

Algoritma Priority-Driven Bitmask Constraint Solver (PDBCS) untuk Penjadwalan Pelajaran Sekolah Otomatis

Naufal Abdillah¹ *, Sugiono²,

^{1,2}Universitas 17 Agustus 1945 Surabaya, Jl. Semolowaru 45, Surabaya

Email : naufal@untag-sby.ac.id¹ *, sugiono@untag-sby.ac.id²

Abstrak. Penjadwalan pelajaran sekolah merupakan permasalahan *Constraint Satisfaction Problem (CSP)* yang kompleks karena harus memenuhi berbagai batasan terkait guru, mata pelajaran, dan slot waktu pembelajaran. Penelitian ini bertujuan mengembangkan algoritma *Priority-Driven Bitmask Constraint Solver (PDBCS)* untuk meningkatkan efisiensi proses penyusunan jadwal pelajaran secara otomatis. Algoritma yang diusulkan mengintegrasikan *Priority Score* untuk menentukan urutan penjadwalan berdasarkan tingkat kesulitan penempatan mata pelajaran, representasi constraint berbasis Bitmask untuk mempercepat validasi konflik, serta *Backtracking* sebagai mekanisme pencarian solusi. Pengujian dilakukan menggunakan data nyata penjadwalan sekolah dan dievaluasi berdasarkan waktu komputasi serta kemampuan memenuhi seluruh *hard constraint*. Hasil penelitian menunjukkan bahwa algoritma PDBCS mampu menghasilkan jadwal yang valid dengan waktu komputasi rata-rata 7 menit 5 detik, lebih cepat dibandingkan pendekatan *backtracking* konvensional yang memerlukan rata-rata 46 menit 45 detik. Peningkatan performa mencapai 84,9% tanpa mengurangi tingkat keberhasilan pemenuhan constraint. Hasil tersebut menunjukkan bahwa PDBCS efektif meningkatkan efisiensi penjadwalan pelajaran sekolah otomatis melalui kombinasi strategi prioritas dan representasi constraint berbasis bitmask.

Kata kunci: penjadwalan pelajaran, *constraint satisfaction problem*, *backtracking*, bitmask, *priority score*.

Abstract. School timetabling is a complex *Constraint Satisfaction Problem (CSP)* that must satisfy various constraints related to teachers, subjects, and instructional time slots. This study proposes the *Priority-Driven Bitmask Constraint Solver (PDBCS)* algorithm to improve the efficiency of automatic school timetable generation. The proposed algorithm integrates a *Priority Score* mechanism to determine scheduling order based on subject placement difficulty, Bitmask-based constraint representation to accelerate conflict validation, and *Backtracking* as the solution search mechanism. The algorithm was evaluated using real-world school timetabling data and assessed based on computational time and its ability to satisfy all hard constraints. The results show that PDBCS successfully generated valid timetables with an average computation time of 7 minutes and 5 seconds, significantly faster than conventional *backtracking*, which required an average of 46 minutes and 45 seconds. The proposed method achieved an 84.9% performance improvement while maintaining a 100% success rate in satisfying all hard constraints. These findings demonstrate that the integration of priority-based scheduling and bitmask-based constraint representation effectively improves the efficiency of automated school timetabling systems.

Keywords: school timetabling, *constraint satisfaction problem*, *backtracking*, bitmask, *priority score*.

PENDAHULUAN

Penjadwalan pelajaran merupakan salah satu aktivitas penting dalam pengelolaan akademik sekolah karena berpengaruh langsung terhadap efektivitas proses pembelajaran. Penyusunan jadwal harus mempertimbangkan berbagai komponen seperti ketersediaan guru, distribusi mata pelajaran, alokasi jam mengajar, serta aturan akademik yang berlaku. Kompleksitas permasalahan meningkat seiring bertambahnya jumlah kelas, guru, mata pelajaran, dan berbagai batasan yang harus dipenuhi secara bersamaan [1], [3]. Oleh karena itu, penjadwalan pelajaran termasuk ke dalam kelompok *Constraint Satisfaction Problem* yang bertujuan mencari solusi yang memenuhi seluruh batasan yang telah ditentukan [1], [2].

Pada lingkungan sekolah, terdapat berbagai *hard constraint* yang wajib dipenuhi, seperti guru tidak boleh mengajar pada dua kelas pada waktu yang sama, kelas tidak boleh memiliki lebih dari satu mata pelajaran pada slot yang sama, serta mata pelajaran tertentu harus ditempatkan pada waktu



khusus sesuai kebutuhan kurikulum [1], [3]. Penelitian ini berfokus pada pemenuhan hard constraint yang menentukan validitas jadwal, seperti konflik guru, konflik kelas, penempatan mata pelajaran khusus, dan pemisahan sesi mata pelajaran multi-jam pada hari yang berbeda.

Permasalahan timetabling telah menjadi topik penelitian yang aktif selama beberapa dekade dan dikenal sebagai salah satu permasalahan optimasi kombinatorial yang sulit diselesaikan secara optimal pada ukuran data yang besar [4], [8], [9], [14]. Berbagai pendekatan telah digunakan untuk menyelesaikan permasalahan timetabling, mulai dari Constraint Satisfaction Problem (CSP), Integer Programming, hingga metode metaheuristic [3], [14], [15], [16]. Pendekatan berbasis CSP dan backtracking memiliki keunggulan dalam menjamin pemenuhan seluruh hard constraint karena proses pencarian dilakukan secara sistematis hingga ditemukan solusi yang valid. Namun demikian, performa metode tersebut sangat dipengaruhi oleh ukuran ruang pencarian yang harus dieksplorasi dan biaya komputasi yang diperlukan untuk melakukan validasi constraint pada setiap langkah pencarian.

Di sisi lain, metode metaheuristik seperti Genetic Algorithm, Particle Swarm Optimization (PSO), dan Hyper-Heuristic mampu menghasilkan solusi dengan kualitas yang baik melalui mekanisme pencarian yang lebih fleksibel [5], [6], [7]. Meskipun demikian, metode tersebut umumnya memerlukan evaluasi konflik secara berulang dan tidak selalu menjamin diperolehnya solusi yang memenuhi seluruh hard constraint. Ketika ukuran permasalahan semakin besar, proses evaluasi dan eksplorasi solusi dapat menyebabkan peningkatan waktu komputasi yang signifikan. Oleh karena itu, diperlukan pendekatan yang tidak hanya mampu mempertahankan kemampuan pemenuhan constraint, tetapi juga meningkatkan efisiensi proses pencarian solusi dan validasi constraint.

Selain pendekatan berbasis metaheuristik, beberapa penelitian juga telah memanfaatkan representasi bitvector untuk memodelkan constraint pada permasalahan timetabling [11]. Representasi tersebut memungkinkan proses evaluasi constraint dilakukan secara lebih efisien melalui operasi bit-level dibandingkan struktur data konvensional. Namun implementasinya umumnya digunakan dalam kerangka optimasi yang lebih kompleks seperti satisfiability solving, integer programming, maupun constraint programming dan belum banyak dikombinasikan dengan strategi pengurutan prioritas pada proses pencarian solusi berbasis backtracking [11], [12], [13].

