JUSINFO
(Jurnal Sains dan Informatika)
Journal homepage: https://ejournal.pelitabina.com/index.php/Jusinfo
Pp: 1-7
Published By: Pelita Bina Neysha

Volume. 1 Nomor 1
Januari 2025

Penerapan Algoritma Backtracking Dalam Penyelesaian Masalah

Nurika Siregar[1], Hendra Jumianto[2]

[1] Teknik Informatika, STMIK Pelita Nusantara, Medan (email: nurikaagans94@gmail.com)
[2] Teknik Informatika, STMIK Pelita Nusantara, Medan (email: hendrajumianto2@gmail.com)

[Diserahkan: 23 Desember 2024, Direvisi: 30 Desember 2024, Diterima: 10 Januari 2025]
Corresponding Author: Nurika Siregar (email: nurikaagans94@gmail.com)

Kata Kunci: Backtracking, implementasi, labirin

Abstrak : Penelitian ini mengeksplorasi implementasi algoritma Backtracking untuk menemukan jalan keluar dari labirin yang memiliki banyak cabang dan jalan buntu. Algoritma Backtracking dirancang untuk efisiensi tinggi dengan menelusuri simpul tujuan dan memangkas langkah yang tidak perlu, berbeda dengan algoritma Depth First Search (DFS). Penelitian menggunakan matriks labirin berukuran η × η . Hasil penelitian menunjukkan bahwa algoritma ini berhasil menemukan jalan keluar dengan jumlah langkah yang minimal,karena Backtracking merunut balik dari simpul tujuan untuk melihat apakah solusi yang sedang dicari menuju pada simpul tujuan yang diinginkan. Dengan demikian algoritma Backtracking dapat memangkas langkah-langkah yang tidak perlu dalam sebuah pencarian dan dapat mencari rute terpendek dalam sebuah pencarian. Dari sisi tersebut algoritma Backtracking memiliki kelebihan dibandingan algoritma Depth First Search yang tidak mempertimbangkkan apakah solusi yang sedang dicari menuju pada titik tujuan yang diinginkan. Hasil yang didapat dalam penelitian ini berupa jumlah langkah dan simpul- simpul yang dilewati selama pencarian sampai menuju simpul tujuan.

PENDAHULUAN

Perkembangan teknologi informasi dan ilmu komputer telah menghasilkan berbagai pendekatan untuk menyelesaikan permasalahan yang kompleks. Perkembangan ilmu pengetahuan dan teknologi memungkinkan manusia menghasilkan karya yang semakin canggih dan kompleks. Meskipun komputer dapat melakukan perhitungan lebih cepat daripada manusia pada umumnya, komputer tidak dapat memecahkan masalah tanpa diajarkan oleh manusia melalui urutan langkah (algoritma) yang telah ditentukan sebelumnya. Selain digunakan untuk menyelesaikan masalah oleh komputer, algoritma juga dapat diterapkan untuk menyelesaikan masalah sehari-hari yang membutuhkan serangkaian proses atau langkah-langkah prosedural. Dalam banyak kasus, penyelesaian masalah melibatkan eksplorasi ruang solusi yang sangat besar, yang tidak memungkinkan untuk diselesaikan dengan metode brute-force secara efisien. Oleh karena itu, algoritma yang mampu mengeksplorasi solusi secara cerdas dan efisien menjadi kebutuhan utama dalam dunia komputasi. Salah satu pendekatan yang paling menonjol dalam menangani masalah-masalah tersebut adalah algoritma backtracking(Alya Aulia et al.,2020). Backtracking adalah metode algoritmik yang menggunakan pendekatan pencarian berbasis rekursi untuk membangun solusi secara bertahap. Metode ini bekerja dengan mencoba setiap kemungkinan solusi, dan jika solusi yang dihasilkan tidak memenuhi kriteria,

