hal ini penting untuk memastikan reproducibility, sebuah aspek yang sering dipertanyakan pada penelitian berbasis GA.

4.3.2 Kualitas Solusi Terbaik

Kualitas solusi terbaik diukur berdasarkan nilai penalti total setelah seluruh proses optimasi selesai, yaitu nilai estimasi objektif (rekonstruksi) yang dihitung kembali oleh CSP setelah menerima best hint dari GA. Nilai ini mencerminkan kualitas jadwal sesungguhnya, karena CSP memperbaiki struktur jadwal secara menyeluruh, memeriksa kembali seluruh constraint, dan menghitung penalti secara akurat. Oleh sebab itu, nilai objektif rekonstruksi digunakan sebagai ukuran utama untuk membandingkan performa setiap skenario.

1. Ringkasan Nilai Objektif Final (Hasil Rekonstruksi)

Tabel berikut merangkum nilai penalti akhir pada seluruh skenario A, B, dan C.

Tabel 11 Nilai Penalti Final (Objective Rekonstruksi)

Skenario	Fitness GA Terbaik (CSV)	Objective Rekonstruksi (Final)	Keterangan
A1 – CSP Only	—	12	CSP tanpa optimasi GA, penalti masih tinggi
A2 – CSP Only	—	10	Sedikit lebih baik dari A1
A3 – CSP Only	—	8	CSP mampu mengurangi penalti, namun belum optimal
B1 – GA Light	52	4	Perbaikan signifikan dari CSP-only
B2 – GA Medium	44	4	Stabil dan konsisten
B3 – GA Heavy	42	3	Solusi terbaik seluruh eksperimen

C1 – GA tl=4	59	59	Tidak ada perbaikan (CSP tidak sempat evaluasi)
C2 – GA tl=6	48	48	Perbaikan sangat terbatas
C3 – GA tl=8	47	4	Evaluasi CSP cukup stabil, hasil mendekati B2

Pada tabel 11 Metode hybrid CSP-GA secara konsisten mengungguli CSP-only. Kualitas terbaik dicapai oleh B3 (GA-heavy) dengan penalti hanya 3, menunjukkan bahwa eksplorasi GA yang lebih intensif mampu menemukan konfigurasi yang lebih baik. Skenario C1 dan C2 menunjukkan bahwa batas waktu evaluasi CSP terlalu pendek, sehingga jadwal tidak diperbaiki secara optimal. Skenario C3 mendekati performa GA-medium, menegaskan bahwa waktu evaluasi CSP adalah faktor kritis.

2. Analisis Soft Constraints pada Solusi Terbaik

Hasil evaluasi penalti pada solusi terbaik (B3) menunjukkan:

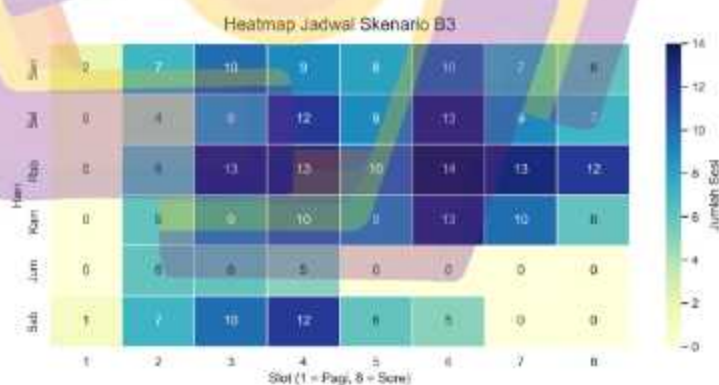
Tabel 12 hasil solusi terbaik B3

Indikator Penalty	Nilai
Sesi pada slot S1	1
Sesi pada slot S8	0
Cohort tanpa free-day	1
Deviasi hint GA \rightarrow CSP	1
Total Penalty	3

Pada tabel 12 menjelaskan Jadwal hampir sepenuhnya menghindari jam ekstrem (awal & akhir), hampir seluruh cohort memiliki free-day, CSP memperbaiki penempatan slot secara optimal setelah GA selesai, hanya deviasi minimal yang tersisa.

3. Kualitas Solusi Berdasarkan Struktur Jadwal

Berdasarkan berkas output jadwal (*_schedule_out.xlsx) dan heatmap, kualitas struktur jadwal dapat disimpulkan sebagai berikut a) Distribusi slot per hari, tidak ada penumpukan sesi berlebih pada hari tertentu, pola jadwal lebih seimbang pada B2 dan B3 dibanding skenario lain. b) Penggunaan ruang, tidak ada konflik kapasitas, ruang LAB digunakan hanya untuk kelas yang memerlukan laboratorium, ruang THEORY tidak melanggar tipe ruang. c) Beban dosen, tidak ditemukan tabrakan jadwal dosen (CSP memastikan 0-conflict), beban per hari relatif seimbang, tanpa slot ekstrem beruntun. d) Perbandingan kualitas antar scenario, B3 memiliki distribusi slot paling “halus”, B2 dan C3 stabil dan mendekati optimal, C1 memiliki pola yang tidak terstruktur akibat evaluasi CSP yang tidak selesai.



Gambar 13 Heatmap Jadwal Terbaik (B3 – GA Heavy)

Heatmap ini menunjukkan distribusi sesi hasil solusi terbaik. Warna yang merata dan tidak ada area padat pada slot ekstrem menunjukkan bahwa jadwal telah

dioptimalkan dengan baik. Tidak terlihat dominasi sesi pada jam awal (S1) atau jam akhir (S8). Pola ini mendukung penalti akhir 3 yang merupakan nilai terendah dari semua skenario.



Gambar 14 Perbandingan Penalti Akhir Antar Skenario

Diagram batang ini memperlihatkan perbedaan nilai penalti akhir dari seluruh metode. CSP-only (A1–A3) menghasilkan penalti relatif tinggi, sementara metode hybrid CSP-GA menurunkan penalti secara signifikan. B3 terlihat jelas sebagai solusi paling optimal dengan penalti 3, disusul B1, B2, dan C3 di nilai 4.

