

Penerapan Algoritma Bracktracking Dalam Penyelesaian Masalah Sistem Pakar

Riska Ayusi¹, Aulia Triantika², Zanziqbar Alaydrus³

¹ Teknik Informatika, STMIK Pelita Nusantara, Medan (riskaayusi73@gmail.com)

² Teknik Informatika, STMIK Pelita Nusantara, Medan, (auliatriantika@gmail.com)

³ Teknik Informatika, STMIK Pelita Nusantara, Medan, (zanziqbaralaydrus@gmail.com)

[Diserahkan: 15 Desember 2024, Direvisi: 21 Desember 2024, Diterima: 10 Januari 2025]

Corresponding Author: Riska Ayusi (email: riskaayusi73@gmail.com)



Kata Kunci: Algoritma Bracktracking, penyelesaian masalah, kombinatorik, permainan logika, teka teki matematis.

JUSINFO: Jurnal Sains dan Informatika is licensed under a Creative Commons Attribution-NonCommercial 4.0 International (CC BY-NC 4.0).

Abstrak : Penelitian ini berfokus pada penerapan algoritma backtracking sebagai metode penyelesaian berbagai permasalahan kompleks di bidang komputasi, permainan, dan diagnosis sistem pakar. Algoritma backtracking digunakan untuk memecahkan masalah seperti permainan teka-teki (Minesweeper, N-Queen Problem, Math Maze, Knight Tour, Nonogram, dan Sudoku), sistem checkpoint pada permainan edukatif, serta optimalisasi diagnosis penyakit. Berbagai pendekatan diperbandingkan, termasuk brute force dan soft computing, menunjukkan bahwa backtracking memiliki keunggulan signifikan dalam efisiensi waktu dan akurasi solusi.

Dalam permainan Minesweeper, backtracking berhasil memangkas ruang pencarian dengan kompleksitas $O(2^n \cdot n)$, dibandingkan brute force dengan $O(n \cdot n! / (n-d)!)!$. Pada N-Queen Problem, algoritma ini menghasilkan solusi optimal untuk papan berukuran kecil. Penyelesaian permainan Knight Tour menunjukkan efisiensi pada papan menengah, sementara Sudoku dan Nonogram menghasilkan akurasi 100% dengan perbandingan kecepatan terhadap metode lain. Di bidang sistem pakar diagnosis, penerapan backtracking pada metode backward chaining meningkatkan efisiensi sebesar 88%.

Hasil penelitian menunjukkan bahwa algoritma backtracking efektif dan akurat dalam memecahkan permasalahan kombinatorik dan sistem kompleks. Namun, efisiensi algoritma berkurang pada skala besar, sehingga diperlukan optimasi lebih lanjut untuk menangani permasalahan dengan kompleksitas tinggi.

PENDAHULUAN

Algoritma backtracking adalah metode penyelesaian masalah secara sistematis dengan menggunakan pendekatan rekursif dan eksplorasi depth-first search (DFS). Teknik ini bekerja dengan cara mencoba semua kemungkinan solusi, dan jika suatu langkah tidak memenuhi kriteria solusi, algoritma akan kembali mundur (backtrack) ke langkah sebelumnya untuk mencoba alternatif lain.

Dalam beberapa masalah yang bersifat kombinatorik, algoritma ini memungkinkan pemangkasan jalur yang tidak mengarah ke solusi melalui teknik pruning. Efektivitas algoritma ini terlihat dalam permasalahan seperti permainan catur (N-Queen Problem, Knight Tour), teka-teki angka (Sudoku, Nonogram), pencarian jalur (Math Maze, labirin), serta optimasi sistem diagnosis penyakit.

Contoh implementasi:

Pada N-Queen Problem, backtracking mengeksplorasi semua kemungkinan penempatan ratu secara rekursif hingga solusi valid ditemukan.

Pada permainan Minesweeper, backtracking digunakan untuk mencari solusi lebih efisien dibandingkan brute force dengan memangkas solusi yang tidak mungkin valid.

Masalah Backtracking

Masalah yang diselesaikan dengan backtracking umumnya melibatkan ruang solusi yang sangat besar dan memiliki karakteristik sebagai berikut:

Eksplorasi Solusi Parsial:

Solusi dibangun langkah demi langkah.

Validasi Solusi:

Setiap langkah dievaluasi apakah memenuhi kriteria.

Pengembalian Mundur:

Jika langkah tidak valid, algoritma mundur ke langkah sebelumnya dan mencoba jalur lain.

Dari 10 jurnal yang dirangkum, berikut contoh permasalahan yang diselesaikan:

1. N-Queen Problem:

Menempatkan N ratu pada papan catur $N \times N$ tanpa saling menyerang (Rahmawati et al., 2023).