Tabel 1. Perbandingan Penelitian Terdahulu

Penelitian	CSP / Backtracking	PSO	Hyper-Heuristic	Bitmask / Bitvector	Priority Ordering	School Timetabling
Joshi&Agarwal (2024)	✓					✓
Thakur (2024)	✓					✓
Tassopoulos,dkk(2023)		✓				✓
Ismail, dkk. (2024)			✓			✓
Demirović&Musliu(2017)				✓		✓
PDBCS (Usulan)	✓			✓	✓	✓

Berdasarkan Tabel 1, penelitian terdahulu menunjukkan bahwa berbagai pendekatan telah digunakan untuk menyelesaikan permasalahan timetabling, termasuk CSP, metaheuristik, hyper-heuristic, dan representasi constraint berbasis bitvector. Sebagian penelitian berfokus pada pengembangan strategi pencarian solusi yang lebih efektif, sementara penelitian lainnya menitikberatkan pada peningkatan efisiensi representasi constraint. Namun demikian, sebagian besar pendekatan tersebut masih mengoptimalkan salah satu aspek secara terpisah, yaitu pengurangan ruang pencarian atau percepatan validasi constraint. Kombinasi kedua mekanisme tersebut dalam satu kerangka algoritma penjadwalan sekolah masih relatif terbatas dan belum banyak dievaluasi pada implementasi berbasis backtracking [14], [15], [16].



Meskipun berbagai pendekatan telah dikembangkan untuk menyelesaikan permasalahan timetabling, sebagian besar penelitian berfokus pada optimasi strategi pencarian solusi atau peningkatan kualitas jadwal yang dihasilkan. Penelitian yang memanfaatkan representasi constraint berbasis bitmask umumnya diterapkan pada kerangka satisfiability solving atau constraint programming, sedangkan penerapannya pada algoritma backtracking untuk penjadwalan sekolah masih relatif terbatas. Selain itu, penggunaan mekanisme pengurutan prioritas berdasarkan tingkat kesulitan penjadwalan belum banyak dikombinasikan secara langsung dengan representasi constraint berbasis bitmask dalam satu kerangka algoritma. Oleh karena itu, masih terdapat celah penelitian terkait pengembangan metode yang mampu mengurangi ruang pencarian sekaligus mempercepat proses validasi constraint pada penjadwalan pelajaran sekolah.

Kebaruan (novelty) penelitian ini terletak pada pengembangan algoritma Priority-Driven Bitmask Constraint Solver (PDBCS) yang mengintegrasikan dua mekanisme optimasi yang bekerja pada aspek berbeda dalam proses penyelesaian Constraint Satisfaction Problem. Priority Score digunakan untuk menentukan urutan penjadwalan berdasarkan tingkat kesulitan penempatan mata pelajaran sehingga ruang pencarian dapat dikurangi sejak awal proses pencarian solusi. Sementara itu, representasi constraint berbasis Bitmask digunakan untuk mempercepat proses deteksi konflik melalui operasi bitwise. Berbeda dengan penelitian terdahulu yang umumnya menerapkan kedua pendekatan tersebut secara terpisah atau dalam kerangka optimasi yang berbeda, penelitian ini mengintegrasikan keduanya secara langsung ke dalam mekanisme backtracking untuk meningkatkan efisiensi penjadwalan pelajaran sekolah. Integrasi tersebut memungkinkan pengurangan ruang pencarian dan percepatan validasi constraint dilakukan secara simultan dalam satu kerangka algoritma, yang belum banyak dibahas pada penelitian penjadwalan sekolah sebelumnya.

Pada pendekatan CSP berbasis backtracking konvensional, efisiensi algoritma sangat dipengaruhi oleh dua faktor utama, yaitu jumlah ruang pencarian yang harus dieksplorasi dan biaya komputasi untuk memvalidasi constraint pada setiap langkah pencarian. Priority Score dan Bitmask dirancang untuk mengatasi kedua faktor tersebut secara bersamaan. Priority Score mengurangi jumlah cabang pencarian yang perlu dieksplorasi dengan menempatkan komponen yang paling sulit dijadwalkan terlebih dahulu, sedangkan Bitmask mempercepat proses validasi constraint melalui operasi bitwise yang lebih efisien dibandingkan pemeriksaan berbasis iterasi. Kombinasi keduanya diharapkan mampu menghasilkan peningkatan performa yang lebih signifikan dibandingkan penggunaan salah satu teknik secara terpisah maupun pendekatan CSP konvensional.

Berdasarkan permasalahan tersebut, penelitian ini mengusulkan Algoritma Priority-Driven Bitmask Constraint Solver untuk penjadwalan pelajaran sekolah otomatis. Algoritma yang diusulkan mengombinasikan tiga komponen utama, yaitu pengurutan mata pelajaran berdasarkan tingkat prioritas, representasi constraint menggunakan bitmask, dan mekanisme *backtracking* untuk pencarian solusi. Prioritas mata pelajaran ditentukan berdasarkan tingkat kesulitan penjadwalan yang dihitung dari berbagai karakteristik mata pelajaran dan guru. Sementara itu, representasi bitmask digunakan untuk mempercepat proses validasi constraint melalui operasi bitwise sehingga konflik jadwal dapat dideteksi dengan lebih efisien dibandingkan pemeriksaan konvensional.

Pengujian dilakukan menggunakan data nyata dari SMP Negeri 1 Sedati Sidoarjo yang terdiri atas 32 kelas, 60 guru, 14 mata pelajaran, dan 33 slot pembelajaran per minggu. Evaluasi dilakukan melalui empat skenario pengujian, yaitu tanpa prioritas dan tanpa bitmask, dengan prioritas tanpa bitmask, tanpa prioritas dengan bitmask, serta menggunakan kombinasi prioritas dan bitmask pada algoritma yang diusulkan. Perbandingan tersebut bertujuan untuk mengukur kontribusi masing-masing komponen terhadap peningkatan performa sistem.

Tujuan penelitian ini adalah mengembangkan algoritma penjadwalan pelajaran sekolah otomatis yang mampu menghasilkan jadwal valid dengan waktu komputasi yang lebih efisien dibandingkan pendekatan konvensional. Hipotesis penelitian yang diajukan adalah bahwa integrasi mekanisme pengurutan prioritas dan representasi constraint berbasis bitmask mampu mengurangi



waktu komputasi secara signifikan dibandingkan pendekatan backtracking konvensional tanpa mengurangi tingkat keberhasilan pemenuhan constraint.

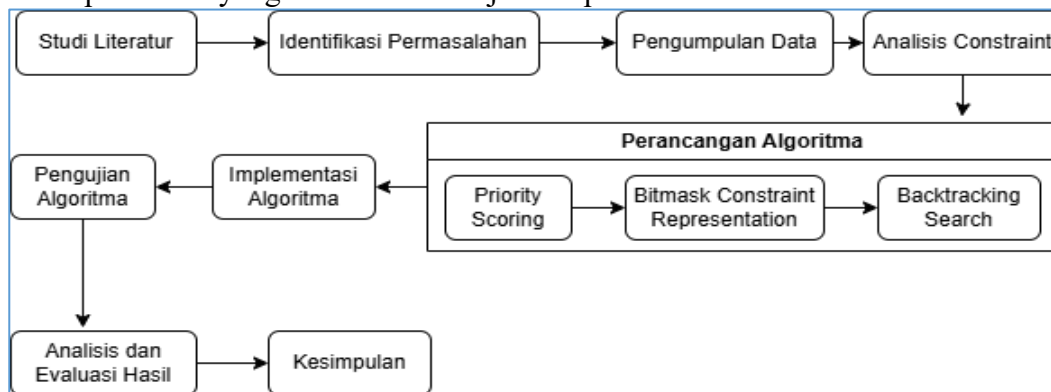
METODOLOGI PENELITIAN

Penelitian ini bertujuan mengembangkan algoritma Priority-Driven Bitmask Constraint Solver (PDBCS) untuk menghasilkan jadwal pelajaran sekolah secara otomatis dengan waktu komputasi yang lebih efisien. Algoritma yang diusulkan mengintegrasikan mekanisme pengurutan prioritas mata pelajaran, representasi constraint berbasis bitmask, dan pencarian solusi menggunakan teknik backtracking. Metode penelitian terdiri atas tahapan persiapan data, pemodelan constraint, perhitungan prioritas, proses penjadwalan, serta evaluasi kinerja algoritma.

