**Aplikasi Pencarian Jalur Aman Evakuasi Gedung Berbasis Backtracking**

**Nur Adnan Yusri1, Fatahillah Furqan As-sidiq2**

*1Department of Informatics, Universitas Muhammadiyah Makassar, Indonesia*

*21Department of Informatics, 1Universitas Muhammadiyah Makassar, Indonesia*

*1Universitas Muhammadiyah Makassar*

*\*email: 105841100123@student.unismuh.ac.id*

**Abstract**

*Emergency evacuation in multi-story buildings involves high risk due to spatial complexity, dynamic obstacles, and hazardous zones that may change during disasters. Most existing evacuation systems rely on conventional pathfinding algorithms that generate a single optimal route, which limits flexibility when the main path becomes blocked or unsafe. This study proposes a grid-based building evacuation system using a backtracking algorithm to generate multiple safe evacuation routes (*multi-solution evacuation*). The building environment is modeled as a two-dimensional grid representing free space, obstacles, and hazard zones with associated risk penalties. The backtracking algorithm explores all feasible paths recursively while applying pruning rules based on obstacles, step limits, and risk thresholds to ensure efficiency. The proposed system is implemented in a web-based application named* ***EVACUATE-X****, integrating a frontend visualization module and a backend pathfinding engine. Experimental evaluation is conducted on a 10 × 10 grid environment with static obstacles and hazard zones. The results show that the proposed approach successfully generates multiple valid evacuation routes, from which the Top-10 safest paths are selected based on a combined score of path length and risk. The algorithm achieves fast computation time and provides flexible evacuation alternatives, making it suitable for interactive evacuation decision support systems.*

*Keywords: evacuation route, backtracking algorithm, grid-based modeling, multi-solution pathfinding, emergency evacuation*

**Abstrak**

Evakuasi darurat pada bangunan bertingkat memiliki tingkat risiko yang tinggi akibat kompleksitas ruang, keberadaan hambatan, serta zona berbahaya yang dapat berubah secara dinamis. Sebagian besar sistem evakuasi yang ada masih mengandalkan algoritma pencarian jalur konvensional yang hanya menghasilkan satu jalur optimal, sehingga kurang fleksibel ketika jalur utama tidak dapat digunakan. Penelitian ini mengusulkan sistem pencarian jalur evakuasi gedung berbasis grid dua dimensi menggunakan algoritma backtracking untuk menghasilkan banyak alternatif jalur evakuasi aman (multi-solution evacuation). Lingkungan bangunan dimodelkan dalam bentuk grid yang merepresentasikan area aman, obstacle, dan zona bahaya dengan penalti risiko tertentu. Algoritma backtracking menelusuri seluruh kemungkinan jalur secara rekursif dengan menerapkan aturan pruning berdasarkan hambatan, batas langkah, dan ambang risiko. Sistem diimplementasikan dalam aplikasi berbasis web bernama **EVACUATE-X** yang mengintegrasikan modul visualisasi frontend dan mesin pencarian jalur backend. Evaluasi eksperimen dilakukan pada lingkungan grid berukuran 10 × 10 dengan konfigurasi obstacle dan zona bahaya statis. Hasil penelitian menunjukkan bahwa algoritma mampu menghasilkan banyak jalur evakuasi valid dan menyeleksi Top-10 jalur terbaik berdasarkan kombinasi panjang jalur dan tingkat risiko. Pendekatan ini terbukti efisien, responsif, serta meningkatkan fleksibilitas dan keandalan pengambilan keputusan evakuasi.

Kata kunci: *jalur evakuasi, algoritma backtracking, grid map, multi-solusi, evakuasi darurat*

1. **Pendahuluan**
   1. **Latar belakang**

Bangunan bertingkat memiliki risiko tinggi dalam situasi darurat seperti kebakaran dan gempa bumi karena keterbatasan ruang evakuasi, kepadatan penghuni, serta kompleksitas struktur bangunan. Kondisi tersebut dapat meningkatkan waktu evakuasi dan membahayakan keselamatan apabila jalur evakuasi tidak dirancang secara optimal dan adaptif [1], [2]. Berbagai pendekatan komputasi telah diterapkan untuk mendukung perencanaan jalur evakuasi, mulai dari algoritma berbasis graf hingga metode cerdas seperti *reinforcement learning*. Meskipun demikian, pendekatan cerdas tersebut umumnya membutuhkan komputasi tinggi dan tidak selalu efisien untuk diterapkan pada sistem evakuasi sederhana atau berskala kecil, khususnya pada kondisi darurat yang membutuhkan respons cepat [3], [4]. Sebagian besar sistem evakuasi yang digunakan saat ini masih bersifat statis, yaitu mengandalkan rambu atau jalur tetap tanpa mempertimbangkan perubahan kondisi lingkungan secara dinamis. Pendekatan ini memiliki kelemahan karena tidak mampu menyesuaikan jalur ketika terjadi hambatan seperti jalur tertutup, zona berbahaya, atau kepadatan manusia, sehingga berpotensi mengarahkan penghuni ke rute yang tidak aman [2], [5]. Selain itu, algoritma pencarian jalur yang umum digunakan, seperti *Dijkstra*, *Breadth-First Search* (BFS), dan *Depth-First Search* (DFS),umumnya hanya menghasilkan satu jalur evakuasi terbaik (*single-path solution*). Keterbatasan jalur tunggal ini mengurangi fleksibilitas evakuasi ketika jalur utama tidak dapat digunakan. Penelitian komparatif menunjukkan bahwa meskipun algoritma optimasi seperti *Ant Colony Optimization* (ACO) mampu meningkatkan efisiensi pencarian jalur, sistem yang dihasilkan tetap berfokus pada satu solusi jalur [6]. Oleh karena itu, diperlukan pendekatan pencarian jalur evakuasi yang mampu menghasilkan beberapa alternatif jalur aman (*multi-solution evacuation routes*). Algoritma *backtracking* memiliki keunggulan dalam menelusuri seluruh kemungkinan jalur secara sistematis dan menghasilkan lebih dari satu solusi valid. Namun, pemanfaatan *backtracking* sebagai metode utama dalam sistem evakuasi bangunan bertingkat masih terbatas dan belum banyak dibahas secara mendalam dalam penelitian sebelumnya [7]