2. Permainan Sudoku:

Menyelesaikan teka-teki angka 9×9 dengan aturan baris, kolom, dan blok (Hanafi et al., 2018).

3. Knight Tour:

Mencari jalur untuk kuda catur agar mengunjungi setiap kotak papan hanya satu kali (Novianto & Pramadhana, 2019).

4. Math Maze dan Labirin:

Menemukan jalur optimal dalam teka-teki labirin (Teneng et al., 2010; Bharlirus, 2019).

5. Permainan Minesweeper:

Menemukan kotak bebas bom secara efisien (Tanoto, 2022).

6. Nonogram:

Menyelesaikan teka-teki logika dengan petunjuk angka di baris dan kolom (Sidiq et al., 2023).

7. Sistem Pakar Diagnosis:

Meningkatkan efisiensi metode backward chaining dalam diagnosis DBD (Kiswanto et al., 2024).

Solusi yang diberikan Backtracking

Algoritma backtracking memberikan solusi dengan langkah-langkah berikut:

1. Membangun Solusi Parsial:

Menyusun solusi langkah demi langkah, dimulai dari keadaan awal.

2. Validasi Langkah:

Memeriksa apakah solusi parsial memenuhi kriteria validitas.

3. Backtrack:

Jika solusi parsial tidak valid, algoritma akan mundur ke langkah sebelumnya untuk mencoba pilihan lain.

4. Solusi Optimal:

Proses ini berlanjut hingga solusi optimal atau semua solusi ditemukan.

Contoh solusi dari beberapa jurnal:

1. Pada N-Queen Problem (Rahmawati et al., 2023), backtracking menemukan solusi valid seperti [2,4,1,3] untuk $N=4$.

2. Pada Knight Tour (Novianto & Pramadhana, 2019), backtracking menemukan jalur kunjungan kuda catur pada papan ukuran 5×5 hingga 12×12 .

3. Pada Sudoku (Hanafi et al., 2018), algoritma backtracking menyelesaikan grid Sudoku dengan

akurasi 100%.

4. Pada Minesweeper (Tanoto, 2022), backtracking meningkatkan efisiensi dengan kompleksitas $O(2^n \cdot n)$ dibandingkan brute force.

Solusi ini menunjukkan bahwa backtracking dapat menyelesaikan masalah kompleks secara sistematis, meskipun pada kasus besar, waktu eksekusi dapat meningkat drastis.

Definisi Backtracking

Algoritma backtracking adalah metode eksplorasi solusi menggunakan pendekatan rekursif dan depth-first search (DFS). Algoritma ini bekerja dengan membangun solusi secara bertahap, dan jika solusi tidak memenuhi kriteria, algoritma akan mundur ke langkah sebelumnya untuk mencoba jalur lain.

Karakteristik Backtracking:

1. Solusi Parsial: Solusi dibangun secara bertahap.
2. Evaluasi Validitas: Setiap langkah diperiksa kebenarannya.
3. Backtrack: Jika solusi tidak valid, algoritma kembali ke langkah sebelumnya dan mencoba alternatif lain.
4. Kompleksitas: Bergantung pada jumlah kemungkinan langkah dan ukuran ruang solusi.

Contoh Definisi dalam Konteks Permasalahan:

1. Pada Sudoku: Algoritma ini mencoba semua angka 1–9 untuk setiap sel kosong dan memvalidasi angka tersebut berdasarkan aturan permainan.
2. Pada Labirin: Backtracking mengeksplorasi semua jalur yang mungkin dari titik awal hingga titik akhir. Jika menemui jalan buntu, algoritma kembali ke simpul sebelumnya.

TINJAUAN LITERATUR

Dalam penelitian ini, algoritma backtracking dipilih sebagai solusi utama untuk menyelesaikan berbagai permasalahan, baik di bidang permainan, pencarian jalur, teka-teki, maupun optimasi diagnosis berbasis sistem pakar. Berikut adalah tinjauan literatur terkait algoritma backtracking dari 10 jurnal yang telah dirangkum.

1. Algoritma Backtracking

Algoritma backtracking adalah metode eksplorasi solusi yang menggunakan pendekatan rekursif dan Depth First Search (DFS) untuk mencoba semua kemungkinan solusi secara sistematis. Jika suatu langkah tidak valid, algoritma akan mundur (backtrack) ke langkah sebelumnya untuk mencoba solusi lain. Proses ini berlangsung hingga solusi optimal ditemukan atau semua solusi dievaluasi.