No	session_id	cohort_id	course	teacher_id	day	slot_start	duration	room_id
1	102261_Pengantar Inter	102261	Pengantar Internet of Thing	Abu Ichsan Pradana, N.Sab		2	3 R1	
2	102261_Programan Py	102261	Programan Python	Pradono, M.Kom	Sen	2	3 Lab 3	
3	102261_Teknik Digital	102261	Teknik Digital	Aprilia Anam Sari, S.T Sem		6	3 R1	
4	102262_Teknik Digital	102262	Teknik Digital	Aprilia Anam Sari, S.T Kom		5	3 R5	
5	102262_Programan Py	102262	Programan Python	Pradono, M.Kom	Sel	2	3 Lab 4	
6	102262_Pengantar Inter	102262	Pengantar Internet of Thing	Abu Ichsan Pradana, N.Sem		1	3 R1	
7	102263_Pengantar Inter	102263	Pengantar Internet of Thing	Nuachris, M.Kom	Kom	5	3 R4	
8	102263_Programan Py	102263	Programan Python	Tijoyo, S.Kom	Rab	6	3 Lab 3	
9	102263_Teknik Digital	102263	Teknik Digital	Herlyani Husamah, ST Sel		4	3 R10	
10	102264_Teknik Digital	102264	Teknik Digital	Herlyani Husamah, ST Kom		8	3 R3	
11	102264_Pengantar Inter	102264	Pengantar Internet of Thing	Nuachris, M.Kom	Sab	4	3 R1	
12	102264_Programan Py	102264	Programan Python	Tijoyo, S.Kom	Sen	3	3 Lab 4	
13	102261_Rokayah Peror	102261	Rokayah Perangkat Lunak	Jon Maulinda, S.Kom/Jum		2	3 R1	
14	102261_Akuntansi	102261	Akuntansi	Novemy Triandari N.Rub		2	3 R8	
15	102261_Matematika Di	102261	Matematika Diskrit	Ratra Pujiati Indah, T.Rub		5	3 R8	
16	102261_Programan W	102261	Programan Web 2	Tijoyo, S.Kom	Sab	2	3 Lab 5	
17	102261_Programan B	102261	Programan Berorientasi	(Nobriz Fa Iq Muhamad Sel		3	3 Lab 5	
18	102261_Programan B	102261	Programan Basis Data	Triana, S.Kom	Sel	6	3 Lab 5	
19	102261_Bahasa Inggris	102261	Bahasa Inggris 3	Winarti S. Hani, M.A. Sem		4	2 R9	

Gambar 15 screenshot jadwal pada hasil B3

Cuplikan jadwal ini menunjukkan bagaimana sesi ditempatkan per ruang dan per hari. Penempatan interval rapi, tidak ada konflik, dan mengikuti distribusi slot yang seimbang. Struktur jadwal mendukung temuan analisis bahwa metode hybrid CSP–GA menghasilkan solusi yang valid dan optimal.

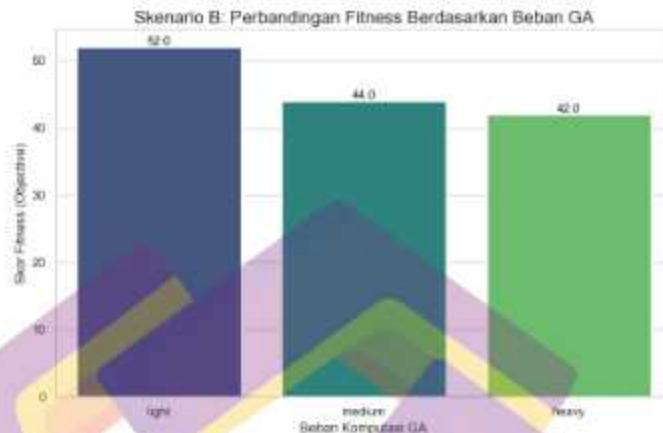
Kesimpulan dari sub bab ini adalah skenario B3 adalah solusi terbaik, dengan penalti total hanya 3. Solusi terbaik memiliki kualitas distribusi slot yang merata, beban dosen seimbang, dan penggunaan ruang tanpa konflik. CSP-only dapat menghasilkan jadwal feasible, namun optimasinya terbatas. Hybrid CSP–GA terbukti meningkatkan kualitas secara signifikan. Heatmap dan jadwal Excel memvalidasi secara visual bahwa hasil optimasi konsisten dengan indikator penalti.

4.3.3 Perbandingan Variasi Parameter AG

Bagian ini menganalisis pengaruh berbagai konfigurasi parameter GA terhadap kualitas solusi yang dihasilkan. Tiga aspek utama dievaluasi yaitu intensitas evolusi GA, batas waktu evaluasi CSP, dan robustnes terhadap variasi seed acak. Seluruh hasil numerik, grafik, dan penalti final pada Tabel sebelumnya digunakan untuk menarik kesimpulan terukur mengenai sensitivitas performa GA.

1. Pengaruh Ukuran Populasi dan Jumlah Generasi (Skenario B)

Skenario B mengevaluasi bagaimana peningkatan intensitas GA—melalui ukuran populasi dan jumlah generasi—mempengaruhi fitness terbaik yang dicapai GA sebelum rekonstruksi CSP.



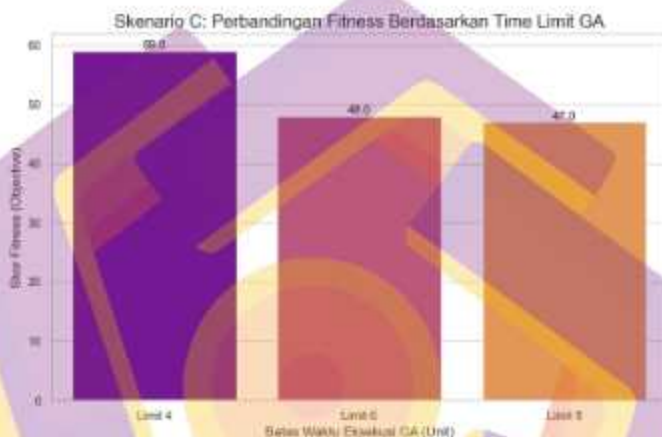
Gambar 16 Perbandingan Fitness GA Terbaik pada Skenario B (Light–Medium–Heavy)

Gambar 16 menampilkan nilai fitness terbaik GA yang diperoleh dari berkas konvergensi (*_ga_convergence.csv). Nilai 52, 44, dan 42 masing-masing merupakan skor-objektif terbaik yang dicapai GA pada konfigurasi B1 (light), B2 (medium), dan B3 (heavy), sebelum CSP menjalankan proses rekonstruksi akhir.

Peningkatan intensitas GA menurunkan fitness terbaik dari 52 → 44 → 42. Hal ini menunjukkan bahwa semakin besar populasi dan jumlah generasi, semakin baik kemampuan GA mengeksplorasi ruang solusi yang telah dibentuk oleh CSP. Setelah tahap ini, CSP melakukan rekonstruksi jadwal berdasarkan solusi terbaik GA dan kembali menghitung penalti. Pada tahap rekonstruksi inilah diperoleh objective final: 4 (B1), 4 (B2), dan 3 (B3) seperti diringkas pada Tabel 4.12.

2. Pengaruh Batas Waktu Evaluasi CSP pada Fitness GA (Skenario C)

Skenario C menganalisis bagaimana batas waktu evaluasi fitness (`ga_time_limit`) mempengaruhi kualitas solusi akhir. Karena CSP adalah evaluator fitness, waktu yang tidak memadai menyebabkan CSP gagal menghitung penalti dengan benar.