A. Tahapan Penelitian

Penelitian ini dilakukan melalui beberapa tahapan yang meliputi studi literatur, pengumpulan data, analisis kebutuhan sistem, perancangan algoritma, implementasi, pengujian, dan evaluasi hasil. Tahapan tersebut dirancang untuk memastikan bahwa algoritma yang diusulkan mampu menyelesaikan permasalahan penjadwalan pelajaran sekolah secara efektif dan efisien.

Alur penelitian yang dilakukan ditunjukkan pada Gambar 1.



Gambar 1. Tahapan Penelitian

Penelitian diawali dengan studi literatur untuk mengidentifikasi perkembangan penelitian dan celah riset terkait school timetabling, constraint satisfaction problem (CSP), backtracking, dan teknik optimasi penjadwalan. Selanjutnya dilakukan identifikasi permasalahan, pengumpulan data dari SMP Negeri 1 Sedati, serta analisis hard constraint yang digunakan dalam penyusunan jadwal.

Berdasarkan hasil analisis tersebut, dirancang algoritma Priority-Driven Bitmask Constraint Solver (PDBCS) yang mengintegrasikan Priority Score untuk mengurangi ruang pencarian, Bitmask untuk mempercepat validasi constraint, dan Backtracking sebagai mekanisme pencarian solusi. Algoritma kemudian diimplementasikan pada sistem penjadwalan berbasis web menggunakan JavaScript dengan dukungan Web Worker.

Pengujian dilakukan menggunakan empat skenario, yaitu backtracking konvensional, Priority Score, Bitmask, dan kombinasi keduanya dalam PDBCS. Setiap skenario diuji sebanyak 100 kali menggunakan dataset yang sama. Hasil pengujian dianalisis berdasarkan tingkat keberhasilan penyusunan jadwal, waktu komputasi, serta kontribusi Priority Score dan Bitmask terhadap performa algoritma untuk mengevaluasi efektivitas PDBCS..

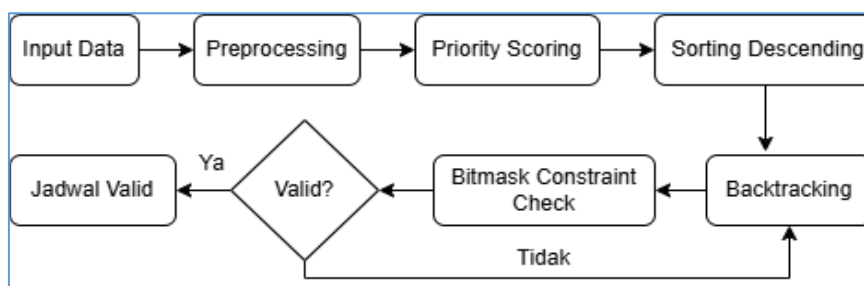
B. Arsitektur Algoritma Priority-Driven Bitmask Constraint Solver (PDBCS)

Algoritma Priority-Driven Bitmask Constraint Solver (PDBCS) merupakan metode penjadwalan yang mengombinasikan tiga komponen utama, yaitu priority scoring, representasi

constraint berbasis bitmask, dan pencarian solusi menggunakan teknik backtracking. Priority scoring digunakan untuk menentukan urutan penjadwalan mata pelajaran berdasarkan tingkat kesulitan penempatannya. Representasi bitmask digunakan untuk mempercepat proses validasi constraint melalui operasi bitwise, sedangkan backtracking digunakan untuk mencari kombinasi jadwal yang memenuhi seluruh constraint.

Secara umum, algoritma PDBCS bekerja melalui lima tahapan utama, yaitu preprocessing data, perhitungan skor prioritas, pengurutan mata pelajaran, validasi constraint menggunakan bitmask, dan pencarian solusi menggunakan backtracking.

Arsitektur algoritma yang diusulkan ditunjukkan pada Gambar 2.



Gambar 2. Arsitektur Algoritma

Berdasarkan arsitektur tersebut, implementasi setiap komponen algoritma dijelaskan lebih rinci pada Bagian E.

C. Dataset dan Lingkungan Implementasi

Penelitian ini menggunakan data penjadwalan pelajaran dari SMP Negeri 1 Sedati, Kabupaten Sidoarjo, sebagai studi kasus untuk menguji kinerja algoritma Priority-Driven Bitmask Constraint Solver (PDBCS). Data yang digunakan merupakan data aktual yang digunakan dalam proses penyusunan jadwal sekolah sehingga dapat merepresentasikan kondisi nyata yang dihadapi oleh pengelola akademik.

Dataset terdiri atas data kelas, guru, mata pelajaran, alokasi jam pembelajaran, slot waktu, serta berbagai constraint yang digunakan dalam proses penjadwalan. Data tersebut kemudian diproses menjadi struktur yang dapat digunakan oleh algoritma untuk melakukan pencarian solusi secara otomatis.

Karakteristik dataset yang digunakan dalam penelitian ditunjukkan pada Tabel 2.

Tabel 2. Karakteristik Dataset Penelitian

Parameter	Nilai
Jumlah Kelas	32
Jumlah Guru	60
Jumlah Mata Pelajaran	14
Jumlah Slot Pembelajaran per Minggu	33
Tingkat Kelas	VII, VIII, dan IX

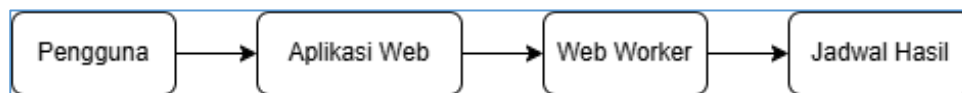
Sebanyak 32 kelas yang terdiri atas tiga tingkat pendidikan harus dijadwalkan ke dalam 33 slot pembelajaran setiap minggu. Setiap mata pelajaran memiliki alokasi jam yang berbeda-beda dan diajarkan oleh guru tertentu sesuai bidang keahlian masing-masing. Selain itu, beberapa mata pelajaran memiliki kebutuhan penjadwalan khusus yang harus dipenuhi selama proses pembentukan jadwal.

Untuk menjamin validitas jadwal yang dihasilkan, penelitian ini menerapkan empat hard constraint sebagaimana dijelaskan pada bagian sebelumnya yang harus dipenuhi sepenuhnya agar jadwal dinyatakan valid.

Lingkungan Implementasi

Implementasi algoritma dilakukan pada sistem penjadwalan berbasis web menggunakan bahasa pemrograman JavaScript. Seluruh proses penjadwalan dijalankan pada sisi klien menggunakan teknologi Web Worker sehingga proses komputasi dapat dilakukan secara terpisah dari antarmuka pengguna. Pendekatan ini memungkinkan proses pencarian solusi berlangsung tanpa mengganggu responsivitas aplikasi selama proses penjadwalan berlangsung.

Arsitektur implementasi yang digunakan dalam penelitian ditunjukkan pada Gambar 3.



Gambar 3. Arsitektur Implementasi

Proses komputasi dilakukan menggunakan model single-thread sehingga seluruh peningkatan performa yang diperoleh berasal dari optimasi algoritma yang diusulkan, bukan dari pemanfaatan pemrosesan paralel atau komputasi terdistribusi. Dengan demikian, hasil pengujian dapat menggambarkan kontribusi nyata dari mekanisme priority scoring dan representasi constraint berbasis bitmask terhadap efisiensi proses penjadwalan.

Untuk mengevaluasi performa algoritma, pengujian dilakukan menggunakan empat skenario berbeda yang membandingkan pengaruh penggunaan priority scoring dan bitmask terhadap waktu komputasi. Setiap skenario dijalankan sebanyak 100 kali menggunakan dataset yang sama sehingga hasil pengujian dapat dianalisis secara konsisten dan objektif.