algoritma akan "mundur" atau kembali ke langkah sebelumnya untuk mencoba alternatif lain). Teknik ini memungkinkan eliminasi opsi yang tidak relevan secara dini, sehingga dapat mengurangi jumlah langkah yang diperlukan untuk menemukan solusi yang valid. Penerapan algoritma backtracking sangat luas dan mencakup berbagai jenis masalah, mulai dari masalah kombinatorik seperti teka-teki sudoku, pencarian jalur pada permainan catur, hingga masalah optimasi seperti knapsack problem atau penjadwalan tugas(Teneng et al., 2010). Keunggulan algoritma ini terletak pada kesederhanaan konsepnya dan fleksibilitasnya dalam menangani berbagai bentuk permasalahan. Namun, di sisi lain, algoritma ini juga memiliki kelemahan, terutama ketika ruang solusi sangat besar sehingga eksplorasi menjadi tidak efisien. Dalam situasi ini, diperlukan strategi tambahan seperti pruning dan heuristik untuk meningkatkan kinerja algoritma(Yosafat Adiguna et al., 2020).

Selain itu, dalam konteks aplikasi nyata, penerapan algoritma backtracking sering kali memerlukan penyesuaian agar sesuai dengan karakteristik masalah yang dihadapi. Misalnya, dalam masalah permainan seperti sudoku, algoritma ini dapat dilengkapi dengan teknik pencarian lebih dalam untuk mempercepat proses penyelesaian. Algoritma Backtracking merupakan salah satu algoritma yang penting dalam pemrograman karena mampu menyelesaikan permasalahan yang sulit dengan efisien. Dalam beberapa kasus, algoritma ini dapat menjadi alternatif yang lebih baik dibandingkan dengan pendekatan brute force. Dalam masalah penjadwalan, algoritma backtracking dapat dikombinasikan dengan teknik optimasi untuk menghasilkan solusi yang lebih baik dalam waktu yang lebih singkat. Oleh karena itu, pemahaman mendalam tentang prinsip kerja algoritma backtracking, serta eksplorasi cara-cara untuk meningkatkan efisiensinya, menjadi hal yang sangat penting((Abdul Kadir et al., 2016). Penelitian ini bertujuan untuk mengeksplorasi penerapan algoritma backtracking dalam penyelesaian masalah tertentu, baik dari segi teori maupun aplikasi. Fokus utama adalah menganalisis efektivitas dan efisiensi algoritma ini dalam menemukan solusi optimal, serta mengevaluasi strategi yang dapat diterapkan untuk meningkatkan performanya. Dengan pendekatan ini, diharapkan penelitian ini tidak hanya memberikan pemahaman yang lebih baik tentang algoritma backtracking, tetapi juga menawarkan solusi yang dapat diterapkan dalam berbagai bidang yang membutuhkan penyelesaian masalah secara efektif(Yulyanto et al ., 2023).

Disamping itu juga, penelitian-penelitian pada komputer terus dilakukan, terkait pemanfaatan perangkat penyusun data yang bernama algoritma. Penelitian-penelitian algoritma dilakukan sejak tahun 1950 hingga sampai sekarang ini (Ekowati et al., 2022). Dari berbagai pembagian macam algoritma tersebut, tentu memiliki perbedaan antara satu dan yang lainnya. Perbedaan yang pada algoritma pengurut, dapat dilihat dari hal-hal yang kompleks yang ada pada masing-masing algoritma, yaitu dari sisi kelebihan dan kekurangannya. Perbedaan algoritma tersebut, dapat dibagi pada tingkat: Kecepatan waktu pengelolahan data, penggunaan jumlah ruang memori yang terpakai, Keefesienan dalam cara penggunaannya dan lain sebagainnya(Zutshi & Goswami, 2021). Sehingga, melalui penelitian yang dilakukan dengan memperhatikan kelebihan serta keefienannya algoritma *Backtracking* dan *Depth First Search* . sehingga dapat mempermudah dalam rasa kebingungan pada para penggunanya, terutama para para pemula pemograman. Penggunaanya dapat memilih algoritma *Backtracking* dan *Depth First Search*, yang disesuai pada suatu permasalahan yang akan diselesaikan menggunakan alat pengurut yang ada pada algoritma pemograman.