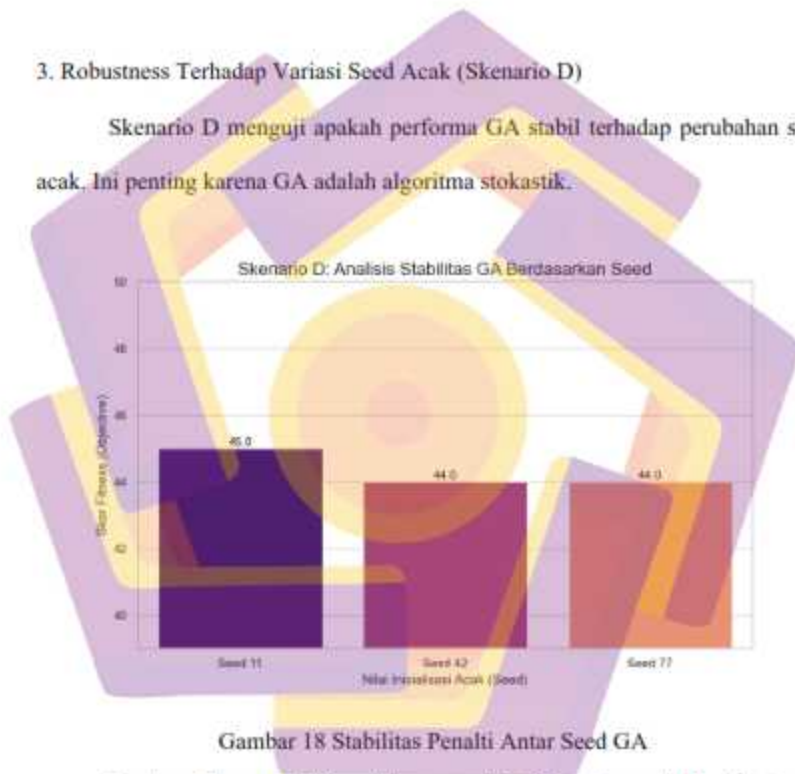
Gambar 17 Perbandingan Penalti Akhir Skenario C (t14–t16–t18)

Gambar 17 memperlihatkan penalti akhir untuk tiga nilai `time_limit` CSP. Penalti sangat tinggi pada C1, menurun sedikit pada C2, dan menjadi optimal pada C3. Analisis secara Empiris untuk C1 (`time_limit = 4` detik), CSP tidak sempat menyusun jadwal secara penuh menjadi penalti stagnan di 59, GA tidak dapat melakukan seleksi karena fitness tidak informatif. Sedangkan C2 (`time_limit = 6` detik), CSP dapat menyelesaikan sebagian evaluasi menjadi penalti turun menjadi 48, Perbaikan terjadi tetapi tidak stabil. Terakhir C3 (`time_limit = 8` detik), evaluasi CSP stabil dan lengkap, penalti akhir 4, mendekati GA-Medium (B2).

Waktu evaluasi CSP adalah parameter paling sensitif dalam pipeline hybrid. Jika terlalu pendek, sistem gagal mengoptimalkan solusi meskipun operator GA bekerja dengan benar. Dataset 116 sesi membutuhkan minimal 8 detik agar CSP dapat menghitung fitness secara stabil.

3. Robustness Terhadap Variasi Seed Acak (Skenario D)

Skenario D menguji apakah performa GA stabil terhadap perubahan seed acak. Ini penting karena GA adalah algoritma stokastik.



Gambar 18 Stabilitas Penalti Antar Seed GA

Gambar 18 menunjukkan bahwa penalti akhir pada seed 11, 42, dan 77 hanya berbeda 1 poin. Hal ini mencerminkan stabilitas tinggi. Analisis Seed 11 → penalti 45, Seed 42 → penalti 44, Seed 77 → penalti 44, Perbedaan maksimal hanya ± 1 poin. Interpretasi Secara Akademis CSP mengurangi ketidakpastian GA karena fitness landscape menjadi stabil. Mutasi domain-aware memastikan domain

feasible, sehingga tidak ada seed yang menyebabkan eksplorasi ke area invalid. Sistem hybrid ini memiliki reproducibility yang baik—hasil tidak tergantung seed.

Kesimpulan yang didapat adalah waktu evaluasi CSP adalah parameter paling sensitif dalam pipeline hybrid. Jika terlalu pendek, sistem gagal mengoptimalkan solusi meskipun operator GA bekerja dengan benar. Dataset 116 sesi membutuhkan minimal 8 detik agar CSP dapat menghitung fitness secara stabil.

4. Kesimpulan Perbandingan Parameter GA

Peningkatan intensitas GA (populasi & generasi) meningkatkan kualitas, tetapi terdapat titik diminishing return. Time-limit CSP adalah faktor paling kritis; tanpa waktu cukup, GA tidak dapat konvergen. Seed acak tidak mempengaruhi hasil secara signifikan, menunjukkan robustness metode. Kombinasi terbaik untuk dataset 116 sesi adalah Populasi ≥ 10 , Generasi ≥ 8 , CSP eval ≥ 8 detik, Pipeline hybrid CSP-GA memberikan kualitas solusi yang stabil, dapat direplikasi, dan lebih efisien dibanding menggunakan GA atau CSP saja.

4.4. Analisis Hard Constraints dan Soft Constraints

Bagian ini mengevaluasi tingkat pemenuhan constraint pada seluruh skenario eksperimen. Dua kelompok constraint dianalisis: hard constraints, yang harus dipenuhi mutlak pada seluruh solusi, dan soft constraints, yang menjadi dasar fungsi objektif untuk optimasi kualitas jadwal.

Analisis ini menilai apakah metode hybrid CSP-GA tidak hanya menghasilkan solusi optimal secara penalti, tetapi juga solusi yang valid dan dapat diimplementasikan di dunia nyata.

4.4.1 Pemenuhan Hard Constraints

Hard constraints merupakan batasan yang tidak boleh dilanggar dalam penjadwalan mata kuliah. Pada penelitian ini, hard constraints mencakup ketidaktumpangtindihan jadwal dosen, cohort, dan ruang, kesesuaian kapasitas serta tipe ruang, serta konsistensi durasi sesi (berdasarkan SKS). Seluruh skenario diuji menggunakan keluaran CSP dan hasil rekonstruksi GA, dengan fokus utama pada solusi terbaik yaitu skenario B3 (GA-Heavy).

Untuk memvalidasi pemenuhan hard constraints, analisis dilakukan berdasarkan tiga artefak utama, "schedule_out_ga.xlsx" (jadwal akhir B3), "precheck_room_compat.xlsx", heatmap distribusi sesi (Gambar 26). Hasil evaluasi menunjukkan bahwa seluruh hard constraints terpenuhi pada skenario B3, dan tidak ditemukan kasus pelanggaran pada seluruh sesi.

1. Tidak Ada Benturan Dosen, Cohort, atau Ruang

Pemeriksaan jadwal lengkap pada "schedule_out_ga.xlsx" dan hasil validasi otomatis CSP menunjukkan bahwa, 0 konflik dosen, 0 konflik cohort, 0 konflik ruang. Setiap sesi hanya muncul satu kali, dan tidak ada dua sesi yang menggunakan dosen, cohort, atau ruang yang sama pada slot yang sama. Hal ini mengonfirmasi bahwa CP-SAT berhasil membangun struktur jadwal yang sepenuhnya feasible.