Dalam penelitian:

- Tanoto (2022) menunjukkan bahwa backtracking lebih efisien dibandingkan metode brute force dalam menyelesaikan Minesweeper, dengan kompleksitas waktu $O(2^n \cdot n)$ dibandingkan brute force $O(n \cdot n! / (n-d)!)$.
- Rahmawati et al. (2023) menerapkan backtracking untuk menyelesaikan N-Queen Problem dengan mengeksplorasi semua posisi ratu di papan catur secara rekursif hingga solusi valid ditemukan.
- Hanafi et al. (2018) menggunakan backtracking untuk menyelesaikan Sudoku, di mana setiap sel di grid diperiksa satu per satu, memastikan angka yang ditempatkan memenuhi aturan baris, kolom, dan blok.

2. Penerapan Backtracking pada Permainan

Algoritma backtracking sering digunakan untuk menyelesaikan permasalahan dalam permainan logika dan strategi karena sifatnya yang sistematis. Beberapa studi terkait penerapan ini meliputi:

- Permainan Math Maze: Backtracking digunakan untuk mencari jalur optimal dalam menyelesaikan teka-teki matematika berbasis grid (Teneng et al., 2010).
- Permainan Knight Tour: Backtracking mengeksplorasi semua kemungkinan jalur gerakan kuda pada papan catur sehingga setiap kotak dikunjungi sekali (Novianto & Pramadhana, 2019).

- Nonogram: Penelitian Sidiq et al. (2023) menunjukkan backtracking efektif dalam menyelesaikan teka-teki logika Nonogram, meskipun waktu eksekusi meningkat seiring bertambahnya ukuran grid.
- Minesweeper: Tanoto (2022) menerapkan backtracking untuk memangkas ruang solusi dalam permainan Minesweeper, menghasilkan solusi dengan lebih cepat. Penerapan ini menunjukkan bahwa backtracking mampu menyelesaikan masalah kombinatorik dan eksplorasi jalur dengan keakuratan tinggi.

3. Backtracking untuk Sistem Pakar dan Diagnosis

Algoritma backtracking juga diimplementasikan dalam sistem pakar untuk mengoptimalkan proses pencarian solusi:

- Backward Chaining: Dalam diagnosis penyakit demam berdarah (DBD), backtracking dikombinasikan dengan metode backward chaining untuk meminimalkan proses penelusuran aturan yang tidak relevan. Penelitian Kiswanto et al. (2024) menunjukkan peningkatan efisiensi sebesar 88% dibandingkan pendekatan biasa.
- Pencarian Data Skripsi: Backtracking digunakan untuk meningkatkan kecepatan pencarian data skripsi serupa, dengan hasil kemiripan yang dihitung secara efisien (Juliantus & Sitorus, 2019). Penerapan ini membuktikan bahwa backtracking dapat digunakan tidak hanya dalam permainan, tetapi juga dalam sistem berbasis aturan dan optimasi proses.

4. Kompleksitas dan Efisiensi Backtracking

Algoritma backtracking memiliki kompleksitas yang bervariasi tergantung pada ukuran ruang solusi dan teknik optimasi yang digunakan. Dalam penelitian:

- Tanoto (2022) menunjukkan bahwa backtracking memiliki kompleksitas $O(2^n \cdot n)$ dengan pruning untuk Minesweeper, lebih baik dibandingkan brute force.
- Hanafi et al. (2018) mencatat bahwa penyelesaian Sudoku dengan backtracking memberikan akurasi 100%, tetapi memerlukan optimasi untuk tingkat kesulitan yang lebih tinggi.
- Rahmawati et al. (2023) menyelesaikan N-Queen Problem secara efisien untuk ukuran kecil ($N=4$), tetapi membutuhkan waktu yang lebih lama pada skala yang lebih besar. Optimasi seperti pruning, memoization, dan teknik branch and bound sering digunakan untuk mempercepat eksplorasi dan menghindari perhitungan berulang.

5. Perbandingan Backtracking dengan Metode Lain

Penelitian ini juga membandingkan algoritma backtracking dengan metode lain untuk menilai efisiensi dan keakuratan:

- Backtracking vs Soft Computing: Dalam penelitian Sidiq et al. (2023), backtracking memiliki stabilitas lebih baik dalam penyelesaian Nonogram, meskipun soft computing lebih cepat dalam beberapa skenario.

Backtracking vs Brute Force: Penelitian Tanoto (2022) membuktikan bahwa backtracking lebih unggul dibandingkan brute force dalam penyelesaian Minesweeper karena teknik pruning yang mampu memangkas jalur solusi yang tidak valid.

Hasil perbandingan ini menunjukkan bahwa backtracking unggul dalam stabilitas dan akurasi, tetapi sering memerlukan optimasi agar lebih efisien pada ruang solusi yang besar.