* 1. **Research Objectives, and Contributions**

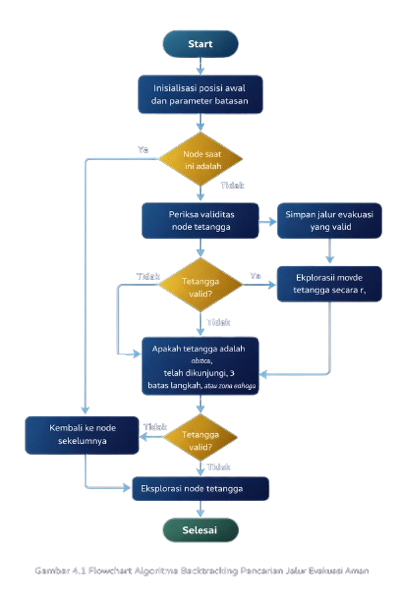
Sebagian besar sistem evakuasi gedung masih menggunakan algoritma pencarian jalur konvensional seperti Dijkstra, Breadth-First Search (BFS), dan Depth-First Search (DFS) yang umumnya hanya menghasilkan satu jalur evakuasi optimal. Pendekatan jalur tunggal ini kurang fleksibel dalam kondisi darurat karena tidak menyediakan rute alternatif ketika jalur utama terhalang oleh hambatan atau zona berbahaya, terutama pada bangunan bertingkat dengan kompleksitas ruang tinggi [1],[6]. Meskipun metode berbasis machine learning dan reinforcement learning telah diusulkan untuk meningkatkan adaptivitas jalur evakuasi, pendekatan tersebut memiliki kompleksitas komputasi tinggi dan membutuhkan sumber daya besar, sehingga kurang sesuai untuk sistem evakuasi yang menuntut respons cepat dan implementasi ringan [8], [4]. Penelitian ini mengusulkan sistem pencarian jalur evakuasi berbasis grid dua dimensi yang menerapkan algoritma backtracking untuk menghasilkan beberapa alternatif jalur evakuasi aman (multi-solution). Lingkungan gedung dimodelkan sebagai grid yang merepresentasikan ruang kosong, hambatan, dan zona bahaya, kemudian dieksplorasi secara menyeluruh dengan mempertimbangkan kendala keselamatan dan efisiensi [1]. Kinerja sistem dievaluasi berdasarkan jumlah jalur aman, panjang jalur, dan waktu komputasi, serta dibandingkan secara kualitatif dengan algoritma konvensional seperti BFS, DFS, dan Dijkstra [9], [10]. Kontribusi utama penelitian ini adalah penyediaan solusi evakuasi multi-jalur yang fleksibel, ringan, dan mudah diimplementasikan, dilengkapi visualisasi berbasis web untuk mendukung pengambilan keputusan evakuasi yang lebih andal [6], [5].

1. **Metode Penelitian**
   1. **Algorithm Overview**

Penelitian ini mengusulkan algoritma backtracking sebagai metode utama pencarian jalur evakuasi aman pada lingkungan gedung berbasis grid graph. Algoritma bekerja secara rekursif dengan menelusuri setiap kemungkinan jalur dari start node menuju exit node serta melakukan backtrack ketika jalur melanggar batasan lingkungan, seperti obstacle, batas langkah, atau ambang risiko. Pendekatan ini menjamin kelengkapan solusi dan sesuai untuk permasalahan pencarian jalur pada ruang diskret [2], [8]. Berbeda dengan algoritma pencarian jalur konvensional yang hanya menghasilkan satu solusi, algoritma backtracking dalam penelitian ini dirancang untuk melakukan multi-solution search. Setiap jalur yang mencapai titik keluar dan memenuhi kriteria keamanan disimpan sebagai solusi yang valid, sehingga sistem mampu menyediakan beberapa alternatif jalur evakuasi pada kondisi darurat yang dinamis [6], [5]. Proses pencarian dilakukan dengan mengeksplorasi node tetangga pada empat arah utama (four-connected grid) dan memvalidasi setiap langkah sebelum dilanjutkan secara rekursif. Jalur yang tidak valid akan dihentikan lebih awal melalui mekanisme pruning dan algoritma kembali ke node sebelumnya. Alur kerja algoritma secara ringkas ditunjukkan pada **Gambar 1.**