Spesifikasi Perangkat Pengujian

Pengujian algoritma dilakukan menggunakan perangkat keras dan perangkat lunak sebagaimana ditunjukkan pada Tabel 3. Seluruh skenario pengujian dijalankan pada perangkat yang sama untuk menjaga konsistensi hasil pengukuran waktu komputasi.

Tabel 3. Spesifikasi Perangkat Pengujian

Komponen	Spesifikasi
Prosesor	Intel Core i3-1115G4 Generasi ke-11 (2 Core, 4 Thread, 3.00 GHz)
RAM	4 GB
Sistem Operasi	Windows 11 Pro
Browser	Google Chrome 148
Bahasa Pemrograman	JavaScript
Platform	Aplikasi Web
Metode Eksekusi	Web Worker
Arsitektur Pemrosesan	Single Thread

Penggunaan Web Worker memungkinkan proses komputasi dijalankan pada thread terpisah dari antarmuka pengguna sehingga aplikasi tetap responsif selama proses penjadwalan berlangsung. Meskipun demikian, penelitian ini hanya menggunakan satu Web Worker sehingga seluruh proses pencarian solusi tetap berjalan secara single-thread. Dengan demikian, peningkatan performa yang diperoleh berasal dari optimasi algoritma yang diusulkan, yaitu mekanisme priority scoring dan representasi constraint berbasis bitmask, bukan dari pemanfaatan komputasi paralel.

D. Pemodelan Constraint

Permasalahan penjadwalan pelajaran sekolah pada penelitian ini dimodelkan sebagai Constraint Satisfaction Problem (CSP), yaitu permasalahan pencarian solusi yang harus



memenuhi sejumlah constraint yang telah ditentukan. Setiap solusi yang dihasilkan berupa penempatan mata pelajaran pada slot waktu tertentu untuk setiap kelas dengan tetap memperhatikan seluruh batasan yang berlaku.

Pada penelitian ini, constraint yang digunakan hanya hard constraint tanpa menggunakan soft constraint. Hard constraint merupakan aturan yang wajib dipenuhi agar jadwal dinyatakan valid.

D.1 Struktur Data Penjadwalan

Sebelum proses penjadwalan dilakukan, setiap aktivitas pembelajaran direpresentasikan sebagai tuple: (*Mapel, Guru, Jam, Sesi*). Setiap slot pembelajaran direpresentasikan menggunakan pasangan: (*Hari, Slot*). Representasi tersebut memungkinkan setiap slot memiliki identitas unik yang digunakan dalam proses validasi constraint berbasis bitmask.

D.2 Hard Constraint

Hard constraint merupakan aturan yang harus dipenuhi seluruhnya agar jadwal dianggap valid. Apabila salah satu hard constraint dilanggar, maka solusi langsung ditolak oleh algoritma.

Daftar hard constraint yang digunakan :

- a. HC1. Guru tidak boleh mengajar bersamaan.
- b. HC2. Kelas tidak boleh memiliki bentrok jadwal.
- c. HC3. Mata pelajaran tertentu memiliki slot khusus.
- d. HC4. Mata pelajaran multi-sesi harus berada pada hari berbeda.

D.3 Representasi Constraint

Pada penelitian ini seluruh constraint direpresentasikan menggunakan struktur bitmask untuk mempercepat proses validasi selama pencarian solusi. Setiap slot pembelajaran direpresentasikan sebagai sebuah bit pada bilangan biner. Nilai bit bernilai 1 menunjukkan bahwa slot tersebut termasuk ke dalam constraint tertentu atau telah digunakan, sedangkan nilai 0 menunjukkan slot masih tersedia.

Sebagai contoh:

Bitmask Guru = *111000010110*

Menunjukkan bahwa beberapa slot tertentu tidak dapat digunakan oleh guru yang bersangkutan. Dengan representasi tersebut, proses validasi constraint dapat dilakukan menggunakan operasi bitwise yang memiliki kompleksitas waktu konstan sehingga lebih efisien dibandingkan pemeriksaan berbasis iterasi.

E. Implementasi Algoritma Priority-Driven Bitmask Constraint Solver (PDBCS)

Implementasi algoritma PDBCS terdiri atas lima tahap utama, yaitu preprocessing data, perhitungan prioritas mata pelajaran, validasi constraint berbasis bitmask, alokasi slot pembelajaran, dan pencarian solusi menggunakan backtracking.

E.1 Preprocessing Data

Tahap preprocessing bertujuan mengubah data masukan menjadi struktur yang lebih efisien sebelum proses penjadwalan dilakukan. Proses preprocessing meliputi:

- a. Konversi constraint guru ke bitmask.
- b. Konversi constraint mata pelajaran ke bitmask.
- c. Inisialisasi slot yang telah terisi.
- d. Inisialisasi penggunaan mata pelajaran per hari.
- e. Perhitungan jumlah constraint guru.
- f. Perhitungan jumlah constraint mata pelajaran.
- g. Pengurutan mata pelajaran berdasarkan skor prioritas.



E.2 Perhitungan Priority Score

Untuk mengurangi ruang pencarian, setiap mata pelajaran diberikan skor prioritas. Skor dihitung menggunakan empat parameter: jumlah jam pelajaran; jumlah constraint guru; jumlah constraint mata pelajaran; dan status guru khusus.

Formula yang digunakan:

$$Score_i = 1000J_i + 20G_i + 20P_i + 500K_i \quad (1)$$

Keterangan:

- J_i = jumlah jam pelajaran.
- G_i = jumlah constraint guru.
- P_i = jumlah constraint mata pelajaran.
- K_i = guru khusus.

Pemilihan bobot tidak dimaksudkan sebagai parameter optimasi matematis, melainkan sebagai mekanisme heuristik untuk membentuk urutan prioritas penjadwalan yang sesuai dengan tingkat kesulitan masing-masing mata pelajaran.

Tabel 4. Justifikasi Bobot Priority Score

Faktor	Bobot	Alasan
Jumlah Jam Pelajaran (J)	1000	Faktor utama yang menentukan tingkat kesulitan penjadwalan
Guru Khusus (K)	500	Ketersediaan guru terbatas sehingga berpotensi menimbulkan konflik lebih besar
Constraint Guru (G)	20	Faktor pembeda tambahan
Constraint Mata Pelajaran (P)	20	Faktor pembeda tambahan

Nilai bobot pada Priority Score ditentukan menggunakan pendekatan heuristik yang dikembangkan berdasarkan karakteristik permasalahan penjadwalan sekolah serta hasil eksperimen pendahuluan. Bobot 1000 diberikan pada jumlah jam pelajaran untuk memastikan faktor tersebut menjadi komponen dominan dalam proses pengurutan, karena mata pelajaran dengan alokasi jam lebih banyak memiliki tingkat kesulitan penjadwalan yang lebih tinggi. Bobot 500 diberikan pada guru khusus untuk memberikan prioritas tinggi terhadap mata pelajaran yang memiliki keterbatasan sumber daya pengajar. Sementara itu, bobot 20 pada constraint guru dan constraint mata pelajaran digunakan sebagai faktor pembeda tambahan ketika terdapat mata pelajaran dengan jumlah jam dan karakteristik guru yang serupa. Konfigurasi bobot ini dirancang untuk membentuk hierarki prioritas yang stabil sehingga perubahan pada faktor utama tidak dapat dengan mudah diimbangi oleh akumulasi faktor sekunder.

Setelah skor dihitung, seluruh mata pelajaran diurutkan secara menurun sehingga mata pelajaran dengan tingkat kesulitan tertinggi diproses terlebih dahulu.