Kesimpulan Tinjauan Literatur

Tinjauan literatur dari 10 jurnal ini menunjukkan bahwa algoritma backtracking adalah pendekatan yang efektif untuk menyelesaikan masalah kombinatorik, eksplorasi jalur, dan sistem berbasis aturan. Dengan pendekatan rekursif dan eksplorasi depth-first search (DFS), algoritma ini mampu menemukan solusi secara sistematis dan akurat. Namun, backtracking memiliki tantangan dalam efisiensi waktu pada skala besar, sehingga teknik optimasi seperti pruning dan branch and bound diperlukan untuk mempercepat proses.

Penerapan backtracking telah berhasil diimplementasikan dalam:

- Permainan logika:
N-Queen Problem, Minesweeper, Sudoku, Math Maze, Nonogram, dan Knight Tour.

- Sistem optimasi diagnosis:
Backward chaining dalam diagnosis penyakit.
- Pencarian data:
Efisiensi pencarian data serupa.
Dengan keunggulan dalam akurasi dan stabilitas, algoritma backtracking menjadi solusi penting dalam menyelesaikan berbagai permasalahan dengan ruang solusi yang kompleks.
Dengan keunggulan dalam akurasi dan stabilitas, algoritma backtracking menjadi solusi penting dalam menyelesaikan berbagai permasalahan dengan ruang solusi yang kompleks.

METODE PENELITIAN

Identifikasi masalah penelitian ini mengidentifikasikan berbagai permasalahan yang dapat diselesaikan menggunakan algoritma backtracking, yang merupakan metode eksplorasi solusi secara rekursif dengan mengevaluasi semua kemungkinan langkah dan mundur (backtrack) jika solusi yang diuji tidak valid.

Permasalahan yang teridentifikasi meliputi:

- N-Queen Problem: Menempatkan N ratu pada papan catur tanpa saling menyerang (Rahmawati et al., 2023).
- Sudoku: Penyelesaian teka-teki angka dengan aturan baris, kolom, dan blok (Hanafi et al., 2018).
- Permainan Minesweeper: Mencari kotak bebas bom dengan efisiensi lebih baik dibandingkan brute force (Tanoto, 2022).
- Knight Tour: Menemukan jalur pergerakan kuda catur agar semua kotak pada papan dikunjungi sekali (Novianto & Pramadhana, 2019).
- Permainan Labirin dan Math Maze: Menentukan jalur optimal dalam teka-teki berbasis grid (Teneng et al., 2010; Bharlirus, 2019).
- Nonogram: Penyelesaian teka-teki logika berdasarkan petunjuk angka di baris dan kolom (Sidiq et al., 2023).
- Diagnosis Penyakit DBD: Meningkatkan efisiensi metode backward chaining dalam sistem pakar diagnosis (Kiswanto et al., 2024).
- Pencarian Data Skripsi: Menemukan kemiripan data skripsi untuk mempermudah pencarian (Juliantus & Sitorus, 2019).

Studi Literatur atau Pengumpulan Data, Studi literatur dilakukan dengan mengacu pada berbagai penelitian terdahulu yang menggunakan algoritma backtracking dan metode komputasi serupa. Data yang digunakan dalam penelitian meliputi:

1. Permainan Teka-teki:

Data berupa papan catur untuk N-Queen dan Knight Tour.

Grid 9x9 untuk permainan Sudoku (Hanafi et al., 2018).

Data grid dan pola petunjuk dalam Math Maze dan Nonogram (Teneng et al., 2010; Sidiq et al., 2023)

2. Data Sistem Pakar:

Aturan diagnosis penyakit demam berdarah (DBD) berdasarkan gejala dan fakta yang dikumpulkan dari pakar kesehatan (Kiswanto et al., 2024).

3. Data Pencarian Skripsi:

Kumpulan judul skripsi dari berbagai fakultas di Universitas Katolik Santo Thomas untuk pengujian algoritma Shift Or dan backtracking (Juliantus & Sitorus, 2019).

4. Permainan Minesweeper:

Data berupa struktur grid yang berisi bom dan kotak kosong dengan petunjuk angka (Tanoto, 2022).

5. Sumber Data: Literatur diambil dari jurnal ilmiah, buku algoritma, studi kasus, dan basis data penelitian.

3. Review Artikel Ilmiah

Review artikel ilmiah dilakukan dengan menganalisis implementasi algoritma backtracking dari berbagai penelitian untuk menilai efisiensi, akurasi, serta perbandingannya dengan metode lain seperti:

- Brute Force:

Brute force mengevaluasi semua kombinasi solusi tetapi memiliki kompleksitas waktu yang lebih tinggi

(Tanoto, 2022).