* 1. **Pseudocode of the Algorithm**

Algoritma backtracking yang diusulkan dirancang untuk menelusuri seluruh kemungkinan jalur evakuasi dari titik awal menuju titik keluar pada lingkungan

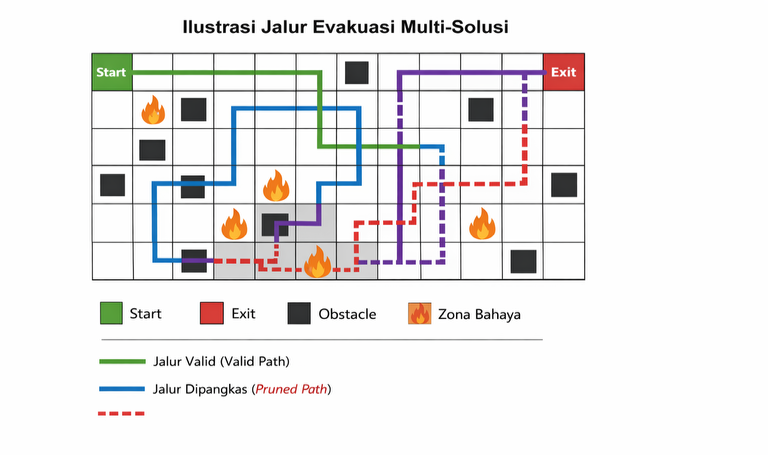


1. Flowchart Algoritma Backtracking

Lingkungan gedung dimodelkan sebagai grid dua dimensi, dengan pencarian jalur evakuasi dilakukan secara rekursif menggunakan algoritma backtracking yang membangun lintasan langkah demi langkah hingga mencapai titik keluar. Proses eksplorasi dihentikan ketika jalur melanggar batasan validitas, efisiensi, atau keselamatan, sehingga seluruh jalur evakuasi yang layak dapat ditemukan tanpa kehilangan solusi potensial. Pendekatan ini sesuai dengan prinsip perencanaan evakuasi multi-rute yang menekankan fleksibilitas dan kelengkapan solusi [11], [12]. Algoritma mengadopsi eksplorasi depth-first dengan penyimpanan multi-solusi serta menerapkan aturan pruning untuk menjaga efisiensi komputasi. Jalur akan dipangkas apabila memasuki obstacle, mengunjungi node yang sama, melebihi batas langkah, atau memiliki akumulasi risiko di atas ambang batas yang ditentukan. Strategi ini memungkinkan algoritma tetap efisien meskipun mengeksplorasi banyak kemungkinan jalur, sekaligus memastikan bahwa hanya jalur evakuasi yang aman dan realistis yang dipertahankan, sejalan dengan pendekatan evaluasi risiko pada sistem evakuasi modern [13], [2], [10], [12].

* 1. **Path Validation and Multi-Solution Handling**

Algoritma backtracking dirancang untuk menghasilkan himpunan jalur evakuasi aman (multi-solution) dengan memvalidasi setiap jalur yang mencapai titik keluar berdasarkan kriteria keamanan dan efisiensi. Pendekatan multi-solution ini penting dalam skenario darurat yang dinamis, karena menyediakan rute alternatif ketika jalur utama menjadi tidak layak akibat perubahan kondisi lingkungan [2], [8]. **Path Validation.** Suatu jalur dinyatakan valid apabila berhasil menghubungkan start ke exit, tidak melewati obstacle, memiliki panjang langkah di bawah batas maksimum, serta memiliki akumulasi risiko di bawah ambang yang ditetapkan. Validasi ini memastikan jalur tidak hanya layak secara struktural, tetapi juga aman dari perspektif risiko lingkungan, sesuai dengan prinsip evaluasi keselamatan jalur evakuasi berbasis grid [14], [15]. **Multi-Solution Handling.** Setiap jalur yang valid disimpan dalam kumpulan solusi tanpa menghentikan proses pencarian pada solusi pertama. Algoritma terus mengeksplorasi hingga diperoleh minimal 10 jalur evakuasi alternatif, yang selanjutnya dibandingkan berdasarkan panjang jalur dan tingkat risiko untuk menentukan jalur terbaik atau rekomendasi rute. Strategi ini mendukung konsep redundant evacuation routing yang menekankan fleksibilitas dan keandalan rute pada evakuasi gedung nyata [12], [11]