Algoritma 1. Perhitungan Priority Score :

```

FOR setiap mata pelajaran
    skor ← jam × 1000
    IF guru memiliki constraint
        skor ← skor + (jumlah constraint guru × 20)
    IF mata pelajaran memiliki constraint
        skor ← skor + (jumlah constraint mapel × 20)
    IF guru khusus
        skor ← skor + 500
    
```



END FOR
Urutkan skor secara descending

E.3 Validasi Constraint Berbasis Bitmask

Seluruh constraint direpresentasikan menggunakan bitmask. Representasi ini memungkinkan pengecekan constraint dilakukan menggunakan operasi bitwise dibandingkan iterasi konvensional.

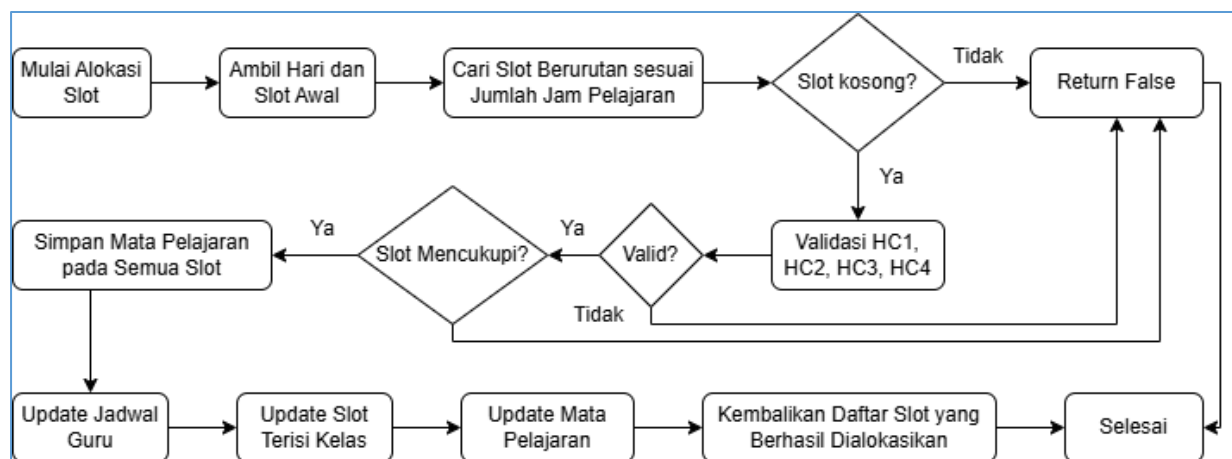
Algoritma 2. Validasi constraint :

```
IF guru sudah mengajar pada slot  
RETURN FALSE  
IF slot termasuk constraint mapel  
RETURN FALSE  
IF slot termasuk constraint guru  
RETURN FALSE  
IF mapel sudah muncul pada hari yang sama  
RETURN FALSE  
RETURN TRUE
```

E.4 Mekanisme Alokasi Slot

Apabila slot dinyatakan valid, algoritma melakukan proses alokasi jadwal. Tahap ini direalisasikan melalui fungsi *alokasikan()*. Proses alokasi meliputi:

1. Menentukan slot berurutan sesuai jumlah jam pelajaran.
2. Memperbarui jadwal guru.
3. Memperbarui slot terisi kelas.
4. Memperbarui penggunaan mata pelajaran harian.



Gambar 4. Mekanisme Alokasi Slot

Gambar 5 menunjukkan proses alokasi slot pada algoritma PDBCS. Proses diawali dengan pencarian slot berurutan sesuai jumlah jam pelajaran yang dibutuhkan. Setiap slot kemudian divalidasi terhadap seluruh hard constraint menggunakan operasi bitmask. Apabila seluruh slot memenuhi constraint dan jumlah slot yang tersedia mencukupi, sistem melakukan pembaruan jadwal guru, slot terisi kelas, dan penggunaan mata pelajaran harian. Sebaliknya, apabila ditemukan konflik atau jumlah slot berurutan tidak mencukupi, proses alokasi dibatalkan dan fungsi mengembalikan nilai gagal untuk memicu mekanisme backtracking.

E.5 Mekanisme Backtracking

Setelah slot berhasil dialokasikan, algoritma melanjutkan proses ke mata pelajaran berikutnya. Apabila pada suatu tahap tidak ditemukan slot yang memenuhi constraint, algoritma melakukan backtracking dengan membatalkan alokasi sebelumnya.

Algoritma 3. Backtracking :

```
FUNCTION Backtrack(i)
  IF seluruh mata pelajaran terjadwal
    RETURN TRUE
  item ← daftarPriority[i]
  FOR setiap kandidat slot
    IF ValidasiBitmask(item,slot)
      Alokasikan(item,slot)
      IF Backtrack(i+1)
        RETURN TRUE
      BatalkanAlokasi(item,slot)
  RETURN FALSE
```

E.6 Algoritma Priority-Driven Bitmask Constraint Solver (PDBCS)

Algoritma Priority-Driven Bitmask Constraint Solver (PDBCS) mengintegrasikan tiga komponen utama, yaitu mekanisme Priority Score untuk menentukan urutan penjadwalan, representasi constraint berbasis Bitmask untuk mempercepat proses validasi konflik, serta mekanisme Backtracking untuk pencarian solusi. Secara umum, algoritma diawali dengan perhitungan Priority Score pada seluruh mata pelajaran, kemudian dilakukan pengurutan berdasarkan nilai prioritas tertinggi. Selanjutnya, proses pencarian solusi dilakukan menggunakan Backtracking dengan validasi constraint berbasis Bitmask pada setiap percobaan penempatan slot.

Algoritma 4. Priority-Driven Bitmask Constraint Solver :

```
FUNCTION PDBCS()
  Hitung Priority Score seluruh mata pelajaran
  Urutkan mata pelajaran berdasarkan Priority Score secara descending
  Inisialisasi seluruh Bitmask Constraint
  status ← Backtrack(0)
  RETURN status
END FUNCTION
```

Pada Algoritma 4, proses pencarian solusi dilakukan melalui pemanggilan fungsi Backtrack() yang dijelaskan pada Algoritma 3. Fungsi tersebut akan menempatkan mata pelajaran sesuai urutan prioritas yang telah ditentukan dan melakukan validasi constraint menggunakan operasi Bitmask sebelum suatu slot dialokasikan. Jika ditemukan konflik atau tidak tersedia slot yang memenuhi constraint, mekanisme Backtracking akan dijalankan untuk mencari alternatif penempatan yang valid.

E.7 Analisis Kompleksitas Algoritma

Secara teoritis, algoritma PDBCS termasuk dalam kategori Constraint Satisfaction Problem berbasis backtracking sehingga kompleksitas waktu terburuknya tetap bersifat eksponensial. Jika terdapat n komponen yang harus dijadwalkan dan setiap komponen memiliki rata-rata s kandidat slot, maka kompleksitas waktu worst-case dapat dinyatakan sebagai $O(s^n)$.



Penggunaan Priority Score tidak mengubah kompleksitas teoritis tersebut, namun berfungsi sebagai heuristic variable ordering yang mengurangi jumlah cabang pencarian yang dieksplorasi dalam praktik. Dengan menempatkan komponen yang memiliki tingkat kesulitan lebih tinggi terlebih dahulu, kemungkinan terjadinya kegagalan pada tahap akhir pencarian dapat dikurangi sehingga jumlah proses backtracking menjadi lebih kecil.

Representasi constraint berbasis bitmask memungkinkan proses validasi konflik dilakukan menggunakan operasi bitwise yang memiliki kompleksitas $O(1)$. Pada pendekatan konvensional, pemeriksaan konflik umumnya memerlukan iterasi terhadap sejumlah slot atau constraint yang relevan sehingga kompleksitas validasi meningkat seiring bertambahnya ukuran data. Oleh karena itu, penggunaan bitmask mampu menurunkan biaya komputasi pada setiap node pencarian tanpa mengubah kompleksitas teoritis algoritma secara keseluruhan.

