

Laporan Project UTS



AI : Maze Solver

Dosen Pengampu:

Indah Werdiningsih, S.Si., M.Kom

Disusun Oleh:

- | | |
|-------------------------------------|------------------|
| 1. Noga Salsabilla Aalalfala | 434231007 |
| 2. Ahmad Lazim | 434231033 |
| 3. Inunk Rodliyah | 434231063 |
| 4. Mohammad Hafidz Al Maaher | 434231117 |

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS VOKASI

UNIVERSITAS AIRLANGGA

2025

Daftar isi

1. Pendahuluan.....	2
2. Tema Proyek.....	2
3. Identifikasi Ruang Keadaan.....	2
4. Keadaan Awal Dan Tujuan.....	3
5. Pohon Pelacakan Graph.....	4
6. BFS dalam Pencarian Jalur.....	5
7. Implementasi AI dengan Python.....	5
8. Struktur Data yang Digunakan.....	5
9. Visualisasi dengan Pygame.....	6
10. Pengujian dan Evaluasi Kinerja.....	6
11. Kesimpulan dan Rekomendasi.....	6

1. Pendahuluan

Dalam dunia kecerdasan buatan, pencarian jalur merupakan salah satu topik yang penting dalam pemecahan masalah berbasis graf. Salah satu permasalahan klasik dalam bidang ini adalah menemukan jalur terpendek dalam sebuah labirin. Oleh karena itu, proyek **AI Maze Solver** ini bertujuan untuk mengembangkan sistem kecerdasan buatan yang dapat mencari jalur terpendek dalam sebuah labirin menggunakan algoritma **Breadth-First Search (BFS)**.

2. Tema Proyek



Tema proyek ini adalah "**AI Maze Solver**", yaitu pembuatan sistem kecerdasan buatan yang dapat menemukan jalur terpendek dalam sebuah labirin menggunakan algoritma **Breadth-First Search (BFS)**. Proyek ini juga mencakup visualisasi labirin menggunakan pygame untuk membantu pengguna memahami bagaimana algoritma bekerja.

3. Identifikasi Ruang Keadaan

Ruang keadaan dalam proyek ini direpresentasikan sebagai grid 5x5. Ruang keadaan dalam konteks ini adalah semua posisi yang dapat diambil dalam labirin, seperti berikut:

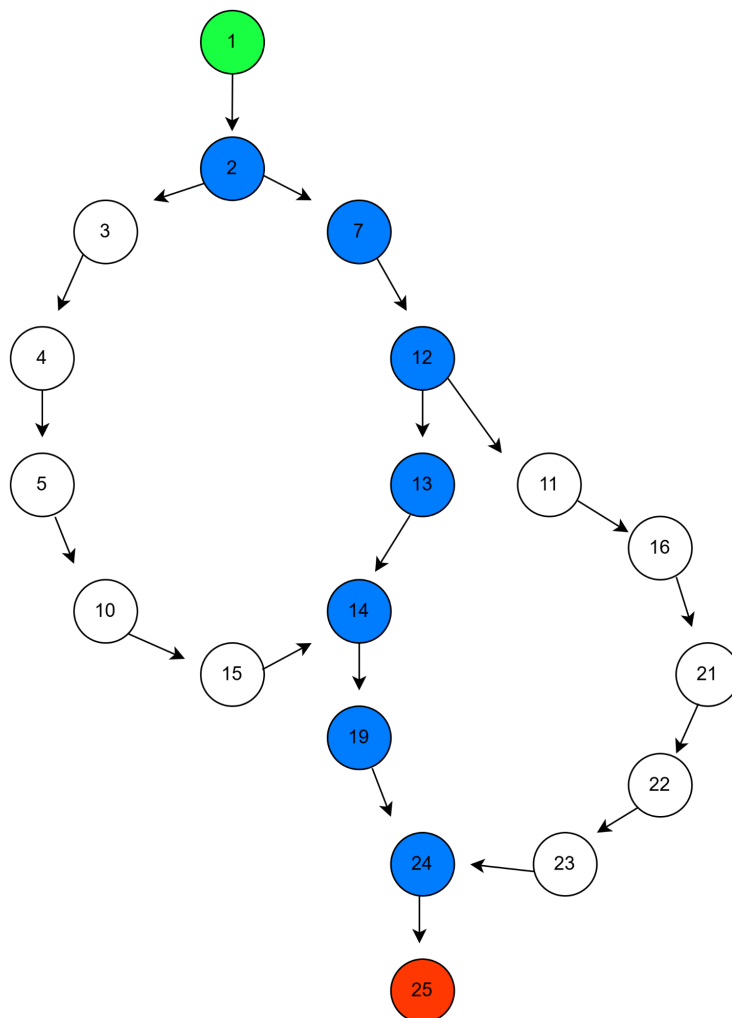
1	2	3	4	5
6	7	8	9	10
11	12		14	15
16	17	18	19	20
21	22	23	24	25

Di mana setiap kotak warna memiliki artian :

-  : Titik awal
-  : Kotak yang dapat dilewati.