TINJAUAN LITERATUR

Penelitian (Nurhasanah et al., 2023) dengan judul penelitian "Penerapan Algorima Backtracking Dengan Bounding Function Dan Depth First Search Pada Permainan Boggle". Algoritma backtracking adalah metode penyelesaian masalah yang digunakan untuk menemukan semua solusi yang mungkin dengan menggabungkan pencarian berulang dan penyelesaian solusi yang tidak valid.

Penelitian (Valdo et al., 2013) dengan judul penelitian "Implementasi Algoritma Backtracking Dengan Optimasi Menggunakan Teknik Hidden Single Pada Penyelesaian Permainan Sudoku". Algoritma Backtracking adalah algoritma perbaikan dari algoritma brute-force dan algoritma exhaustive-search,dimana algoritma ini membangun phpn ruang status *(state-space tree)* untuk menemukan solusi. Prinsip dari algoritma backtracking adalah jika terjadi kesalahan dalam pencarian solusi pada sebuah node, maka akan dilakukan backtrack ke node sebelumnya.

Penelitian (Muhammad Husni Rifqoet, 2013) dengan judul penelitian "Implementasi Algoritma Backtracking Dalam Sistem Informasi Perpustakaan Untuk Pencarian Judul Buku". Algoritma backtracking merupakan perbaikan dari algoritma brute-force, secara sistematis mencari Solusi persoalan diantara semua kemungkinan solusi yang ada. Dasar teknik backtracking ini adalah suatu Teknik pencarian (searching). Algoritma ini berbasis pada algoritma Depth-First Search (DFS), cara kerjanya menelusuri salah satu kemungkinan yang ada hingga mendapatkan hasil maksimal.

Penelitian (Santi Rahmawati et al., 2023) dengan judul penelitian "Penerapan Algoritma Backtracking Pada N-Queen Problem Permainan Catur". Penelitian ini menggunakan pendekatan metode kepustakaan (*library research*), yaitu dengan mengumpulkan data dari berbagai referensi, seperti jurnal, skripsi, dan dokumen lain yang relevan dengan topik penelitian. Selain itu, peneliti memanfaatkan strategi *Depth-First Search* (DFS) pada algoritma backtracking untuk proses pencarian solusi, serta menerapkan konsep pohon solusi (*State Space Tree*) sebagai representasi langkah-langkah yang dapat diambil dalam upaya menemukan solusi.

Penelitian (Alya et al., 2022) dengan judul penelitian "Penyelesaian Permainan Sudoku Menggunakan Algoritma Backtracking Berbasis Artificial Intelligence". Algoritma Backtracking adalah salah satu algoritma pencarian yang efisien, karena algoritma ini melakukan langkah mundur dari simpul tujuan untuk memeriksa apakah solusi yang sedang dicari mengarah pada simpul tujuan yang diinginkan.

Penelitian (|Teneng et al., 2010) dengan judul penelitian "Penerapan Algorima Backtracking Pada Permainan Math Maze". Algoritma backtracking (runut balik) adalah salah satu metode penyelesaian masalah yang menggunakan pendekatan berbasis pencarian dalam ruang status. Algoritma ini bekerja secara rekursif dengan menelusuri semua kemungkinan solusi secara sistematis. Karena algoritma ini didasarkan pada pendekatan *Depth-First Search* (DFS), pencarian solusi dilakukan dengan menelusuri struktur berupa pohon berakar menggunakan urutan *preorder*.