F. Skenario Pengujian

Pengujian dilakukan untuk mengevaluasi pengaruh penggunaan mekanisme priority score dan representasi constraint berbasis bitmask terhadap kinerja algoritma penjadwalan yang diusulkan. Evaluasi difokuskan pada efisiensi waktu komputasi dalam menghasilkan jadwal yang memenuhi seluruh hard constraint serta tingkat keberhasilan pembentukan jadwal.

Dataset yang digunakan merupakan data nyata dari SMP Negeri 1 Sedati Sidoarjo yang terdiri atas 32 kelas, 60 guru, 14 mata pelajaran, dan 33 slot pembelajaran per minggu. Seluruh pengujian dilakukan menggunakan lingkungan implementasi yang sama sebagaimana dijelaskan pada Bagian C sehingga perbedaan hasil yang diperoleh hanya dipengaruhi oleh mekanisme algoritma yang digunakan.

Algoritma PDBCS menerapkan mekanisme pengacakan (*shuffle*) pada urutan kelas dan urutan hari sebelum proses pencarian solusi dilakukan. Pengacakan ini bertujuan untuk meningkatkan variasi jalur pencarian dan mengurangi kemungkinan algoritma terjebak pada pola pencarian tertentu yang menghasilkan ruang pencarian lebih besar. Konsekuensinya, waktu komputasi dapat berbeda pada setiap eksekusi meskipun menggunakan dataset yang sama. Oleh karena itu, setiap skenario pengujian dijalankan sebanyak 100 kali percobaan independen untuk memperoleh hasil yang lebih representatif. Jumlah 100 percobaan dipilih untuk mengurangi pengaruh variasi jalur pencarian yang muncul akibat mekanisme pengacakan dan menghasilkan estimasi performa yang lebih stabil.

Empat skenario pengujian yang digunakan ditunjukkan pada Tabel 5.

Tabel 5. Skenario Pengujian

Kode	Priority Score	Bitmask	Backtracking
S1	X	X	✓
S2	✓	X	✓
S3	X	✓	✓
S4	✓	✓	✓

Keterangan:

- S1 merupakan implementasi backtracking dasar tanpa priority score dan tanpa bitmask.
- S2 menggunakan priority score untuk menentukan urutan penjadwalan mata pelajaran, tetapi masih menggunakan validasi constraint konvensional.
- S3 menggunakan validasi constraint berbasis bitmask tanpa mekanisme priority score.
- S4 merupakan implementasi lengkap algoritma Priority-Driven Bitmask Constraint Solver yang menggabungkan priority score, bitmask, dan backtracking.



Untuk setiap skenario, sistem dijalankan sebanyak 100 kali percobaan menggunakan dataset yang sama. Hasil pengujian dievaluasi menggunakan lima metrik utama sebagaimana ditunjukkan pada Tabel 6.

Tabel 6. Metrik Evaluasi

Kode	Metrik	Deskripsi
M1	Waktu Tercepat	Waktu minimum yang diperoleh dari 100 percobaan
M2	Waktu Terlama	Waktu maksimum yang diperoleh dari 100 percobaan
M3	Waktu Rata-rata	Rata-rata waktu komputasi seluruh percobaan
M4	Standar Deviasi	Tingkat variasi waktu komputasi antar percobaan
M5	Tingkat Keberhasilan	Persentase jadwal yang berhasil dibentuk dan memenuhi seluruh hard constraint.

Selain itu, dilakukan analisis peningkatan performa terhadap skenario dasar (S1) menggunakan persentase percepatan (speedup) yang dihitung menggunakan Persamaan :

$$Improvement (\%) = \frac{T_{baseline} - T_{proposed}}{T_{baseline}} \times 100\% \quad (2)$$

dengan:

- $T_{baseline}$ adalah waktu rata-rata pada skenario S1;
- $T_{proposed}$ adalah waktu rata-rata pada skenario yang dibandingkan.

Evaluasi penelitian difokuskan pada efisiensi waktu komputasi dan tingkat keberhasilan pembentukan jadwal yang memenuhi seluruh hard constraint.

HASIL DAN PEMBAHASAN

Bagian ini menyajikan hasil implementasi dan pengujian Algoritma Priority-Driven Bitmask Constraint Solver (PDBCS) pada kasus penjadwalan pelajaran di SMP Negeri 1 Sedati Sidoarjo. Evaluasi dilakukan untuk mengukur pengaruh penggunaan mekanisme priority score dan representasi constraint berbasis bitmask terhadap efisiensi proses penjadwalan. Pengujian dilakukan menggunakan empat skenario berbeda yang dijalankan sebanyak 100 kali percobaan untuk memperoleh hasil yang representatif. Pembahasan difokuskan pada analisis waktu komputasi, pengaruh Priority Score, pengaruh Bitmask, serta evaluasi kombinasi keduanya dalam algoritma PDBCS.

A. Hasil Pengujian Waktu Komputasi

Pengujian dilakukan menggunakan empat skenario yang telah dijelaskan pada Metode Penelitian. Setiap skenario dijalankan sebanyak 100 kali percobaan menggunakan dataset yang sama, yaitu 32 kelas, 60 guru, 14 mata pelajaran, dan 33 slot pembelajaran per minggu. Seluruh percobaan dilakukan pada lingkungan perangkat keras dan perangkat lunak yang sama untuk memastikan konsistensi hasil pengujian.

Tabel 7. Hasil Pengujian Waktu Komputasi

Skenario	M1	M2	M3	M4	M5
S1	00:21:57	01:13:27	00:46:45	00:11:28	100%
S2	00:19:34	00:55:19	00:37:41	00:08:42	100%
S3	00:05:14	00:17:38	00:12:52	00:03:01	100%
S4	00:01:47	00:15:33	00:07:05	00:02:17	100%



Berdasarkan Tabel 7, seluruh skenario berhasil menghasilkan jadwal yang memenuhi seluruh hard constraint dengan tingkat keberhasilan 100%. Meskipun demikian, terdapat perbedaan yang cukup signifikan pada waktu komputasi yang dibutuhkan untuk menyelesaikan proses penjadwalan. Skenario S1 yang menggunakan backtracking konvensional tanpa priority score dan tanpa bitmask menghasilkan waktu rata-rata tertinggi, yaitu 46 menit 45 detik. Sebaliknya, skenario S4 yang menerapkan kombinasi priority score dan bitmask menghasilkan waktu rata-rata terendah, yaitu 7 menit 5 detik.

Perbedaan waktu komputasi tersebut menunjukkan bahwa mekanisme prioritas dan representasi constraint memiliki pengaruh terhadap efisiensi proses pencarian solusi. Selain itu, standar deviasi pada skenario S4 juga lebih kecil dibandingkan skenario lainnya, yang menunjukkan bahwa performa algoritma relatif lebih stabil pada berbagai percobaan.

Variasi waktu komputasi yang muncul pada setiap percobaan disebabkan oleh penggunaan mekanisme pengacakan (shuffle) pada urutan kelas dan urutan hari selama proses pencarian solusi. Pengacakan tersebut menghasilkan jalur pencarian yang berbeda pada setiap eksekusi sehingga jumlah proses backtracking yang terjadi dapat berubah. Oleh karena itu, evaluasi dilakukan menggunakan 100 kali percobaan dan dianalisis berdasarkan nilai minimum, maksimum, rata-rata, serta standar deviasi untuk memperoleh gambaran performa algoritma yang lebih representatif.

Untuk memperjelas perbandingan antar skenario, Gambar 5 menampilkan rata-rata waktu komputasi yang diperoleh pada masing-masing metode.



Gambar 5. Perbandingan waktu rata-rata.