- Soft Computing:

Lebih cepat dalam penyelesaian teka-teki Nonogram tetapi kurang stabil dibanding backtracking (Sidiq et al., 2023).

- Backward Chaining:

Backtracking diintegrasikan dengan backward chaining untuk optimasi sistem pakar diagnosis penyakit (Kiswanto et al., 2024).

Hasil review menunjukkan bahwa algoritma backtracking unggul dalam akurasi solusi, meskipun memerlukan optimasi untuk meningkatkan efisiensi waktu pada skala besar.

Penyusunan Metode Penelitian

Metode penelitian yang disusun untuk setiap permasalahan dalam 10 jurnal melibatkan langkah-langkah berikut:

Formulasi Masalah

Menentukan ruang solusi dan kondisi batas dari masalah yang dihadapi (N-Queen, Sudoku, Minesweeper, dll.).

Implementasi Algoritma Backtracking

Menyusun struktur data untuk menyimpan solusi parsial.

Menggunakan pendekatan rekursif untuk mengeksplorasi semua kemungkinan solusi.

Melakukan validasi solusi dengan fungsi pembatas (Bounding Function).

Melakukan pruning untuk memangkas jalur yang tidak valid agar proses lebih efisien.

Pengujian dan Analisis Hasil

Kompleksitas Waktu: Mengukur waktu eksekusi algoritma backtracking dibandingkan dengan metode lain seperti brute force dan soft computing.

Validasi Solusi: Memastikan solusi yang dihasilkan memenuhi kriteria masalah.

Evaluasi Efisiensi: Mengukur perbedaan efisiensi backtracking pada kasus kecil, menengah, dan besar.

Studi Kasus

Permainan Teka-teki: Implementasi backtracking pada N-Queen, Sudoku, Minesweeper, Nonogram, Knight Tour, dan Math Maze.

Sistem Pakar: Optimalisasi backward chaining untuk diagnosis DBD.

Pencarian Data: Efisiensi algoritma backtracking dalam menemukan kemiripan data skripsi.

Kesimpulan

Metode penelitian ini menyusun langkah-langkah penerapan algoritma backtracking untuk menyelesaikan berbagai permasalahan seperti permainan logika, sistem pakar diagnosis, dan pencarian data. Dengan studi literatur, implementasi algoritma, dan pengujian terhadap data valid, hasil penelitian menunjukkan bahwa backtracking efektif dalam memberikan solusi yang akurat dan stabil, meskipun diperlukan optimasi lebih lanjut untuk permasalahan berskala besar.



Gambar 1. Contoh gambar

Berikut adalah contoh gambar yang menggambarkan algoritma Backtracking yang diterapkan pada N-Queen Problem. Gambar ini menunjukkan proses penempatan ratu pada papan catur, termasuk langkah validasi, eksplorasi jalur, dan langkah backtrack saat solusi tidak valid.

Langkah-langkah Penyelesaian N-Queen dengan Backtracking:

1. Inisialisasi Papan Catur

Papan catur berukuran 4x4 kosong.

Tujuannya adalah menempatkan 4 ratu di papan sehingga tidak ada dua ratu yang berada di baris, kolom, atau diagonal yang sama.

2. Mulai dari Baris Pertama (Langkah 1)

Letakkan ratu pertama (Q1) di kolom pertama pada baris pertama (misalnya, posisi (1,1)).

Papan catur sekarang memiliki satu ratu.

Algoritma kemudian melanjutkan ke baris kedua untuk menempatkan ratu kedua (Q2).

3. Penempatan Ratu Kedua (Langkah 2)

Di baris kedua, algoritma mencoba menempatkan Q2 di setiap kolom:

Cek kolom 1 → Tidak valid (ada ratu di (1,1)).

Cek kolom 2 → Valid (tidak ada ratu yang saling menyerang).

Letakkan Q2 di kolom kedua pada baris kedua (posisi (2,2)).

4. Penempatan Ratu Ketiga (Langkah 3)

Di baris ketiga, algoritma mencoba menempatkan Q3:

Cek kolom 1 → Tidak valid (di diagonal ratu Q2).

Cek kolom 2 → Tidak valid (sama kolom dengan Q2).

Cek kolom 3 → Valid (tidak ada konflik).

Letakkan Q3 di kolom ketiga pada baris ketiga (posisi (3,3)).

5. Penempatan Ratu Keempat (Langkah 4)

Di baris keempat, algoritma mencoba menempatkan Q4:

Cek kolom 1 → Tidak valid (di diagonal ratu Q3).

Cek kolom 2 → Tidak valid (di diagonal ratu Q2).