Penelitian (Yosafat Adiguna et al., 2020) dengan judul penelitian "Implementasi Algoritma Backtracking untuk Mencari Jalan Keluar Labirin". Labirin merupakan teka-teki yang sering digunakan dalam permainan, dengan banyak cabang dan jalan buntu yang membuat pencarian jalan keluar menjadi sulit. Algoritma pencarian dapat membantu dalam menemukan jalan keluar dari labirin, tetapi tidak semua algoritma pencarian cocok diterapkan pada labirin. Algoritma Backtracking adalah salah satu algoritma pencarian yang efisien, karena algoritma ini melakukan langkah mundur dari simpul tujuan untuk memeriksa apakah solusi yang sedang dicari mengarah pada simpul tujuan yang diinginkan.

METODE PENELITIAN

Penelitian ini menggunakan metode library research yaitu suatu penelitian yang menggunakan literatur kepustakaan. Data-data tersebut dianalisis dan diambil dengan sumber rujukan seperti buku, artikel, jurnal dan lainnya. Pada penelitian kali ini, sumber yang menjadi acuan adalah jurnal. Data-data yang dikumpulkan tersebut kemudian dianalisis satu per satu hingga ditemukan jawaban terkait persoalan yang sedang di teliti.Cara penyusunannya disusun dengan memperhatikan teori yang ada dan saling berhubungan. Penulisan dalam pengkajian gagasan yang menjadi rujukan untuk menghasilkan suatu informasi terkait pembahasan perbandingan Algoritma *Backtracking* dan *Depth First Search* pada tingkat keefisienan dalam penggunaanya. Sehingga dari informasi yang dihasilkan, dapat diketahui tingkat kelebihan dan kekurangan antara keduanya.

- **Depth-First Search (DFS)**

Depth first search (DFS) pertama kali diperkenalkan oleh Tarjan dan Hopcroft pada abad ke-19. Mereka menunjukkan cara DFS dapat digunakan untuk membuat banyak algoritma grafik yang efektif. Metode pencarian mendalam adalah metode Depth First Search (DFS). Metodenya adalah dengan menelusuri salah satu kemungkinan yang tersedia hingga akar-akar maksimal, dan kemudian menelusuri kemungkinan lain hingga akar-akar maksimal juga.

Perlu diperhatikan bahwa kebutuhan waktu proses pencarian dengan DFS sejalan dengan kedalaman pohon pencarian yang paling tinggi. Algoritma mungkin tidak akan berhenti jika kedalaman pohon tidak terbatas. Ini dapat terjadi dalam kasus ruang pencarian tidak terbatas atau jika ruang pencarian mengandung siklus keadaan. Oleh karena itu, algoritma DFS hanya menunjukkan beberapa ruang pencarian dan tidak menunjukkan sifat komplitnya.

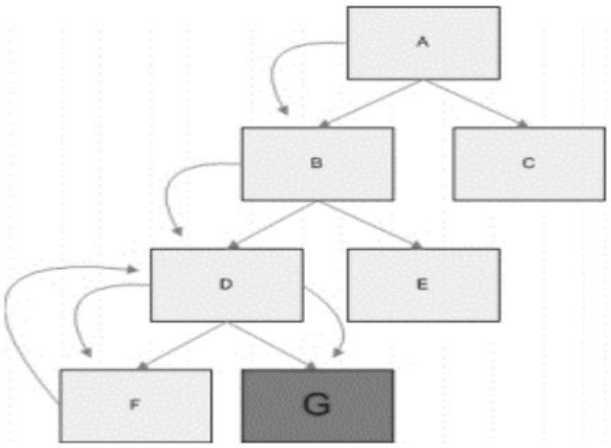
HASIL PENELITIAN

Algoritma Backtracking adalah metode algoritmik yang menggunakan pendekatan pencarian berbasis rekursi untuk membangun solusi secara bertahap. Metode ini bekerja dengan mencoba setiap kemungkinan solusi, dan jika solusi yang dihasilkan tidak memenuhi kriteria, algoritma akan "mundur" atau kembali ke langkah sebelumnya untuk mencoba alternatif lain). Teknik ini memungkinkan eliminasi opsi yang tidak relevan secara dini, sehingga dapat mengurangi jumlah langkah yang diperlukan untuk menemukan solusi yang valid. salah satu teknik pemecahan masalah yang menggunakan metode mencoba-coba (trial and error) untuk mencari solusi yang memenuhi semua syarat atau kendala masalah(Santi Rahmawati et al., 2023).