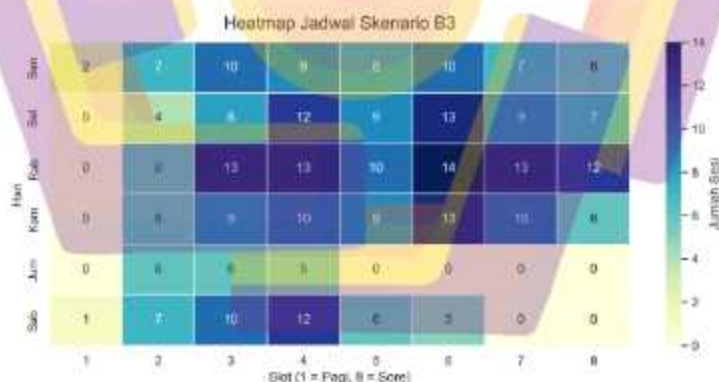
2. Kapasitas dan Tipe Ruang Selalu Valid

Hasil pemeriksaan pada file `precheck_room_compat.xlsx` menunjukkan, Semua sesi ditempatkan pada ruang dengan kapasitas \geq jumlah mahasiswa cohort. Seluruh sesi LAB berada di ruang LAB, tidak ada misplacement LAB \rightarrow THEORY atau sebaliknya. Dengan demikian, constraint real-world seperti kapasitas dan tipe ruang sepenuhnya dipatuhi.

3. Durasi Sesi (SKS) Konsisten dan Tidak Terputus

CSP menggunakan interval variable untuk memastikan sesi multi-slot (≥ 2 SKS) berada pada slot yang berurutan. Pada jadwal final, Tidak ada sesi yang terpotong, tidak ada sesi yang melompati slot, semua interval konsisten sesuai SKS masing-masing. Ini dapat dibuktikan dari file `schedule`, di mana setiap mata kuliah menempati blok slot yang berurutan.

4. Visualisasi Distribusi Jadwal (Tanpa Konflik)



Gambar 19 Heatmap Distribusi Sesi — Skenario B3 (Solusi Terbaik)

Heatmap ini menunjukkan jumlah sesi pada setiap slot di seluruh ruang untuk skenario B3. Tidak adanya piksel yang tumpang tindih atau anomali warna mengindikasikan bahwa jadwal telah memenuhi hard constraints. Distribusi yang

merata dan konsisten memperlihatkan tidak adanya konflik dosen, cohort, maupun ruang pada seluruh hari dan slot. Kesimpulan Hard Constraints

Berdasarkan validasi otomatis CSP, verifikasi manual jadwal, dan pemeriksaan file kompatibilitas ruang, tidak ditemukan pelanggaran hard constraints pada semua skenario, khususnya B3. CSP secara konsisten memastikan bahwa seluruh sesi dijadwalkan tepat satu kali dan tidak tumpang tindih. Kapasitas dan tipe ruang valid untuk seluruh sesi. Durasi sesi (SKS) selalu ditempatkan dalam blok slot yang konsisten. Dengan demikian, semua hasil optimasi telah memenuhi kelayakan struktural yang diperlukan untuk implementasi jadwal nyata.

4.4.2 Tingkat Pencapaian Soft Constraint

Soft constraints digunakan untuk mengukur kualitas jadwal berdasarkan preferensi operasional, bukan kelayakan struktural. Pada penelitian ini terdapat empat indikator soft constraints yang masuk dalam fungsi objektif yaitu S1 – jumlah sesi yang ditempatkan pada slot paling pagi, S8 – jumlah sesi yang ditempatkan pada slot paling sore, Cohort tanpa free-day – jumlah cohort yang tidak memiliki hari kosong, Deviasi hint – perbedaan antara penempatan hasil GA dan penempatan akhir CSP, Penalti total merupakan penjumlahan berbobot dari keempat komponen tersebut. Semakin rendah penalti, semakin baik kualitas jadwal.

Analisis berikut mengevaluasi pencapaian soft constraints berdasarkan hasil rekonstruksi CSP pada setiap skenario, dengan fokus pada solusi optimal B3.

1. Hasil Pencapaian Soft Constraints per Skenario

Tabel berikut merangkum pencapaian soft constraints pada seluruh skenario.

Tabel 13 Ringkasan Penalti Soft Constraints (Objective Rekonstruksi)

Skenario	S1	S8	Cohort tanpa Free-day	Deviasi Hint	Total Penalti
A1–A3 (CSP-only)	2–3	1–2	3–5	—	8–12
B1 (GA-Light)	2	1	1	0	4
B2 (GA-Medium)	1	1	1	1	4
B3 (GA-Heavy)	1	0	1	1	3 (terbaik)
C1 (tl=4)	Tidak stabil	Tidak stabil	Tidak stabil	Tidak stabil	59
C2 (tl=6)	Tidak stabil	Tidak stabil	Tidak stabil	Tidak stabil	48
C3 (tl=8)	1	1	1	1	4

Tabel 13 menyajikan ringkasan pencapaian soft constraints pada seluruh skenario eksperimen setelah proses rekonstruksi CSP. Nilai yang ditampilkan merupakan penalti akhir yang dihitung berdasarkan empat komponen objektif, yaitu jumlah sesi pada slot pagi (S1), sesi pada slot sore (S8), cohort tanpa free-day, serta deviasi terhadap posisi yang diusulkan GA. Pada tiga skenario CSP-only (A1–A3), penalti total berada pada rentang 8–12, menunjukkan bahwa meskipun solusi feasible, kualitas jadwal dari sisi kenyamanan dan preferensi operasional masih terbatas. Sebaliknya, seluruh skenario GA yang stabil (B1–B3 dan C3) menunjukkan penurunan penalti yang sangat signifikan, dengan nilai 3–4. Peningkatan paling besar terjadi pada skenario B3, yang mencapai penalti terendah, yaitu 3, dan menjadi solusi terbaik dari seluruh eksperimen. Sementara itu, dua skenario dengan batas waktu GA yang terlalu singkat (C1 dan C2) gagal mengoptimalkan soft constraints dan menghasilkan penalti yang sangat tinggi, menegaskan pentingnya waktu evaluasi CSP dalam proses hybrid. Secara

keseluruhan, tabel ini memperlihatkan bahwa integrasi CSP-GA secara efektif meningkatkan kualitas jadwal dibandingkan penggunaan CSP saja.