1. Jalur Evakuasi Multi-Solusi

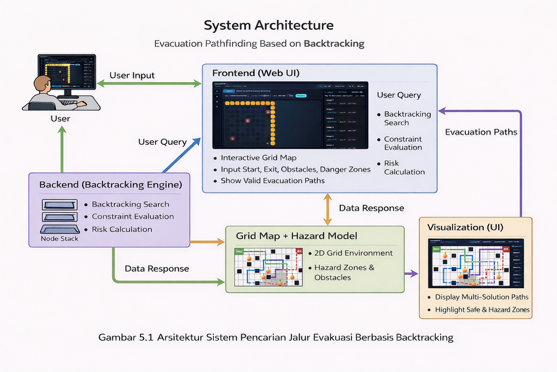
Gambar ilustrasi jalur evakuasi multi-solusi menunjukkan hasil eksplorasi algoritma backtracking pada lingkungan gedung berbasis grid graph dengan keberadaan obstacle dan zona bahaya. Jalur valid ditunjukkan oleh lintasan yang berhasil menghubungkan titik awal (start) ke titik keluar (exit) tanpa melanggar batasan lingkungan dan dengan tingkat risiko yang dapat diterima, sedangkan jalur lain dipangkas (pruned) karena memasuki area berisiko, mendekati obstacle, atau melebihi batas langkah. Ilustrasi ini menegaskan kemampuan algoritma backtracking dalam menghasilkan beberapa alternatif jalur evakuasi aman (multi-solution search), sehingga meningkatkan fleksibilitas dan keandalan proses evakuasi dibandingkan pendekatan pencarian jalur konvensional berbasis satu solusi.

* 1. **Time and Space Complexity Analysis**

Algoritma backtracking mengeksplorasi jalur evakuasi secara rekursif pada grid graph dengan mempertimbangkan seluruh kemungkinan pergerakan dari titik awal menuju titik keluar. Pada kondisi terburuk, kompleksitas waktu algoritma dinyatakan sebagai , dengan sebagai faktor percabangan dan sebagai kedalaman pencarian atau panjang jalur maksimum. Pada grid empat arah, nilai dibatasi hingga empat, namun dapat meningkat seiring ukuran lingkungan dan batas langkah yang ditetapkan [1], [2], [3]. Kompleksitas ruang terutama dipengaruhi oleh kedalaman rekursi dan penyimpanan jalur aktif, yakni sebesar untuk call stack, ditambah memori untuk menyimpan solusi yang valid. Penerapan pruning berbasis obstacle, batas langkah, dan ambang risiko secara signifikan mengurangi ruang pencarian, sehingga kompleksitas aktual jauh lebih rendah dibandingkan skenario terburuk teoritis [7] [8] Meskipun terdapat trade-off antara jumlah solusi dan waktu komputasi, pembatasan jumlah solusi (≥10 jalur) dipilih sebagai kompromi yang seimbang antara efisiensi dan fleksibilitas evakuasi, sesuai dengan rekomendasi pada perencanaan evakuasi multi-rute modern [5].

1. **Hasil Penelitian**
   1. **System Architecture**

Arsitektur sistem pada penelitian ini menggunakan pendekatan client–server yang memisahkan antarmuka pengguna (frontend) dan pemrosesan algoritma (backend) untuk meningkatkan modularitas serta memastikan komputasi algoritma backtracking berjalan terisolasi dari proses visualisasi. Pendekatan ini umum diterapkan pada sistem evakuasi berbasis simulasi guna menjaga konsistensi antara model lingkungan, algoritma pencarian jalur, dan hasil visualisasi [11]. Frontend berfungsi sebagai antarmuka interaktif untuk memodelkan lingkungan gedung berbasis grid, menentukan posisi awal, titik keluar, zona bahaya, dan hambatan, serta menampilkan visualisasi jalur evakuasi menggunakan perbedaan warna. Visualisasi ini membantu pengguna memahami alternatif rute evakuasi yang tersedia secara intuitif [8], [5]. Sementara itu, backend menangani pemodelan grid, evaluasi kendala, perhitungan risiko, dan eksekusi algoritma backtracking untuk menghasilkan beberapa jalur evakuasi aman (multi-solution paths). Data dikirim dari frontend ke backend untuk diproses dan hasil jalur valid dikembalikan untuk divisualisasikan kembali, sebagaimana digambarkan pada Gambar 5.1 [10].



1. Arsitektur Sistem Pencarian Jalur Evakuasi Berbasis Backtracking

Gambar 3 ini menunjukkan arsitektur sistem aplikasi pencarian jalur aman evakuasi gedung berbasis algoritma backtracking. Sistem terdiri dari tiga komponen utama, yaitu **Frontend (Web UI), Backend (Backtracking Engine),** serta **Grid Map dan Hazard Model**. Pengguna berinteraksi melalui antarmuka web untuk menentukan parameter evakuasi seperti posisi awal (start), titik keluar (exit), rintangan (obstacle), dan zona bahaya (hazard). Data tersebut diproses oleh backend menggunakan pencarian rekursif backtracking untuk mengeksplorasi berbagai kemungkinan jalur, melakukan evaluasi risiko, serta memfilter jalur yang tidak aman. Hasil perhitungan berupa beberapa jalur evakuasi aman kemudian dikirim kembali ke frontend dan divisualisasikan dalam bentuk peta grid interaktif yang menampilkan jalur terbaik, alternatif jalur, serta zona bahaya secara visual, sehingga mendukung pengambilan keputusan evakuasi secara cepat dan informatif.