Cek kolom 3 → Tidak valid (sama kolom dengan Q3).

Cek kolom 4 → Valid (tidak ada konflik).

Letakkan Q4 di kolom keempat pada baris keempat (posisi (4,4)).

6. Solusi Ditemukan

Setelah Q4 ditempatkan, seluruh papan diisi 4 ratu dengan aturan valid (tidak ada konflik antar ratu).

Solusi yang ditemukan:

Q1 → (1,1)

Q2 → (2,2)

Q3 → (3,3)

Q4 → (4,4)

7. Backtracking (Jika Solusi Tidak Valid)

Jika pada langkah tertentu tidak ada posisi valid untuk menempatkan ratu, algoritma kembali (backtrack) ke langkah sebelumnya.

Misalnya:

Jika Q4 tidak bisa ditempatkan di baris keempat, algoritma akan kembali ke baris ketiga dan memindahkan Q3 ke posisi lain.

Proses ini berulang hingga solusi valid ditemukan.

HASIL PENELITIAN

Penelitian ini berfokus pada penerapan algoritma backtracking dalam berbagai bidang seperti permainan logika, teka-teki, sistem diagnosis penyakit, dan pencarian data. Hasil dari penerapan algoritma ini menunjukkan efektivitas dan efisiensi dalam menyelesaikan permasalahan dengan pendekatan eksplorasi solusi secara sistematis dan optimal.

1. Penyelesaian Masalah Logika dan Permainan

- N-Queen Problem:

Backtracking berhasil menempatkan N ratu pada papan catur berukuran 4x4 tanpa saling menyerang. Solusi yang dihasilkan adalah [2,4,1,3] dan [3,1,4,2] (Rahmawati et al., 2023).

Algoritma bekerja efektif untuk ukuran kecil tetapi membutuhkan waktu komputasi lebih lama untuk ukuran papan yang lebih besar.

- Knight Tour:

Backtracking mampu menemukan jalur gerakan kuda catur yang mengunjungi semua kotak pada papan berukuran 5x5 hingga 12x12 (Novianto & Pramadhana, 2019).

Efektivitas algoritma menurun seiring dengan bertambahnya ukuran papan.

- Sudoku:

Algoritma backtracking menyelesaikan teka-teki Sudoku dengan akurasi 100%.

Namun, waktu penyelesaian meningkat signifikan seiring dengan bertambahnya jumlah sel kosong dan tingkat kesulitan (Hanafi et al., 2018).

- Nonogram:

Backtracking berhasil menyelesaikan teka-teki Nonogram dengan stabilitas tinggi.

Dibandingkan metode soft computing, backtracking lebih akurat tetapi membutuhkan waktu komputasi lebih lama (Sidiq et al., 2023).

- Math Maze dan Labirin:

Algoritma backtracking berhasil menemukan jalur optimal untuk menyelesaikan permainan Math Maze dan teka-teki labirin (Teneng et al., 2010; Bharlirus, 2019).

Jalur eksplorasi ditentukan melalui langkah rekursif, dengan mundur ke langkah sebelumnya saat menemui jalan buntu.

- Minesweeper:

Backtracking meningkatkan efisiensi penyelesaian permainan Minesweeper dengan kompleksitas $O(2^n \cdot n)$.

Dibandingkan metode brute force, algoritma ini mampu memangkas ruang solusi dengan teknik pruning (Tanoto, 2022).

2. Optimasi Sistem Diagnosis Penyakit

Backtracking diintegrasikan dengan metode backward chaining dalam sistem pakar diagnosis penyakit Demam Berdarah Dengue (DBD).

Hasil penelitian menunjukkan peningkatan efisiensi sebesar 88%, dengan waktu eksekusi berkurang secara signifikan:

3 aturan: 168,50 ms → 19,53 ms.

6 aturan: 161,63 ms → 20,88 ms.

10 aturan: 161,68 ms → 17,38 ms.

Integrasi ini meminimalkan proses evaluasi aturan yang tidak relevan (Kiswanto et al., 2024).

3. Pencarian Data dan Optimasi

Backtracking diterapkan dalam proses pencarian data skripsi untuk menemukan kemiripan judul skripsi.

Hasil pengujian menunjukkan tingkat kemiripan bervariasi antara 10% hingga 100%, dengan pencarian yang lebih cepat dibandingkan metode manual (Juliantus & Sitorus, 2019).

Ringkasan Hasil Penelitian

1. Permainan dan Teka-Teki:

- Algoritma backtracking efektif menyelesaikan permainan seperti N-Queen Problem, Sudoku, Minesweeper, Math Maze, dan Knight Tour dengan akurasi tinggi.
- Dibandingkan metode lain (brute force atau soft computing), backtracking lebih stabil tetapi membutuhkan waktu komputasi lebih lama untuk skala besar.