Cara Kerja Backtracking yakni

- Mulai dari langkah awal: Pilih langkah pertama dalam ruang solusi.
- Lalu Evaluasi: Periksa apakah langkah tersebut memenuhi syarat;
  Jika ya, lanjutkan ke langkah berikutnya.
  Jika tidak, kembali (backtrack) ke langkah sebelumnya dan coba alternatif lain.
- Lalu Ulangi: Proses ini diulangi sampai solusi ditemukan atau semua kemungkinan dicoba.

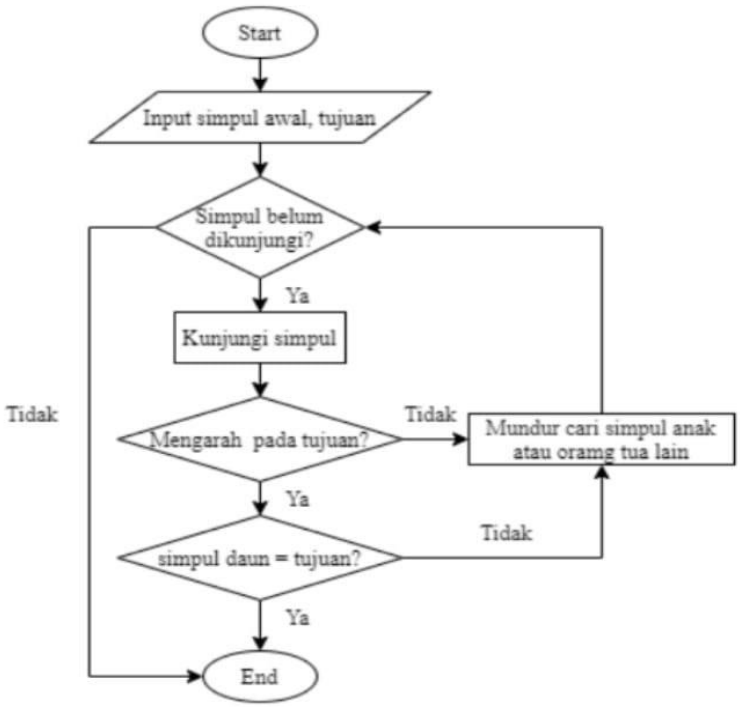
Backtracking adalah cara sistematis untuk mengulang seluruh konfigurasi yang mungkin terjadi dari sebuah ruang pencarian. Konfigurasi ini dapat berupa semua susunan permutasi objek yang dapat terjadi atau seluruh cara yang mungkin untuk membuat kumpulan dari objek-objek tersebut.Backtracking adalah algoritma yang berbasis pada Depth First Search (DFS) untuk mencari solusi persoalan lebih cepat. Runut balik yang merupakan perbaikan dari algoritma brute-force, secara sistematis mencari solusi persoalan di antara semua.



Gambar 1 adalah gambaran cara kerja algoritma Backtracking. Pada Gambar 1 simpul akar adalah A dan simpul tujuan adalah G. Pencarian dimulai dari memeriksa simpul anak dari simpul A yang adalah simpul B dan C. Lalu simpul B dipilih karena menuju pada simpul tujuan. Lalu diperiksa simpul anak dari simpul B yaitu D dan E. Simpul D lalu dipilih Karena mengarah pada simpul tujuan. Lalu diperiksa simpul anak dari D yaitu F dan G. lalu simpul F secara default, karena simpul F adalah simpul daun maka diperiksa apakah simpul F merupakan tujuan dan karena F bukan tujuan maka pencarian dilakukan dengan backtrack ke simpul D lalu memilih simpul anak lain dari simpul D. Simpul G lalu dipilih dan diperiksa apakah simpul G merupakan tujuan, karena simpul G merupakan simpul tujuan pencarian selesai.