Berdasarkan Gambar 5 terlihat bahwa penggunaan bitmask memberikan penurunan waktu komputasi yang lebih besar dibandingkan penggunaan priority score secara terpisah. Namun demikian, waktu komputasi terendah diperoleh ketika kedua mekanisme digunakan secara bersamaan pada algoritma PDBCS. Hasil ini mengindikasikan bahwa priority score dan bitmask memiliki kontribusi yang saling melengkapi dalam meningkatkan efisiensi proses penjadwalan.

B. Analisis Pengaruh Priority Score

Priority Score pada algoritma yang diusulkan digunakan untuk menentukan urutan penempatan mata pelajaran berdasarkan tingkat kesulitannya, yang dihitung dari jumlah jam pelajaran, jumlah constraint guru, constraint mata pelajaran, serta kebutuhan khusus guru. Mata pelajaran dengan tingkat kesulitan lebih tinggi ditempatkan terlebih dahulu agar konflik dapat terdeteksi lebih awal dan ruang pencarian menjadi lebih kecil.

Pengaruh Priority Score dianalisis dengan membandingkan skenario S1 (backtracking konvensional) dan S2 (backtracking dengan Priority Score). Hasil pengujian pada Tabel 7 dan Gambar 5 menunjukkan bahwa penggunaan Priority Score mampu meningkatkan performa sebesar 19,4% dibandingkan metode konvensional. Peningkatan ini terjadi karena urutan penjadwalan yang lebih optimal mengurangi jumlah proses backtracking dan cabang pencarian yang harus dieksplorasi.

Meskipun memberikan peningkatan yang cukup signifikan, kontribusi Priority Score masih lebih rendah dibandingkan penggunaan bitmask. Temuan ini menunjukkan bahwa Priority Score

berperan dalam mengurangi ruang pencarian, sedangkan percepatan utama proses penjadwalan tetap dipengaruhi oleh efisiensi mekanisme validasi constraint. Dengan demikian, Priority Score berkontribusi terhadap peningkatan performa algoritma, namun bukan faktor dominan dalam percepatan proses penjadwalan.

C. Analisis Pengaruh Bitmask

Representasi constraint berbasis bitmask digunakan untuk mempercepat proses validasi jadwal dengan merepresentasikan setiap slot sebagai bit biner sehingga pemeriksaan konflik dapat dilakukan menggunakan operasi bitwise. Pengaruh penggunaan bitmask dianalisis melalui perbandingan skenario S1 dan S3 yang sama-sama menggunakan backtracking, namun hanya S3 yang menerapkan bitmask.

Hasil pengujian pada Tabel 7 dan Gambar 5 menunjukkan bahwa penggunaan bitmask mampu menurunkan rata-rata waktu komputasi dari 46 menit 45 detik menjadi 12 menit 52 detik, dengan peningkatan performa sebesar 72,5%. Percepatan ini terjadi karena validasi constraint tidak lagi memerlukan iterasi terhadap seluruh slot jadwal, melainkan cukup menggunakan operasi logika seperti AND dan OR untuk mendeteksi konflik secara cepat.

Selain mempercepat validasi, bitmask juga mengurangi kebutuhan akses ke struktur data yang kompleks sehingga jumlah operasi pada setiap proses pencarian menjadi lebih kecil. Hasil ini menunjukkan bahwa optimasi representasi data memberikan kontribusi yang sangat besar terhadap performa algoritma penjadwalan dan efektif diterapkan pada algoritma backtracking untuk meningkatkan efisiensi proses pencarian solusi..

D. Analisis Kombinasi Priority Score dan Bitmask pada PDBCS

Algoritma Priority-Driven Bitmask Constraint Solver (PDBCS) menggabungkan Priority Score untuk mengurangi ruang pencarian dan Bitmask untuk mempercepat validasi constraint melalui operasi bitwise. Kombinasi kedua mekanisme ini memungkinkan proses pencarian solusi menjadi lebih efisien dibandingkan backtracking konvensional.

Hasil pengujian pada Tabel 7 dan Gambar 5 menunjukkan bahwa PDBCS menurunkan rata-rata waktu komputasi dari 46 menit 45 detik menjadi 7 menit 5 detik, dengan peningkatan performa sebesar 84,9% tanpa mengurangi tingkat keberhasilan penyusunan jadwal. Secara terpisah, Priority Score meningkatkan performa sebesar 19,4%, sedangkan Bitmask sebesar 72,5%, yang menunjukkan bahwa sebagian besar percepatan berasal dari efisiensi validasi constraint berbasis Bitmask.

Selain lebih cepat, PDBCS juga menunjukkan performa yang lebih stabil, ditandai dengan nilai standar deviasi yang lebih rendah dibandingkan metode konvensional. Secara keseluruhan, hasil penelitian membuktikan bahwa integrasi Priority Score dan Bitmask merupakan pendekatan yang efektif untuk meningkatkan efisiensi dan konsistensi algoritma penjadwalan pelajaran sekolah otomatis tanpa melanggar seluruh hard constraint yang ditetapkan.

E. Analisis Efisiensi Memori

Selain meningkatkan kecepatan validasi constraint, representasi Bitmask juga memiliki keuntungan dari sisi penggunaan memori. Informasi ketersediaan slot disimpan dalam bentuk bilangan biner sehingga beberapa kondisi dapat direpresentasikan dalam satu variabel integer. Dibandingkan penyimpanan berbasis array atau struktur data yang memerlukan banyak elemen, pendekatan ini berpotensi mengurangi kebutuhan memori terutama pada permasalahan dengan jumlah slot yang besar.

F. Analisis Skalabilitas dan Stabilitas

Hasil pengujian menunjukkan bahwa mekanisme Priority Score dan Bitmask tetap memberikan keuntungan pada dataset dengan jumlah komponen yang relatif besar, yaitu 32



kelas, 60 guru, dan 33 slot pembelajaran. Secara teoritis, manfaat Priority Score akan semakin terlihat pada ukuran permasalahan yang lebih besar karena jumlah cabang pencarian yang dapat dieliminasi meningkat seiring bertambahnya jumlah komponen penjadwalan. Demikian pula, penggunaan Bitmask memungkinkan biaya validasi constraint tetap konstan pada setiap node pencarian sehingga pertumbuhan waktu komputasi dapat ditekan dibandingkan pendekatan konvensional.

Meskipun pengujian pada penelitian ini menggunakan satu dataset utama, konsistensi peningkatan performa pada seluruh percobaan menunjukkan bahwa mekanisme Priority Score dan Bitmask bekerja secara independen terhadap urutan pencarian solusi yang terbentuk. Oleh karena itu, pendekatan ini berpotensi mempertahankan karakteristik peningkatan performa pada dataset sekolah lain yang memiliki karakteristik constraint yang sejenis, meskipun diperlukan pengujian lebih lanjut pada berbagai dataset untuk memvalidasi temuan tersebut.

G. Perbandingan dengan Penelitian Terdahulu

Berbagai penelitian sebelumnya telah mengusulkan pendekatan berbeda untuk menyelesaikan permasalahan timetabling, termasuk Genetic Algorithm, Particle Swarm Optimization (PSO), Hyper-Heuristic, dan Constraint Programming. Sebagian besar penelitian berfokus pada pencarian solusi yang memenuhi constraint dan meningkatkan kualitas jadwal, namun tidak secara khusus mengoptimalkan efisiensi proses validasi constraint pada algoritma pencarian. Oleh karena itu, hasil penelitian ini dibandingkan secara kualitatif dengan beberapa penelitian terdahulu untuk menunjukkan posisi kontribusi algoritma PDBCS.