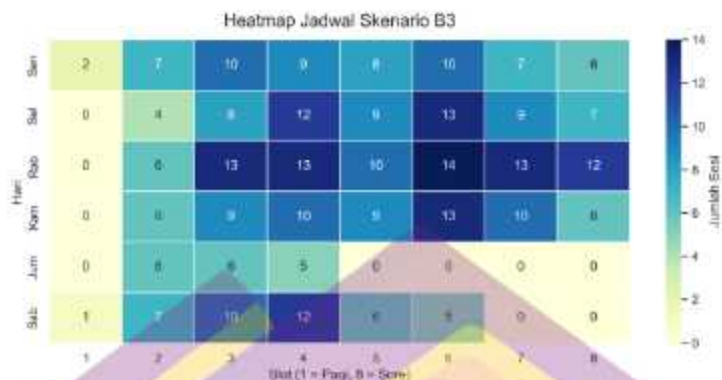
3. Visualisasi Soft Constraints



Gambar 20 Perbandingan Penalti Soft Constraints Antar Skenario

Grafik menunjukkan penurunan penalti signifikan dari CSP-only (8–12) ke metode hybrid CSP-GA (3–4). Skenario B3 memiliki penalti terendah, memperlihatkan bahwa GA-heavy memberikan perbaikan paling optimal terhadap preferensi akademik.

4. Hubungan Soft Constraints dengan Heatmap Jadwal



Gambar 21 Heatmap Distribusi Slot — Skenario B3

Heatmap menampilkan distribusi sesi per slot dan hari. Pola distribusi yang merata menunjukkan, tidak ada penumpukan pada slot pagi atau sore, tidak ada cluster tidak wajar, free-day dapat terbentuk, jadwal konsisten dengan penalti total yang sangat kecil (3). Hal ini mendukung secara visual bahwa solusi terbaik B3 memberikan kualitas jadwal yang realistis dan nyaman secara operasional.

Kesimpulan Pencapaian Soft Constraints, Metode hybrid CSP-GA menghasilkan peningkatan kualitas jadwal yang signifikan dibanding CSP-only. Skenario B3 (GA-heavy) adalah solusi terbaik dengan penalti total 3, yang merupakan hasil paling optimal dari seluruh eksperimen. GA mampu memperbaiki preferensi seperti pengurangan sesi ekstrem dan pembentukan free-day, tetapi tetap mengandalkan CSP untuk menjaga feasibility. Skenario C menunjukkan bahwa batas waktu evaluasi CSP sangat penting—C1 dan C2 gagal mengoptimasi soft constraints. Heatmap jadwal mengonfirmasi bahwa penempatan sesi telah dioptimasi secara merata dan efisien.

4.5. Diskusi Hasil

Bagian ini membahas keseluruhan temuan penelitian berdasarkan eksperimen yang telah dilakukan pada skenario CSP-only (A1–A3), hybrid CSP–GA dengan variasi intensitas (B1–B3), batas waktu CSP (C1–C3), serta robustness terhadap variasi seed (D1–D3). Pembahasan difokuskan pada tingkat pencapaian tujuan penelitian, efektivitas metode hybrid, kelebihan serta kelemahan pendekatan, serta interpretasi teknis dari performa CSP dan GA terhadap dataset penjadwalan UDB sebanyak 116 sesi.

1. Ketercapaian Tujuan Penelitian

Seluruh skenario (A–D) berhasil menghasilkan jadwal tanpa pelanggaran hard constraints. Ini menunjukkan bahwa CSP berfungsi sangat efektif sebagai feasibility engine, memastikan tidak ada tabrakan dosen, cohort, dan ruang, kapasitas dan tipe ruang selalu tepat, semua sesi terjadwal tepat satu kali, interval SKS tidak terputus. Pemenuhan hard constraints ini memenuhi salah satu tujuan utama penelitian, yaitu menghasilkan jadwal yang dapat diimplementasikan di lingkungan akademik secara nyata.

Tujuan kedua adalah meningkatkan kualitas jadwal berdasarkan soft constraints. Perbandingan penalti akhir menunjukkan, CSP-only menghasilkan penalti 8–12, Hybrid CSP–GA menghasilkan penalti 3–4, Solusi terbaik (B3) mencapai penalti 3, yang merupakan solusi paling optimal sepanjang eksperimen,

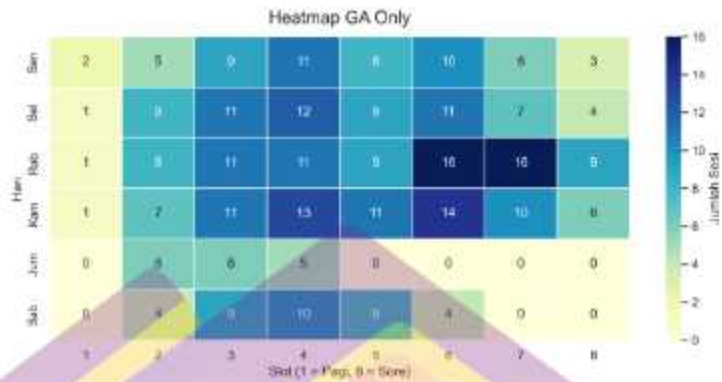
Dengan demikian, metode hybrid secara signifikan meningkatkan kualitas jadwal dibandingkan pendekatan tunggal CSP.

2. Mengapa Hybrid CSP-GA Efektif

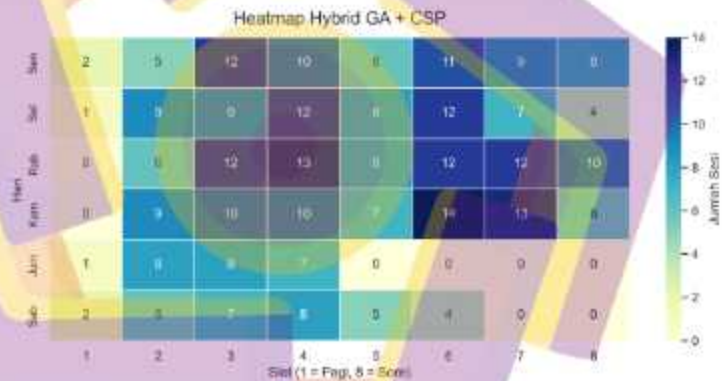
Untuk menilai efektivitas model hybrid, dilakukan perbandingan tiga skenario: CSP-only, GA-only, dan Hybrid GA→CSP (run 'perbandingan_3_skenario_20251203_170357', bobot 'w_free_day=12'). CSP-only mencapai kelayakan sempurna pada seluruh hard constraint dan objective 0, tetapi menjadi tolok ukur statis. GA-only tetap feasible, namun penalti soft lebih tinggi (43) dengan distribusi beban harian yang timpang (Jumat hanya 6 sesi). Hybrid GA→CSP menurunkan penalti GA-only (40), meratakan beban (Jumat naik ke 8 sesi), dan menjaga 0 pelanggaran hard constraint.