Gambar 2 adalah gambar flowchart algoritma Backtracking untuk mencari jalan keluar sebuah labirin. Langkah pertama yang dilakukan dalam algoritma Backtracking seperti yang terlihat dalam gambar 2 adalah

memasukkan simpul awal dan simpul tujuan. Lalu dilakukan pengecekan apakah ada simpul yang belum dikunjungi dari simpul aktif jika tidak maka pencarian dihentikan. Jika ada yang belum dikunjungi maka simpul dikunjungi lalu dilakukan pengecekan apakah jalan yang ditempuh mengarah pada simpul tujuan, jika tidak maka dilakukan backtrack ke simpul anak atau orang tua lain lalu pencarian dilanjutkan dari langkah pengecekan simpul yang belum dikunjungi. Jika simpul menuju pada simpul tujuan lalu dilakukan proses perulangan dengan memilih simpul yang sebagai simpul aktif. Jika mencapai simpul daun maka dilakukan pengecekan apakah simpul daun yang didapat adalah simpul tujuan dan jika tidak maka dilakukan backtrack. Jika simpul daun adalah tujuan maka pencarian selesai.

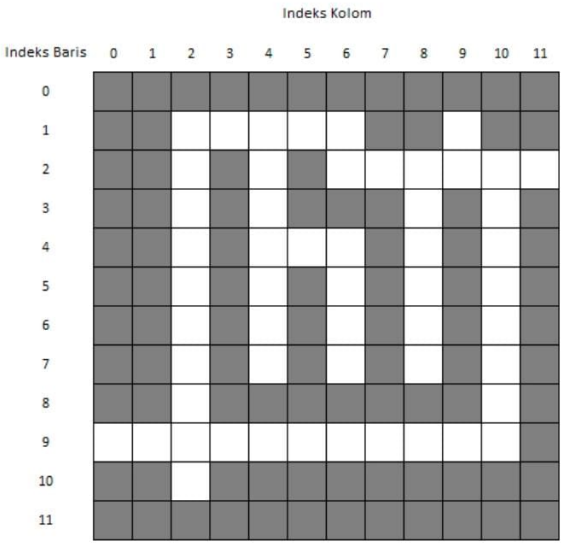


Gambar 2. Flowchart Algoritma *Backtracking*

Indeks baris	Indeks kolom 0	1	2	3	4	5	6	7	8	9	10	11
0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	1	1	1	1	1	0	0	1	0	0
2	0	0	1	0	1	0	1	1	1	1	1	1
3	0	0	1	0	1	0	0	0	1	0	1	0
4	0	0	1	0	1	1	1	0	1	0	1	0
5	0	0	1	0	1	0	1	0	1	0	1	0
6	0	0	1	0	1	0	1	0	1	0	1	0
7	0	0	1	1	1	0	1	0	1	0	1	0
8	0	0	1	0	0	0	0	0	0	0	1	0
9	1	1	1	1	1	1	1	1	1	1	1	0
10	0	0	1	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0

Gambar 3. Contoh Matriks Labirin

Contoh matriks labirin yang digunakan dalam penelitian ini adalah matriks seperti pada gambar 3. Gambar 3 merupakan matriks labirin yang akan digunakan untuk penelitian ini, matriks ini berisi angka 1 dan 0. Angka 1 mewakili jalan yang dapat dilalui sedangkan angka 0 mewakili tembok yang tidak dapat dilalui. Matriks memiliki ukuran 12 x 12, dengan 12 baris yang memiliki indeks 0 sampai 11 dimulai dari kiri atas dan 12 kolom dengan jumlah indeks yang sama.