2. Optimasi Diagnosis Penyakit:

- Integrasi backtracking dengan backward chaining berhasil meningkatkan efisiensi sistem diagnosis penyakit DBD sebesar 88%.
-

3. Pencarian Data:

Penerapan backtracking pada pencarian data skripsi memberikan hasil pencarian yang akurat dengan kecepatan lebih baik dibandingkan metode manual.

PEMBAHASAN

Pembahasan ini mengintegrasikan hasil penelitian dari 10 jurnal terkait penerapan algoritma backtracking dalam menyelesaikan berbagai permasalahan, seperti permainan logika, teka-teki kompleks, sistem diagnosis penyakit, dan optimasi pencarian data. Pembahasan ini difokuskan pada keunggulan, keterbatasan, dan perbandingan backtracking dengan metode lainnya.

1. Keunggulan Algoritma Backtracking

Algoritma backtracking menunjukkan efektivitas dalam menyelesaikan masalah dengan eksplorasi solusi secara rekursif dan sistematis. Keunggulan ini dapat dilihat dari:

a. Akurasi Solusi Tinggi

- Pada N-Queen Problem, backtracking mengeksplorasi semua kemungkinan penempatan ratu secara bertahap, menghasilkan solusi optimal untuk ukuran kecil (Rahmawati et al., 2023).
- Dalam Sudoku, algoritma backtracking mampu menyelesaikan teka-teki grid 9x9 dengan akurasi 100% (Hanafi et al., 2018).
- Untuk Nonogram, solusi yang dihasilkan oleh backtracking lebih stabil dibandingkan soft computing, meskipun waktu eksekusinya lebih lama (Sidiq et al., 2023).

b. Stabilitas Penyelesaian

- Backtracking memiliki keunggulan dalam stabilitas hasil karena mengevaluasi solusi secara menyeluruh. Pada permainan seperti Knight Tour, jalur optimal dapat ditemukan melalui eksplorasi Depth-First Search (Novianto & Pramadhana, 2019).
- Pada Minesweeper, backtracking menunjukkan efisiensi yang lebih baik dibandingkan brute force dengan teknik pruning untuk memangkas jalur yang tidak valid (Tanoto, 2022).

c. Penerapan dalam Sistem Pakar

- Pada sistem diagnosis penyakit, integrasi backtracking dengan metode backward chaining berhasil

meningkatkan efisiensi penelusuran aturan, mengurangi waktu eksekusi hingga 88% (Kiswanto et al., 2024).

d. Fleksibilitas Penggunaan

- Algoritma backtracking dapat diterapkan dalam berbagai jenis masalah, mulai dari permainan logika (N-Queen, Sudoku, Minesweeper) hingga masalah optimasi seperti pencarian data skripsi (Juliantus & Sitorus, 2019).

2. Keterbatasan Algoritma Backtracking

Meskipun memiliki keunggulan dalam akurasi, backtracking memiliki beberapa keterbatasan, terutama pada masalah dengan ruang solusi besar:

a. Kompleksitas Waktu

- Pada permainan N-Queen dan Knight Tour, kompleksitas waktu meningkat signifikan seiring dengan bertambahnya ukuran papan (Rahmawati et al., 2023; Novianto & Pramadhana, 2019).
- Penyelesaian Sudoku dengan jumlah sel kosong yang banyak membutuhkan waktu komputasi lebih lama (Hanafi et al., 2018).
- Nonogram dan Minesweeper memerlukan optimasi lebih lanjut agar efisiensi waktu bisa ditingkatkan (Sidiq et al., 2023; Tanoto, 2022).

b. Konsumsi Memori

- Penyimpanan solusi parsial dan langkah rekursif dalam backtracking dapat meningkatkan konsumsi memori, terutama dalam ruang solusi yang kompleks.
- Pada sistem diagnosis penyakit dengan banyak aturan, optimasi seperti memoization atau branch and bound diperlukan untuk mengurangi penggunaan memori (Kiswanto et al., 2024).

c. Skalabilitas

- Algoritma backtracking bekerja optimal pada masalah kecil hingga menengah. Pada ruang solusi yang besar, metode ini kurang efisien dibandingkan dengan pendekatan heuristik seperti soft computing atau greedy algorithms.