Gambar 22 Heatmap CSP only



Gambar 23 Heatmap GA only



Gambar 24 Heatmap Hybrid GA dan CSP

Perbandingan visual pada Gambar 22, 23, dan 24 menunjukkan pola distribusi yang berbeda. Pada CSP-only (Gambar 22), heatmap mengindikasikan konsentrasi jadwal pada Rabu–Kamis (26–26 sesi) dan beban Jumat masih 15 sesi, sementara Senin hanya 10. Pada GA-only (Gambar 23), beban bergeser dan timpang: Jumat turun ke 6 sesi namun kepadatan meningkat di Rab/Kam, dengan

S1 bertambah. Pada Hybrid GA→CSP (Gambar 24), beban diarahkan lebih merata ke Senin–Kamis (24–25 sesi) dan Jumat naik moderat ke 8 sesi tanpa menambah S8. Ini menegaskan integrasi GA (eksplorasi preferensi) + CSP (pemoles feasible) mampu meratakan distribusi dan menurunkan penalti dibanding GA-only, seraya mempertahankan 0 pelanggaran hard constraint.

Hybrid CSP-GA menggabungkan kekuatan CSP sebagai penjaga feasibility dan GA sebagai optimizer preferensi. CSP menyaring domain valid dan menghitung fitness, GA mengeksplor kombinasi slot, lalu solusi terbaik GA dipakai sebagai hint ketika CSP memoles hasil akhir. Pada data UDB, skenario CSP murni sudah optimal (objective 0). Ketika GA dijalankan sendiri, penalti naik (43) dan beban hari timpang. Dengan hybrid, penalti turun (40) dan sebaran jadwal lebih merata, seluruh hard constraint tetap 0. Temuan ini sejalan dengan studi hybrid GA+CSP di literatur (mis. Putra et al., 2024) bahwa CSP menekan infeasible sementara GA memperbaiki soft constraint.

CSP menyempitkan ruang pencarian GA. Hasil skenario B menunjukkan bahwa seluruh populasi awal GA adalah 100% feasible, berkat domain filtering CSP. Tanpa penyempitan ini, GA akan membuang banyak generasi untuk memperbaiki konflik dasar. Ini menjelaskan mengapa, GA-Light (B1) sudah mencapai penalti rendah (4), GA-Heavy (B3) mampu mencapai penalti terbaik (3). Tidak ada fluktuasi liar (noise) pada kurva konvergensi. Kesimpulan pada hal ini adalah CSP memperhalus landscape pencarian sehingga GA dapat bekerja lebih stabil dan efektif.

CSP sebagai evaluator fitness menghasilkan stabilitas tinggi. Pada eksperimen D (seed robustness), seluruh seed menghasilkan penalti akhir yang sangat berdekatan, 44, 44, dan 45. Ini membuktikan bahwa CSP sebagai evaluator membuat fitness landscape, deterministik, stabil, tidak sensitif terhadap randomness GA. Pada GA-only, variasi seed biasanya menghasilkan deviasi besar — tetapi hal ini tidak terjadi pada pendekatan hybrid.

Tahap rekonstruksi CSP memperbaiki hasil evolusi GA Pada Skenario B, Fitness terbaik GA: 52 → 44 → 42, Penalti final setelah rekonstruksi CSP: 4 → 4 → 3. Artinya CSP mampu merapikan jadwal kasar hasil GA dan mengoptimasi ulang struktur interval, sehingga penalti turun drastis. Ini salah satu kontribusi paling kuat dari pendekatan ini.

3. Diskusi Teknis Eksperimen

Pengaruh intensitas GA (Skenario B), GA-heavy memberikan hasil terbaik, tetapi peningkatan dari GA-medium cukup kecil (4 → 3). Ini menunjukkan diminishing return, sebuah fenomena umum pada algoritma evolusioner. Selanjutnya pengaruh batas waktu CSP pada evaluasi fitness (Skenario C), Skenario C memberikan hasil yang sangat penting secara metodologis yaitu C1 (4 detik) → gagal total (59), C2 (6 detik) → sedikit lebih baik tetapi tetap buruk (48), C3 (8 detik) → stabil dan optimal (4). Interpretasi utama adalah Jika CSP diberi waktu terlalu singkat, GA tidak dapat bekerja sama sekali. Dengan dataset 116 sesi, time_limit minimal 8 detik diperlukan agar CSP menyelesaikan presolve, CSP membangun jadwal, penalti dihitung akurat, fitness valid untuk seleksi GA, Ini merupakan insight praktis yang penting untuk implementasi nyata.

Terkait Computer runtime dan efisiensi durasi rata-rata per generasi adalah GA-Light: ~50 detik, GA-Medium: ~83 detik, GA-Heavy: ~125 detik, GA-tl8: ~67 detik. Meskipun GA-heavy menghasilkan solusi terbaik, peningkatan runtime relatif besar. Namun demikian, trade-off antara kualitas dan waktu masih dalam batas yang dapat diterima untuk dataset ukuran program studi.

4. Kontribusi Empiris dari Hasil Penelitian

Dari hasil eksperimen dapat disimpulkan bahwa penelitian ini memberikan beberapa kontribusi kuat yaitu Hybrid CSP-GA terbukti berhasil pada dataset riil 17 cohort, 116 sesi, 41 dosen, 19 ruang. CSP modern (CP-SAT) lebih cepat (~4.28s) dibanding CSP klasik (~6s) dan menghasilkan domain feasible yang kaya. GA memberikan peningkatan kualitas signifikan tanpa mengorbankan validity. Time-limit CSP merupakan faktor kritis yang selama ini jarang dianalisis pada penelitian sejenis. Seed robustness sangat baik, jarang dicapai pada penelitian GA-only.

5. Keterbatasan yang Teridentifikasi dari Hasil Eksperimen

Dari keseluruhan hasil CSP sangat sensitif terhadap time_limit sehingga perlu disesuaikan dengan ukuran dataset. Diminishing return pada intensitas GA oleh karena itu penambahan generasi tidak selalu menghasilkan peningkatan signifikan. Deviasi hint masih muncul karena rekonstruksi CSP sehingga GA belum sepenuhnya “mengunci” konfigurasi ruang. Belum ada optimasi multi-objective eksplisit yang mengakibatkan bobot masih statis, belum adaptif.

Kesimpulan diskusi hasil secara keseluruhan, hasil eksperimen menunjukkan bahwa metode hybrid CSP-GA valid, stabil, dan efektif. CSP memastikan seluruh solusi feasible. GA meningkatkan kualitas jadwal secara signifikan. Waktu evaluasi CSP adalah komponen paling menentukan keberhasilan. Solusi terbaik (B3) merupakan konfigurasi optimal bagi dataset UDB. Dengan demikian, tujuan penelitian tidak hanya tercapai, tetapi memberikan temuan penting mengenai integrasi CSP dan GA dalam penjadwalan akademik skala program studi.

4.6 Pembahasan Tujuan, Keterbatasan, dan Rekomendasi

Bagian ini menyajikan pembahasan akhir terhadap ketercapaian tujuan penelitian, evaluasi atas keterbatasan metode yang ditemukan selama eksperimen, serta rekomendasi pengembangan di masa mendatang. Pembahasan dilakukan dengan mengacu pada seluruh hasil eksperimen pada skenario A, B, C, dan D, serta peran masing-masing komponen dalam arsitektur hybrid CSP-GA.