Gambar 4. Visualisasi Contoh Matriks Labirin

PEMBAHASAN

Berdasarkan hasil penelitian, dapat disimpulkan bahwa penggunaan algortima *Backtracking* lebih optimal dan lebih efisien daripada penggunanaan algoritma *Depth First Search* pada pencarian jalan keluar. Hal ini dapat diperkuat berdasarkan Analisa jurnal yang diambil sebagai acuan dalam penelitian ini.

1. Jurnal 1 : Menjelaskan penelitian yang menganalisis dan membandingkan penggunaan algoritma Backtracking dengan Bounding Function dan DFS dalam penyelesaian permaianan Boggle. Pada penelitian ini algoritma *backtracking* merupakan algoritma yang memungkinkan pencarian efisien melalui ruang solusi untuk menemukan solusi yang falid (Nurhasanah, Uden Saptoni ,2023).

2. Jurnal 2 : Menjelaskan suatu penelitian untuk melakukan implementasi teknik *hidden single* pada algoritma *backtracking* dengan tujuan melakukan optimasi pada algoritma. Hasil uji coba penelitian menunjukkan bahwa dengan menggunakan optimasi teknik hidden single, algoritma algoritma *backtracking* mampu mengoptimalkan waktu dan kinerja komputasi pada penyelesaian puzzle Sudoku(Valdo et al ., 2013).

3. Jurnal 3 : Menjelaskan tentang Penelitian dengan permasalahan tentang belum efisiennya informasi yang dihasilkan dari system informasi pencarian judul buku yang ada pada UPT perpustakaan UMB. algoritma *backtracking* bisa dengan cepat,tepat dan akurat (Muhammad Husni Rifqoet et al. 2017).

4. Jurnal 4 : Menjelaskan suatu penelitian tentang menemukan konfigurasi penempatan ratu yang memenuhi semua batasan yakni tidak ada dua ratu yang berada dalam baris yang sama,di kolom yang sama, dan di diagonal yang sama. . Dengan membentuk pohon ruang status (State Space Tree) pada Algoritma Backtracking menghasilkan solusi pada N = 4 dengan jalur [2,4,1,3](Santi Rahmawati et al ., 2023).

5. Jurnal 5 : Menjelaskan suatu penelitian tentang permainan Sudoku yang termasuk pada permasalahan NP-Complete dengan menggunakan algoritma *Backtracking*. Dimana,ditemukan tingkat keefesienan yang lebih cepat, dan tidak memakan waktu yang lama(Alya et al ., 2022).

6. Jurnal 6 : Menjelaskan suatu analisis terkait Penerapan Algoritma Backtracking Pada Permainan Math Maze.Hasil percobaan yang dilakukan dengan membandingkan posisi Start dan Goal yang berbeda-beda .Hal ini menandakan bahwa algoritma *Backtracking* merupakan algoritma pengurut yang cara pengerjaannya

memakan waktu yang cukup tinggi, dibandingkan dengan algoritma *Depth First Search* (Teneng et al. 2010).

7. Jurnal 7: Menjelaskan suatu penelitian Menjelaskan suatu penelitian tentang pencarian Solusi pada permainan *puzzle* pada perangkat lunak menjukan bahwa *Backtracking* memiliki cara kerja yang cepat dibandingkan algoritma lainnya (Abdul Kadir Hasani et al .,2016).

8. Jurnal 8 : Menjelaskan suatu penelitian yang dilakukan dalam pencarian strategi pengecohan lawan dalam permainan catur .Dikemukan bahwa Tingkat kecepatan dan keefesienan penggunaan waktu yang lebih baik yaitu pada algoritma *Backtracking*(Sendi et al ., 2019).

9. Jurnal 9 : Menjelaskan suatu penelitian terkait mencari rute terpendek jalan keluar sebuah labirin.

Ditemukan bahwa Algoritma Backtracking memiliki keunggulan dibandingkan dengan algoritma pencarian DFS yang menelusuri semua kemungkinan yang dapat terjadi sehingga membuat algoritma tidak efisien dan memerlukan banyak sumber daya(Yosafat Adiguna et al ., 2020).