3. Perbandingan Backtracking dengan Metode Lain

Hasil dari 10 jurnal menunjukkan bahwa backtracking memiliki keunggulan dan kelemahan ketika dibandingkan dengan metode lainnya:

a. Backtracking vs Brute Force

- Backtracking lebih efisien dibandingkan brute force karena memiliki kemampuan pruning, sehingga jalur yang tidak valid dapat dipangkas lebih awal (Tanoto, 2022).
- Kompleksitas waktu brute force pada Minesweeper adalah $O(n.n!/(n-d)!)$, sementara backtracking dengan pruning memiliki kompleksitas $O(2^n.n)$.

b. Backtracking vs Soft Computing

- Pada Nonogram, soft computing memiliki kecepatan penyelesaian yang lebih baik tetapi hasilnya kurang stabil dibandingkan backtracking yang memberikan solusi lebih akurat (Sidiq et al., 2023).

c. Backtracking vs Backward Chaining

- Dalam sistem diagnosis penyakit, backtracking digunakan untuk mengoptimalkan metode backward chaining. Hasilnya, waktu eksekusi berkurang hingga 88% dengan akurasi tetap terjaga (Kiswanto et al., 2024).

KESIMPULAN

Berdasarkan hasil penelitian dari 10 jurnal, dapat disimpulkan bahwa algoritma backtracking merupakan metode yang efektif dan akurat dalam menyelesaikan berbagai permasalahan logika, kombinatorik, dan sistem berbasis aturan. Akurasi Tinggi: Backtracking mampu menghasilkan solusi yang benar dan valid untuk masalah seperti N-Queen Problem, Sudoku, Nonogram, Minesweeper, dan Knight Tour. Efisiensi Terbatas: Meskipun akurat, algoritma ini memiliki keterbatasan pada masalah berskala besar karena kompleksitas

waktu yang meningkat signifikan. Optimasi Diperlukan: Teknik seperti pruning, memoization, dan branch and bound dapat meningkatkan efisiensi algoritma. Penerapan Luas: Algoritma ini berhasil diterapkan tidak hanya dalam permainan logika, tetapi juga dalam sistem diagnosis penyakit dan pencarian data skripsi. Perbandingan Metode: Dibandingkan metode lain (brute force dan soft computing), backtracking lebih stabil dan akurat meskipun membutuhkan waktu komputasi lebih lama tanpa optimasi. Secara keseluruhan, algoritma backtracking adalah solusi sistematis untuk permasalahan yang memerlukan eksplorasi menyeluruh, dengan optimasi lebih lanjut diperlukan untuk masalah berskala besar.

REFERENCES

- Novianto, S., & Pramadhana, D. (2019). Penerapan Algoritma Backtracking dalam Menyelesaikan Permainan Knight Tour. *Jurnal Informatika*, 6(1), 45-52.
- Hanafi, A. A., Hibban, N., Zulfikar, F. M., & Adhinata, F. D. (2018). Implementasi Metode Backtracking untuk Penyelesaian Permainan Sudoku Berbasis Artificial Intelligence. *Jurnal Teknologi Informasi*, 7(2), 56-65.
- Sidiq, M. A. Z., Supriyadi, A., & Herliana, A. (2023). Perbandingan Keefektifan Algoritma Backtracking dan Soft Computing dalam Permainan Nonogram. *Jurnal Teknik Informatika dan Sistem Informasi*, 3(1), 9-19.
- Bharlirus, M. (2019). Implementasi Metode Backtracking untuk Checkpoint pada Game Mobile Misteri Labirin Prasasti Kerajaan Singosari 3D. *Jurnal Pengembangan Teknologi Informasi*, 5(1), 32-40.
- Teneng, J., Purwadi, J., & Kurniawan, E. (2010). Penerapan Algoritma Backtracking pada Permainan Math Maze. *Jurnal Rekayasa Sistem Komputer*, 4(3), 23-29.
- Kiswanto, R. H., Irjanto, N. S., Risman, M., & Imbiri, C. C. Y. (2024). Penerapan Algoritma Backtracking pada Metode Backward Chaining untuk Diagnosis Penyakit Demam Berdarah. *Jurnal Teknologi Sistem Informasi*, 2(1), 1-15.
- Tanoto, H. V. (2022). Analisis Penerapan Algoritma Backtracking untuk Menyelesaikan Permainan Minesweeper. *Makalah IF2211 Strategi Algoritma*, Institut Teknologi Bandung.
- Zafar, M. A., Supriyadi, A., & Herliana, A. (2023). Penyelesaian Permainan Nonogram dengan Algoritma Backtracking dan Soft Computing. *Jurnal Informatika dan Teknologi*, 3(1), 10-19.
- Juliantus, S., & Sitorus, L. (2019). Implementasi Backtracking dan Shift Or pada Proses Pencarian Data Skripsi. *Jurnal Sistem Informasi*, 4(2), 93-100.