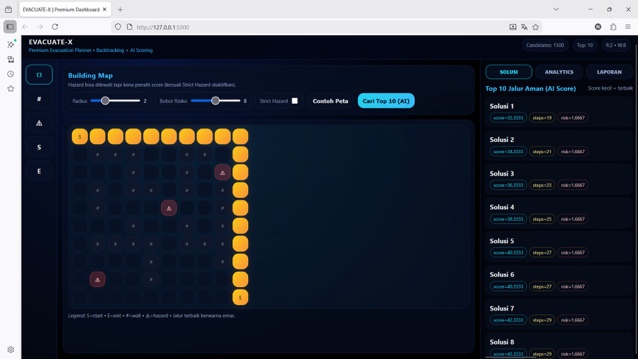
* 1. **Data Structures and Implementation Details**

Grid Map Data Representation Lingkungan gedung pada sistem evakuasi dimodelkan menggunakan grid dua dimensi (2D grid map) berukuran 10 × 10, di mana setiap sel merepresentasikan satu unit ruang diskrit. Setiap unit ruang dapat berupa area aman yang dapat dilalui, rintangan permanen (*obstacle*), atau zona bahaya (*hazard*) yang memiliki tingkat risiko tertentu. Representasi berbasis grid ini menjadi fondasi utama dalam proses pencarian jalur evakuasi menggunakan algoritma *backtracking*, karena memudahkan pemetaan node, evaluasi batasan pergerakan, serta eksplorasi jalur secara sistematis.Representasi Data Grid Pada sisi backend (app.py), grid direpresentasikan dalam bentuk array dua dimensi dengan nilai numerik diskrit. Setiap nilai memiliki makna semantik yang selaras dengan visualisasi pada antarmuka pengguna. Pemetaan nilai grid ditunjukkan pada Tabel berikut.

1. Pemetaan nilai grid

|  |  |  |
| --- | --- | --- |
| Kode | Simbol UI | Makna |
| 0 | Kosong | Sel aman dan dapat dilalui |
| 1 | # | Obstacle / dinding (tidak dapat dilalui) |
| 2 | ⚠ | Hazard / zona bahaya (dapat dilalui dengan penalti risiko) |

Gambar yang ditampilkan menunjukkan visualisasi grid map 10 × 10 yang dibangun langsung dari data numerik tersebut, di mana sel kosong divisualisasikan sebagai area netral, obstacle ditandai dengan simbol #, dan zona bahaya ditunjukkan dengan ikon peringatan.

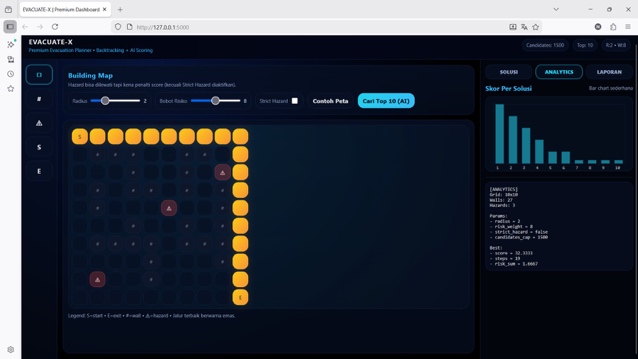
****

1. Visualisasi grid map UI

Grid map dikirim langsung dari frontend ke backend melalui request berbasis JSON saat pengguna menekan tombol **“Cari Top 10 (AI)”,** sehingga representasi data antara antarmuka dan proses komputasi tetap konsisten. Selain grid, sistem mendefinisikan koordinat **Start** dan **Exit** sebagai pasangan (baris, kolom) yang selaras dengan indeks array pada Python dan JavaScript, dan digunakan sebagai acuan awal serta kondisi terminasi pada algoritma backtracking. Pada sisi frontend, grid divisualisasikan sebagai peta interaktif yang memetakan nilai sel (0, 1, 2) ke warna dan ikon untuk membedakan area aman, obstacle, dan zona bahaya. Jalur hasil pencarian kemudian ditampilkan di atas grid dengan penanda visual khusus, sehingga alur **data grid → algoritma backtracking → visualisasi jalur** terintegrasi secara langsung dan konsisten dalam merepresentasikan lingkungan evakuasi.