Tabel 8. Perbandingan dengan Penelitian Terdahulu

Penelitian	Metode	Valid Schedule	Optimasi Pencarian	Optimasi Validasi Constraint	Waktu Komputasi
Tassopoulos (2023)	PSO	Ya	Ya	Tidak	Tidak dilaporkan
Macarubbo (2024)	Genetic Algorithm	Ya	Ya	Tidak	Tidak dilaporkan
Ismail (2024)	Hyper-Heuristic	Ya	Ya	Tidak	Tidak dilaporkan
Joshi & Agarwal (2023)	CSP	Ya	Terbatas	Tidak	Tidak dilaporkan
PDBCS	Priority Score + Bitmask + Backtracking	Ya	Ya	Ya	84,9% lebih cepat

Berdasarkan Tabel 8, penelitian terdahulu umumnya berfokus pada optimasi strategi pencarian solusi melalui metaheuristik atau heuristic search. Sebaliknya, algoritma PDBCS mengombinasikan optimasi ruang pencarian melalui Priority Score dan optimasi validasi constraint melalui representasi Bitmask. Kombinasi kedua mekanisme tersebut memungkinkan peningkatan efisiensi proses penjadwalan tanpa mengubah karakteristik dasar algoritma backtracking.

H. Implikasi Praktis bagi Sekolah

Peningkatan performa sebesar 84,9% menunjukkan bahwa waktu komputasi yang sebelumnya diperlukan untuk menghasilkan jadwal dapat dikurangi menjadi sekitar seperenam dari metode backtracking konvensional. Dalam konteks operasional sekolah, pengurangan waktu komputasi tersebut memungkinkan proses penyusunan jadwal dilakukan secara lebih responsif ketika terjadi perubahan data, seperti pergantian guru, perubahan beban mengajar, atau penambahan kelas. Dengan demikian, sistem tidak hanya memberikan keuntungan dari sisi teknis, tetapi juga meningkatkan fleksibilitas pengelolaan akademik sekolah.



KESIMPULAN

Penelitian ini berhasil mengembangkan algoritma Priority-Driven Bitmask Constraint Solver (PDBCS) yang mengintegrasikan Priority Score, Bitmask, dan Backtracking untuk penjadwalan pelajaran sekolah otomatis. Pengujian menggunakan data SMP Negeri 1 Sedati menunjukkan bahwa algoritma mampu memenuhi seluruh hard constraint dengan tingkat keberhasilan 100% serta meningkatkan efisiensi komputasi hingga 84,9% dibandingkan backtracking konvensional.

Kontribusi utama penelitian ini adalah penggabungan mekanisme pengurangan ruang pencarian melalui Priority Score dan percepatan validasi constraint melalui Bitmask dalam satu kerangka Constraint Satisfaction Problem (CSP). Hasilnya menunjukkan bahwa kedua mekanisme saling melengkapi sehingga mampu meningkatkan performa penjadwalan secara signifikan. Dari sisi praktis, algoritma ini memungkinkan proses penyusunan dan penyesuaian jadwal dilakukan lebih cepat ketika terjadi perubahan data guru, kelas, atau beban mengajar.

Meskipun memberikan hasil yang baik, penelitian ini masih berfokus pada hard constraint. Penelitian selanjutnya dapat mengakomodasi soft constraint, seperti preferensi guru dan distribusi beban mengajar, serta menguji algoritma pada dataset yang lebih besar dan beragam. Secara keseluruhan, PDBCS merupakan alternatif yang efektif untuk menghasilkan jadwal yang valid dengan efisiensi komputasi yang tinggi.

DAFTAR PUSTAKA

- [1] R. K. Thakur, N. K. Agrawal, and P. Kumar, "A practical approach to college timetable scheduling," *Mathematical Modeling and Computing*, vol. 11, no. 3, pp. 710–719, 2024, doi: 10.23939/mmc2024.03.710.
- [2] N. R. Joshi and T. V. Agarwal, "Optimizing University Course Timetabling Using Constraint Satisfaction Models," *International Advanced Journal of Science and Engineering*, vol. 10, no. 4, pp. 5–8, 2023, doi: 10.71086/IAJSE/V10I4/IAJSE1032.
- [3] M. Davison, A. Kheiri, and K. G. Zografos, "Modelling and solving the university course timetabling problem with hybrid teaching considerations," *Journal of Scheduling*, vol. 28, pp. 195–215, 2025, doi: 10.1007/s10951-024-00817-w.
- [4] A. A. Sharif, N. A. M. Isa, and M. M. Deris, "A Survey of the State-of-the-Art of Optimisation Methodologies in School Timetabling Problems," *Expert Systems with Applications*, vol. 165, 2021, doi: 10.1016/j.eswa.2020.113943.
- [5] I. X. Tassopoulos et al., "An Effective Local Particle Swarm Optimization-Based Algorithm for Solving the School Timetabling Problem," *Algorithms*, vol. 16, no. 6, p. 291, 2023, doi: 10.3390/a16060291.
- [6] M. M. R. Macarubbo et al., "Development of a Web-based Course Timetabling System Based on an Enhanced Genetic Algorithm," *Procedia Computer Science*, vol. 234, pp. 1714–1721, 2024, doi: 10.1016/j.procs.2024.03.177.
- [7] A. A. Ismail et al., "Generic University Examination Timetabling System with Steepest-Ascent Hill Climbing Hyper-Heuristic Algorithm," *Procedia Computer Science*, vol. 234, pp. 584–591, 2024, doi: 10.1016/j.procs.2024.03.043.
- [8] T. Müller, H. Rudová, and Z. Müllerová, "Real-world University Course Timetabling at the International Timetabling Competition 2019," *Journal of Scheduling*, vol. 28, pp. 247–267, 2025, doi: 10.1007/s10951-023-00801-w.
- [9] S. Ceschia, L. Di Gaspero, and A. Schaerf, "Educational Timetabling: Problems, Benchmarks, and State-of-the-Art Results," *Annals of Operations Research*, 2022, doi: 10.48550/arXiv.2201.07525.



- [9] S. Ceschia, L. Di Gaspero, and A. Schaerf, "Educational Timetabling: Problems, Benchmarks, and State-of-the-Art Results," *arXiv preprint arXiv:2201.07525*, 2022, doi: 10.48550/arXiv.2201.07525.
- [10] S. K. Singh et al., "Decentralized Exam Timetabling: A Solution for Conducting Exams During Pandemics," *Socio-Economic Planning Sciences*, vol. 92, 2024, doi: 10.1016/j.seps.2024.101802.
- [11] E. Demirović and N. Musliu, "Modeling High School Timetabling with Bitvectors," *Annals of Operations Research*, vol. 252, no. 2, pp. 215–238, 2017, doi: 10.1007/s10479-016-2220-6.
- [12] T. Geibinger, F. Mischek, and N. Musliu, "Constraint Logic Programming for Real-World Test Laboratory Scheduling," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 7, pp. 6358–6366, 2021, doi: 10.1609/aaai.v35i7.16789.
- [13] A. Bădică, C. Bădică, and M. Ivanović, "Block Structured Scheduling Using Constraint Logic Programming," *AI Communications*, vol. 33, no. 1, pp. 41–57, 2020, doi: 10.3233/AIC-200650.
- [14] X. Gu, M. Krish, S. Sohail, S. Thakur, F. Sabrina, and Z. Fan, "From Integer Programming to Machine Learning: A Technical Review on Solving University Timetabling Problems," *Computation*, vol. 13, no. 1, p. 10, 2025, doi: 10.3390/computation13010010.
- [15] P. Vieira, V. Conte, and P. Siqueira, "Class Timetable Allocation in Higher Education Using Binary Integer Programming: A Case Study in the Academic Department of Mathematics at a Public University in Paraná, Brazil," *Journal of Education and Learning*, vol. 15, no. 3, pp. 24–34, 2025, doi: 10.5539/jel.v15n3p24.
- [16] M. Al-Rashed et al., "An Integer Programming Approach to Solve One Semester's Lectures Timetabling Problem with a Case Study for the Faculty of Medicine, Kuwait University," *Operations Research Forum*, vol. 7, Art. no. 50, 2026, doi: 10.1007/s43069-026-00637-y.