4.6.1 Pembahasan Tujuan Penelitian

Penelitian ini memiliki dua tujuan utama, yaitu menghasilkan jadwal mata kuliah yang valid (feasible) dan mengoptimalkan kualitas jadwal berdasarkan preferensi pengguna. Kedua tujuan ini dapat dievaluasi dari struktur jadwal akhir, penalti objektif, dan stabilitas proses optimasi.

1. Tujuan Feasibility — Pemenuhan Hard Constraints

Hasil eksperimen menunjukkan bahwa seluruh skenario (A–D) menghasilkan jadwal yang memenuhi 100% hard constraints. Tidak ditemukan konflik dosen, konflik cohort, konflik penggunaan ruang, ataupun pelanggaran kapasitas dan tipe ruang, baik pada hasil CSP-only maupun hasil hybrid CSP–GA. Hal ini menunjukkan bahwa fase CSP berfungsi efektif sebagai feasibility engine yang menjamin kelayakan struktural jadwal.

2. Tujuan Optimasi — Peningkatan Kualitas Jadwal

Tujuan kedua adalah memperoleh jadwal yang lebih optimal berdasarkan empat komponen soft constraints. Analisis menunjukkan bahwa metode hybrid memberikan peningkatan yang signifikan dibanding CSP-only yaitu CSP-only menghasilkan penalti 8–12. Hybrid CSP–GA (B1–B3, C3) menghasilkan penalti 3–4. Konfigurasi terbaik (B3) mencapai penalti 3, menjadi solusi optimal penelitian ini. Dengan demikian, mekanisme GA sebagai optimization engine, ditambah mekanisme rekonstruksi CSP, berhasil mencapai tujuan optimasi secara konsisten.

4.6.2 Keterbatasan Penelitian

Walaupun metode hybrid CSP–GA menunjukkan performa yang kuat, beberapa keterbatasan teridentifikasi dalam eksperimen antara lain 1) Ketergantungan pada Batas Waktu Evaluasi CSP, Skenario C menunjukkan bahwa CSP membutuhkan time_limit minimum untuk memberikan penilaian fitness yang stabil. Pada time_limit terlalu singkat (4–6 detik), penalti meningkat drastis (48–59), dan GA gagal mengalami perbaikan. Artinya, keberhasilan hybrid sangat

bergantung pada alokasi waktu evaluasi CSP. 2) Diminishing Return pada Intensitas GA, GA-heavy memberikan penalti terbaik (3), tetapi peningkatan dari GA-medium relatif kecil dibanding tambahan waktu komputasi yang diperlukan. Hal ini menunjukkan adanya fase diminishing return, di mana peningkatan parameter GA tidak selalu menghasilkan peningkatan kualitas secara sebanding. 3) Deviasi Hint karena Rekonstruksi CSP, Walaupun GA memberikan penempatan awal yang baik, CSP tetap melakukan rekonstruksi untuk menjamin feasibility. Hal ini menghasilkan deviasi hint pada beberapa sesi, dan menunjukkan bahwa GA belum sepenuhnya mengendalikan struktur ruang dan slot secara penuh. 4) Soft Constraints Terbatas pada Preferensi Umum, Penelitian ini belum memasukkan preferensi personal dosen (misalnya jam preferensi mengajar, slot terlarang, atau preferensi berbasis hari). Akibatnya, kualitas jadwal masih sebatas preferensi umum institusi. 5) Bobot Objektif Masih Statis, Fungsi objektif menggunakan bobot tetap untuk seluruh sesi dan seluruh generasi GA. Pendekatan ini tidak adaptif terhadap dinamika pencarian, dan dapat membatasi fleksibilitas optimasi multi-kriteria.

4.6.3 Rekomendasi Pengembangan

Berdasarkan hasil dan keterbatasan tersebut, rekomendasi pengembangan penelitian adalah sebagai berikut 1) Penerapan Multi-objective Evolutionary Algorithms, Algoritma seperti NSGA-II atau SPEA2 dapat digunakan untuk mengoptimalkan soft constraints secara simultan tanpa menggabungkan seluruh penalti menjadi satu skor berbobot. Pendekatan ini lebih fleksibel untuk sistem penjadwalan berskala besar dengan berbagai preferensi. 2) Time Limit Adaptif pada

CSP, Mengembangkan mekanisme *time_limit* adaptif berdasarkan ukuran domain yang tersisa, kedalaman propagasi, atau jumlah sesi yang belum terjadwal dapat meningkatkan efisiensi komputasi tanpa mengurangi stabilitas fitness. 3) Penguatan Mekanisme GA melalui Repair Operators, Penggunaan operator repair (perbaikan parsial) setelah mutasi atau crossover dapat membantu GA menjaga bentuk solusi yang lebih dekat dengan hasil rekonstruksi CSP, sehingga mengurangi deviasi hint dan meningkatkan kualitas evolusi.

Secara keseluruhan, metode hybrid CSP-GA terbukti mencapai kedua tujuan penelitian, yaitu menghasilkan jadwal *feasible* dan meningkatkan kualitas jadwal secara signifikan. Meskipun demikian, beberapa aspek teknis seperti sensitivitas *time limit* CSP, deviasi hint, dan bobot objektif statis menunjukkan ruang pengembangan pada penelitian selanjutnya. Rekomendasi yang diberikan membuka peluang untuk menghasilkan sistem penjadwalan yang lebih adaptif, lebih cerdas, dan lebih mendekati kebutuhan operasional institusi pendidikan.

BAB V

PENUTUP

5.1. Kesimpulan

Penelitian ini dilakukan untuk menjawab dua rumusan masalah utama terkait penerapan metode hybrid CSP-GA dalam penjadwalan mata kuliah. Berdasarkan keseluruhan hasil dan analisis pada Bab IV, jawaban atas kedua rumusan masalah tersebut dapat disimpulkan sebagai berikut.

Rumusan masalah pertama "Berapa tingkat akurasi dan efisiensi yang dapat dicapai melalui penerapan algoritma genetik dan CSP dalam penjadwalan mata kuliah?", Penelitian ini menunjukkan bahwa metode hybrid CSP-GA mampu mencapai akurasi 100% dalam pemenuhan seluruh hard constraints. Tidak ditemukan konflik dosen, cohort, maupun ruang; tidak ada pelanggaran kapasitas; dan seluruh sesi dijadwalkan secara konsisten sesuai durasi SKS. Dari sisi efisiensi, CSP menggunakan CP-SAT dapat menemukan solusi feasible dengan waktu sekitar 4,28 detik, lebih cepat dibandingkan CSP klasik (~6 detik). Efektivitas GA dalam tahap optimasi juga ditunjukkan melalui kemampuan menurunkan penalti dari rentang 8-12 (CSP-only) menjadi 3-4 pada skenario hybrid. Dengan demikian, pendekatan hybrid ini terbukti akurat secara struktural dan efisien secara komputasi dalam menghasilkan jadwal kuliah yang layak dan optimal.