* 1. **User Interface and Visualization**

Antarmuka pengguna sistem evakuasi dikembangkan berbasis web untuk menampilkan peta gedung dalam bentuk grid interaktif yang merepresentasikan area aman, obstacle, dan zona bahaya berdasarkan data grid map dari backend. Visualisasi berbasis grid memudahkan pengguna memahami struktur ruang dan tingkat risiko secara intuitif, sesuai dengan rekomendasi penelitian evakuasi berbasis simulasi [1], [2]. Jalur evakuasi divisualisasikan menggunakan skema warna konsisten, dengan jalur terbaik hasil algoritma backtracking ditampilkan secara kontras, sehingga memudahkan identifikasi rute aman [8], [5]. Selain peta utama, antarmuka menyediakan panel solusi yang menampilkan beberapa alternatif jalur evakuasi beserta metrik pendukung, yang diperbarui secara dinamis setelah proses pencarian selesai. Dengan pendekatan ini, antarmuka berfungsi tidak hanya sebagai media visualisasi, tetapi juga sebagai alat bantu pengambilan keputusan evakuasi yang informatif dan responsif [3], [7].



1. Tampilan Antarmuka Pengguna (UI) Sistem Evakuasi Gedung Berbasis Backtracking

**Gambar 5.** Tampilan antarmuka pengguna sistem evakuasi gedung berbasis backtracking yang menampilkan grid map, zona bahaya, jalur evakuasi terbaik, serta daftar alternatif jalur (multi-solution) beserta metrik evaluasinya.

* 1. **Experimental Setup**

Eksperimen dilakukan untuk mengevaluasi kinerja algoritma backtracking dalam menemukan jalur evakuasi aman pada lingkungan gedung berbasis grid map, dengan menilai kemampuan menghasilkan banyak solusi, menangani obstacle dan zona bahaya, serta menyeimbangkan panjang jalur dan risiko. Pengujian diimplementasikan menggunakan sistem **EVACUATE-X** yang mengintegrasikan visualisasi frontend dan pemrosesan backend, sejalan dengan pendekatan sistem evakuasi terintegrasi pada penelitian sebelumnya [1], [11]. Lingkungan uji dimodelkan sebagai grid 10 × 10 dengan obstacle dan zona bahaya statis, di mana zona bahaya tetap dapat dilalui dengan penalti risiko untuk merefleksikan kondisi darurat nyata [8], [5]. Pada setiap skenario, algoritma menghasilkan sejumlah jalur valid yang kemudian diseleksi menjadi Top-10 jalur terbaik berdasarkan kombinasi panjang jalur dan risiko, sehingga memungkinkan evaluasi komprehensif terhadap keunggulan backtracking dibandingkan metode pencarian jalur solusi tunggal [3], [6].

1. Parameter Eksperimen

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Nilai** | **Keterangan** |
| Ukuran grid | 10 × 10 | Representasi lingkungan gedung berbasis grid |
| Jumlah sel total | 100 | Total node pada grid map |
| Jumlah obstacle | 27 | Sel tidak dapat dilalui (*wall*) |
| Jumlah zona bahaya | 3 | Sel berisiko dengan penalti |
| Posisi start (S) | (0, 0) | Titik awal evakuasi |
| Posisi exit (E) | (9, 9) | Titik tujuan evakuasi |
| Metode pencarian | Backtracking | Pencarian rekursif *depth-first* |
| Mode solusi | Multi-solution | Menghasilkan banyak jalur valid |
| Batas solusi | Top-10 jalur | Jalur terbaik berdasarkan skor |
| Parameter risiko | Risk weight = 8 | Bobot penalti zona bahaya |
| Radius risiko | R = 2 | Jarak pengaruh zona bahaya |
| Strict hazard | Non-aktif | Zona bahaya masih dapat dilewati |

Table 2. merangkum parameter eksperimen untuk mengevaluasi algoritma backtracking pada pencarian jalur evakuasi berbasis grid map berukuran 10 × 10. Lingkungan uji terdiri atas sel aman, obstacle, dan zona bahaya yang dapat dilalui dengan penalti risiko, sehingga evaluasi jalur tidak hanya berbasis jarak tetapi juga aspek keselamatan [1], [8]. Algoritma dijalankan secara rekursif depth-first untuk menghasilkan banyak jalur valid (multi-solution), kemudian dipilih Top-10 jalur terbaik berdasarkan skor gabungan panjang jalur dan risiko, dengan pengaturan bobot risiko dan radius pengaruh zona bahaya tertentu [5], [9]

* 1. **Performance Metrics**

Evaluasi kinerja algoritma backtracking pada sistem **EVACUATE-X** dilakukan menggunakan metrik jumlah jalur aman, panjang jalur, dan waktu komputasi untuk menilai fleksibilitas, efisiensi, dan kelayakan algoritma pada skenario evakuasi gedung [1], [13] Pada lingkungan grid 10 × 10, algoritma mampu menghasilkan sejumlah jalur evakuasi valid (multi-solution), yang kemudian diseleksi menjadi Top-10 jalur aman terbaik, menunjukkan keunggulan backtracking dibandingkan metode pencarian jalur tunggal [9], [6] Panjang jalur solusi terpilih berada pada rentang 19–29 langkah, mencerminkan bahwa sistem mengoptimalkan jarak sekaligus tingkat risiko terhadap zona bahaya [5], [8]. Dari sisi efisiensi, proses pencarian dan seleksi jalur dapat diselesaikan dalam waktu kurang dari satu detik berkat penerapan pruning dan pembatasan kandidat solusi, sehingga sistem cukup responsif untuk aplikasi evakuasi berbasis web [9], [13]