10. Jurnal 10 : Menjelaskan pengujian algoritma *Backtracking* menemukan jalur terpendek untuk dilalui oleh kelinci untuk mencari rumput dan mencapai tujuan dalam permainan puzzle find grass,ditemukan bahwa *Backtracking* memiliki pengerjaan yang cukup baik dalam berbagai bahasa pemograman(Yulyanto et al., 2023).

KESIMPULAN

Kesimpulan dari penelitian ini menekankan pentingnya pemilihan algoritma pencarian yang sesuai, Algoritma Backtracking dapat digunakan mencari rute terpendek jalan keluar sebuah masalah, namun hal ini sangat bergantung kepada isi dari permasalahan yang akan dicari jalan keluarnya. Algoritma Backtracking memiliki efisiensi yang tinggi karena mempertimbangkan rute yang sedang dicari apakah akan menuju simpul yang diinginkan atau tidak sehingga pencarian tidak perlu dilakukan dengan mencari semua kemungkinan yang dapat terjadi, dengan demikian mempercepat proses pencarian dan mengurangi simpul-simpul yang harus dicari. Algoritma Backtracking memiliki keunggulan dibandingkan dengan algoritma pencarian *Depth First Search* (DFS) yang menelusuri semua kemungkinan yang dapat terjadi sehingga membuat algoritma tidak efisien dan memerlukan banyak sumber daya. Algoritma Backtracking lebih sederhana dibandinganken dengan algoritma pencari rute terpendek seperti A*. Algoritma A* dapat mencari rute terpendek dengan menggunakan open dan closed list.

REFERENCES

Febryanto, Aidil. 2022. "Penerapan Algoritma Sequential Search Untuk Mencari Data Siswa Pada Sekolah Menengah Kejuruan Negeri 3 Bengkalis." *Penerapan Algoritma Sequential Search Untuk Mencari Data Siswa Pada Sekolah Menengah Kejuruan Negeri 3 Bengkalis* 2(1):51–59.

Firmansyah[1]), Andri. 2019. "SIGMA - Jurnal Teknologi Pelita Bangsa SIGMA - Jurnal Teknologi Pelita Bangsa." *SIGMA - Jurnal Teknologi Pelita Bangsa 167* 10(September):167–72.

Onsardi, Onsardi, Muntahanah Muntahanah, and Rozali Toyib. 2020. "Penerpan Algoritma Binary Search Dalam Pencarian Data Potensi Investasi Di Kabupaten Seluma Dengan Smartphone." *JSAI (Journal Scientific and Applied Informatics)* 3(3):129–36. doi: 10.36085/jsai.v3i3.1160.

Rizaldi, Rizaldi. 2020. "Komparasi Algoritma Sequential Searching Dan Interpolation Searching Pada Studi Kasus Pencarian Data Tilang Pengadilan Negeri Samarinda." *Jurnal Rekayasa Teknologi Informasi (JURTI)* 4(1):86. doi: 10.30872/jurti.v4i1.5104.

Sonita, Anisya, and Mayang Sari. 2018. "Implementasi Algoritma Sequential Searching Untuk Pencarian Nomor Surat Pada Sistem Arsip Elektronik." *Pseudocode* 5(1):1–9. doi: 10.33369/pseudocode.5.1.1-9.

Wahyuni, Wafiqah Setyawati, Septi Andryana, and Ben Rahman. 2022. "PENGGUNAAN ALGORITMA SEQUENTIAL SEARCHING PADA APLIKASI PERPUSTAKAAN BERBASIS WEB." *JIPI (Jurnal Ilmiah Penelitian Dan Pembelajaran Informatika)* 7(2):294–302. doi: 10.29100/jipi.v7i2.2646.