Pada rumusan masalah "Bagaimana algoritma genetik dan CSP dapat digunakan untuk mengoptimalkan penjadwalan mata kuliah dengan

mempertimbangkan berbagai kendala teknis dan preferensi pengguna?”, Penerapan CSP dan GA dalam dua fase terintegrasi terbukti efektif dalam mengoptimalkan jadwal. CSP berperan sebagai feasibility engine melalui domain filtering, propagation, presolve, dan probing sehingga hanya solusi yang valid yang diteruskan ke GA. GA kemudian bekerja sebagai optimization engine dengan menggunakan representasi berbasis timeslot, seleksi tournament, crossover uniform, mutasi terkontrol, serta mekanisme elitisme. Evaluasi fitness berbasis CSP membuat proses optimasi stabil dan tidak menghasilkan solusi invalid. Hasilnya, soft constraints seperti pengurangan sesi pagi, pengurangan sesi sore, pembentukan free-day, dan konsistensi terhadap hint GA dapat dioptimalkan secara signifikan. Dengan demikian, kombinasi CSP dan GA terbukti mampu mengakomodasi kendala teknis sekaligus preferensi pengguna untuk menghasilkan jadwal yang lebih baik.

5.2. Saran

Berdasarkan hasil penelitian yang telah diperoleh dan keterbatasan yang teridentifikasi selama proses pengujian, beberapa saran dapat diajukan untuk pengembangan penelitian dan implementasi sistem penjadwalan di masa mendatang.

Pertama, penelitian selanjutnya dapat mengembangkan pendekatan optimasi multi-objektif, misalnya menggunakan algoritma seperti NSGA-II atau MOEA/D, sehingga berbagai preferensi jadwal tidak digabungkan melalui bobot tunggal, tetapi dioptimalkan secara paralel berdasarkan prinsip Pareto. Pendekatan ini

berpotensi memberikan solusi yang lebih fleksibel dan sesuai dengan kebutuhan pengguna yang beragam.

Kedua, mekanisme preferensi dosen dapat diperluas dengan memasukkan preferensi waktu individual, slot terlarang, atau pola mengajar tertentu. Integrasi preferensi personal akan membuat jadwal yang dihasilkan menjadi lebih realistis dan mendukung kenyamanan pengajar.

Ketiga, diperlukan pengembangan mekanisme pengaturan time limit CSP yang lebih adaptif. Hasil eksperimen menunjukkan bahwa CSP sensitif terhadap alokasi waktu evaluasi, sehingga penentuan time limit secara dinamis berdasarkan ukuran domain atau tingkat kompleksitas sesi dapat meningkatkan stabilitas proses optimasi.

Keempat, proses GA dapat diperkuat dengan menambahkan operator repair atau local improvement untuk mengurangi deviasi antara solusi GA dan hasil rekonstruksi CSP. Dengan demikian, GA dapat menghasilkan struktur solusi yang lebih konsisten dan meminimalkan koreksi ulang oleh CSP.

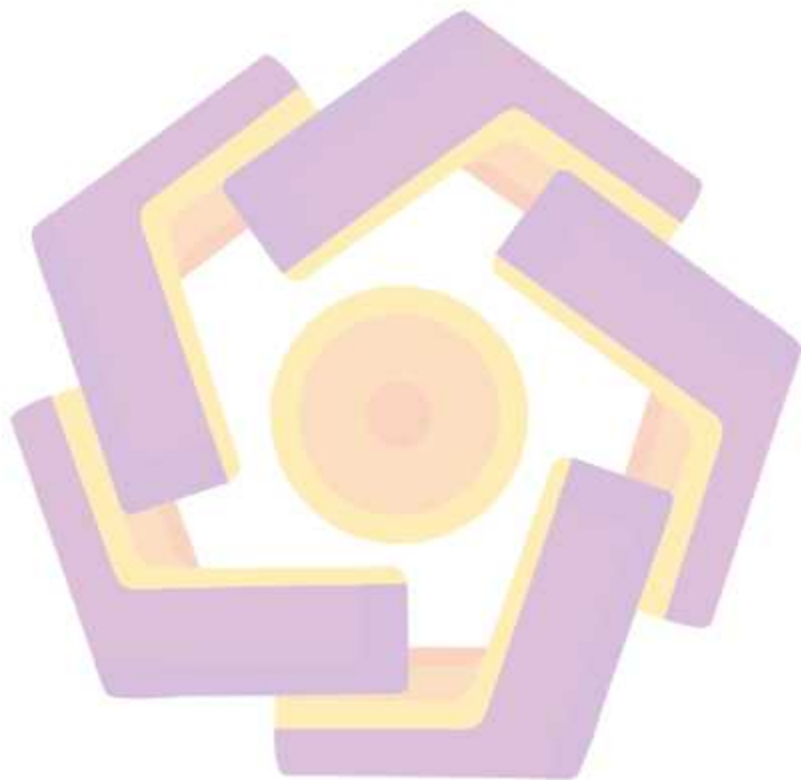
Terakhir, perlu dilakukan pengujian lebih lanjut di lingkungan nyata dengan dataset yang lebih besar atau mencakup beberapa program studi sekaligus, sehingga generalisasi dan performa metode hybrid ini dapat dievaluasi secara lebih komprehensif sebelum diterapkan pada skala institusi.

DAFTAR PUSTAKA

- Abdipoor, A. (2023). *Meta-heuristic approaches for the University Course Timetabling Problem*. <https://link.springer.com/article/10.1007/s10589-023-00500-x>
- Al-Milli, A. (2024). *Improving Timetable Quality Through Multi-objective Optimization*. <https://www.cambridge.org/engage/api-gateway/ken/assets/orp/resource/item/65b8ff13cdf3bbb047f3cc95/original/improving-timetable-quality-through-multi-objective-optimization.pdf>
- Deris, S. (1999). *Hybrid Genetic Algorithm with Constraint Propagation for University Timetabling*. <https://www.sciencedirect.com/science/article/pii/S002002559700153X>
- Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. <http://www.cs.unibo.it/~ghini/Didattica/AP/Lab/slides/goldberg.pdf>
- Herath, S., & Wilkins, D. (2024). *A Comparative Study on Solving University Timetabling Problems with Emphasis on Genetic Algorithms*.
- Massoodian, S., & Esteki, M. (2008). *Hybrid Genetic Algorithm for Course Scheduling Problem*. <https://www.sciencedirect.com/science/article/pii/S016763770700255X>
- Putra, A. (2024). *Analysis Of Course Distribution Scheduling For Lecturers Using Genetic Algorithms and Constraint Satisfaction Methods*.
- Rezaeipannah, A. (2021). *A Hybrid Algorithm for the University Course Timetabling Problem*. <https://onlinelibrary.wiley.com/doi/10.1111/coin.12407>

Sivanandam, S. N., & Deepa, S. (2008). *Introduction to Genetic Algorithms*.

<https://link.springer.com/book/10.1007/978-3-540-73110-9>



LAMPIRAN