1. Ringkasan Hasil Performance Metrics

|  |  |
| --- | --- |
| **Metrik** | **Hasil Eksperimen** |
| Jumlah jalur aman | Puluhan jalur valid, ditampilkan Top-10 |
| Panjang jalur (steps) | 19 – 29 langkah |
| Waktu komputasi | < 1 detik (grid 10 × 10) |

Hasil pada Tabel 3. menunjukkan bahwa algoritma backtracking mampu menghasilkan puluhan jalur evakuasi valid dan menyeleksi Top-10 jalur aman, menunjukkan keunggulan pendekatan multi-solution dibandingkan metode jalur tunggal [1], [14] Panjang jalur berada pada rentang 19–29 langkah, mencerminkan kompromi antara efisiensi rute dan tingkat risiko zona bahaya. Seluruh proses pencarian dapat diselesaikan dalam waktu kurang dari satu detik pada grid 10 × 10 melalui mekanisme pruning, sehingga sistem dinilai responsif dan layak diterapkan pada simulasi evakuasi berbasis web [7], [14]

* 1. **Comparative Analysis**

Analisis komparatif menunjukkan bahwa algoritma backtracking memiliki keunggulan signifikan dibandingkan BFS dan DFS dalam konteks evakuasi gedung berbasis grid. BFS efektif dalam menemukan jalur terpendek berdasarkan jumlah langkah, namun hanya menghasilkan satu solusi sehingga kurang fleksibel ketika jalur utama terhalang atau berisiko tinggi [1], [6]. DFS memang mengeksplorasi ruang pencarian secara mendalam, tetapi umumnya berhenti setelah menemukan satu jalur tanpa evaluasi risiko dan tanpa mekanisme seleksi solusi, sehingga sering menghasilkan lintasan yang panjang dan kurang aman [10]. Sebaliknya, backtracking mampu menelusuri seluruh jalur valid, menerapkan pruning, serta menyimpan banyak solusi evakuasi aman (multi-solution), menjadikannya lebih sesuai untuk skenario darurat yang menuntut alternatif rute. Dibandingkan dengan algoritma pencarian global seperti Dijkstra dan Ant Colony Optimization (ACO), backtracking menawarkan kompromi yang lebih seimbang antara performa dan kompleksitas implementasi. Dijkstra memerlukan bobot graf yang eksplisit dan cenderung menghasilkan satu solusi optimal, sementara ACO bersifat adaptif namun membutuhkan iterasi berulang, parameter tuning, serta biaya komputasi yang lebih tinggi, sehingga kurang efisien untuk aplikasi web interaktif berskala kecil [5], [8]. Dalam sistem EVACUATE-X, backtracking terbukti mampu menghasilkan banyak jalur evakuasi aman dengan waktu komputasi rendah dan implementasi yang relatif sederhana, sehingga relevan untuk simulasi dan sistem pendukung keputusan evakuasi gedung [3], [11]

1. Perbandingan Algoritma Pencarian Jalur Evakuasi

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Kriteria** | **Backtracking** | **BFS** | **DFS** | **Dijkstra** | **ACO** |
| Jumlah solusi | Banyak (multi-solution) | 1 | 1 | 1 | Banyak |
| Optimalitas jarak | Tidak selalu minimum | Minimum | Tidak optimal | Optimal | Mendekati optimal |
| Evaluasi risiko | Ya | Tidak | Tidak | Bergantung bobot | Ya |
| Kompleksitas | Tinggi (terkendali dengan pruning) | Rendah | Rendah | Sedang | Tinggi |
| Parameter tambahan | Tidak | Tidak | Tidak | Bobot graf | Banyak |
| Cocok untuk UI interaktif | Ya | Terbatas | Tidak | Terbatas | Tidak |
| Kesesuaian evakuasi gedung | Sangat sesuai | Cukup | Kurang | Cukup | Kurang |

Berdasarkan hasil perbandingan, algoritma backtracking menunjukkan keunggulan utama dalam **kemampuan menghasilkan banyak jalur evakuasi aman,** fleksibilitas terhadap kondisi lingkungan, serta integrasi yang baik dengan visualisasi UI interaktif. Meskipun memiliki kompleksitas pencarian yang lebih tinggi dibandingkan BFS dan DFS, penerapan pruning dan pembatasan solusi menjadikan backtracking tetap efisien dan praktis untuk digunakan pada sistem evakuasi gedung berbasis grid.

**4. Kesimpulan**

Penelitian ini menyimpulkan bahwa algoritma backtracking efektif untuk pencarian jalur evakuasi gedung berbasis grid karena mampu menghasilkan banyak alternatif jalur aman dengan mempertimbangkan obstacle dan zona bahaya. Pendekatan multi-jalur ini meningkatkan fleksibilitas sistem dibandingkan metode jalur tunggal, terutama pada kondisi darurat yang dinamis. Integrasi algoritma dengan visualisasi berbasis web juga mempermudah pemahaman pengguna terhadap kondisi lingkungan dan pilihan rute, sehingga mendukung pengambilan keputusan evakuasi yang lebih informatif dan adaptif [10]

**5. Saran**

Diperlukan adanya penelitian lebih lanjut analisis kestabilan lereng dengan perangkat lunak geostudio yang berlisensi agar dapat dibuat simulasi retaining wall serta back filling, sehingga akan didapatkan FK lebih akurat.

**6. Daftar Pustaka**

[1] N. Z. Latif, “Makalah IF2120 Matematika Diskrit-Sem. I Tahun,” 2018.

[2] Y. Hong, D. Li, Q. Wu, and H. Xu, “Dynamic Route Network Planning Problem for Emergency Evacuation in Restricted-Space Scenarios,” *J. Adv. Transp.*, vol. 2018, 2018, doi: 10.1155/2018/4295419.

[3] K. Deng, Q. Zhang, H. Zhang, P. Xiao, and J. Chen, “Optimal Emergency Evacuation Route Planning Model Based on Fire Prediction Data,” *Mathematics*, vol. 10, no. 17, Sep. 2022, doi: 10.3390/math10173146.

[4] B. Yang *et al.*, “A novel heuristic emergency path planning method based on vector grid map,” *ISPRS Int. J. Geoinf.*, vol. 10, no. 6, Jun. 2021, doi: 10.3390/ijgi10060370.

[5] J. K. N. Tan, S. Zhang, A. W. K. Law, and S. H. Cheung, “Digital-twin enabled evacuation to improve individual and community resilience of building occupants against indoor Fires: Framework and route-finding algorithm,” *Developments in the Built Environment*, vol. 22, Apr. 2025, doi: 10.1016/j.dibe.2025.100672.

[6] N. Halisah, A. Maheswari Artalaysia Paturusi, and W. Firgiawan, “Metode Optimasi Cerdas Untuk Pencarian Jalur Evakuasi Bencana di Kabupaten Mamuju Menggunakan Ant Colony Optimization,” *Journal of Computer and Information System (J-CIS*, vol. 8, no. 2, 2025, doi: 10.31605/jcis.v8i2.

[7] H. I. Alrahma, Anan Nugroho, A. F. Hastawan, and U. M. Arief, “Application of Q-learning Method for Disaster Evacuation Route Design Case Study: Digital Center Building UNNES,” *Jurnal Ilmu Komputer dan Informasi*, vol. 17, no. 2, pp. 145–158, Jun. 2024, doi: 10.21609/jiki.v17i2.1236.

[8] B. Pang, J. Shi, L. Jiang, and Z. Pan, “A Semantic Approach to Dynamic Path Planning for Fire Evacuation through BIM and IoT Data Integration,” *Advances in Civil Engineering*, vol. 2024, 2024, doi: 10.1155/2024/8839865.

[9] J. P. Rastelli *et al.*, *Path Following with Backtracking Based on Fuzzy Controllers for Fordward and Reverse Driving*. 2012. doi: 10.0/Linux-x86\_64.

[10] K. Deng, Q. Zhang, H. Zhang, P. Xiao, and J. Chen, “Optimal Emergency Evacuation Route Planning Model Based on Fire Prediction Data,” *Mathematics*, vol. 10, no. 17, Sep. 2022, doi: 10.3390/math10173146.

[11] K. Shahabi and J. P. Wilson, “Scalable evacuation routing in a dynamic environment,” *Comput. Environ. Urban Syst.*, vol. 67, pp. 29–40, Jan. 2018, doi: 10.1016/j.compenvurbsys.2017.08.011.

[12] Y. Zhu *et al.*, “Optimal Evacuation Route Planning of Urban Personnel at Different Risk Levels of Flood Disasters Based on the Improved 3D Dijkstra’s Algorithm,” *Sustainability (Switzerland)*, vol. 14, no. 16, Aug. 2022, doi: 10.3390/su141610250.

[13] Kayoko Yamamoto and Ximing Li, “Safety Evaluation Method of Evacuation Routes in Urban Areas in Case of Earthquake Disasters Using Ant Colony Optimization Algorithm and Geographic Information Systems,” *Journal of Environmental Science and Engineering A*, vol. 6, no. 9, Sep. 2017, doi: 10.17265/2162-5298/2017.09.004.

[14] R.-G. Tsai *et al.*, “ERP: Real-time Evacuation Route Planning Algorithm for Fire Scenes in Smart Buildings.” [Online]. Available: https://ssrn.com/abstract=4734657

[15] Q. Huang, Y. Si, and H. Wang, “Fire Evacuation Path Planning Based on Improved MADDPG (Multi-Agent Deep Deterministic Policy Gradient) Algorithm.” [Online]. Available: www.ijacsa.thesai.org