- : Kotak yang tidak dapat dilewati.
- : Titik akhir.
- : Posisi saat ini.

4. Keadaan Awal Dan Tujuan



- Keadaan awal, pada labirin :
- Daerah asal pada labirin :



- Tujuan, pada labirin :
- Daerah akhir pada labirin :



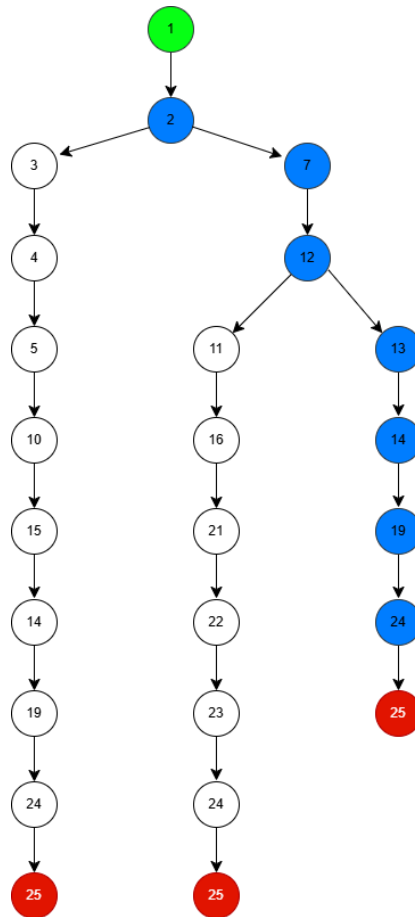
BFS dalam Pencarian Jalur

1. BFS dimulai dari **Start (1)**.
2. Menjelajahi semua jalur yang dapat ditempuh hingga mencapai **End (25)**.
3. BFS memastikan jalur yang ditemukan adalah jalur **terpendek**.
4. Jalur yang ditemukan akan dikonversi ke dalam urutan angka.

5. Pohon Pelacakan Dari Graph

Pohon pelacakan menggambarkan bagaimana BFS bekerja dalam menemukan jalur terpendek dalam labirin dengan langkah-langkah berikut:

1. BFS dimulai dari titik awal (Start).
2. BFS menjelajahi semua jalur yang dapat ditempuh hingga mencapai titik akhir (End).
3. BFS memastikan bahwa jalur yang ditemukan adalah jalur terpendek.
4. Jalur yang ditemukan akan dikonversi ke dalam bentuk representasi angka untuk dianalisis lebih lanjut.



6. BFS dalam Pencarian Jalur

Algoritma **BFS** dalam pencarian jalur bekerja dengan menjelajahi semua kemungkinan gerakan yang dapat dilakukan dari titik awal hingga titik akhir secara sistematis. **BFS** menggunakan struktur data antrian (**queue**) untuk mengelola proses pencarian dan memastikan bahwa setiap simpul yang dieksplorasi memiliki jarak minimal ke titik awal.

7. Implementasi AI dengan Python

Implementasi dilakukan menggunakan **pygame** dan **BFS**:

- **Pygame** digunakan untuk visualisasi grid dan pergerakan agen dalam labirin.
- **BFS** mencari jalur terpendek menuju titik akhir berdasarkan eksplorasi simpul.
- Setiap langkah **BFS** ditampilkan dalam bentuk animasi pergerakan dalam **pygame** untuk memberikan pemahaman lebih intuitif bagi pengguna.

8. Struktur Data yang Digunakan

Proyek ini menggunakan beberapa struktur data utama dalam implementasi **BFS**, yaitu:

- **Queue (Antrian):** Digunakan untuk menyimpan simpul yang akan dieksplorasi.
- **Graph (Graf):** Merepresentasikan hubungan antara simpul-simpul dalam labirin.
- **Dictionary:** Menyimpan informasi terkait jalur yang telah ditempuh.

9. Visualisasi dengan Pygame

Pygame digunakan untuk menampilkan labirin dan pergerakan **BFS** dalam bentuk animasi:

- Grid labirin divisualisasikan dengan warna yang berbeda untuk membedakan rintangan, jalur yang bisa dilewati, dan titik awal serta akhir.
- Pergerakan **BFS** divisualisasikan dalam bentuk animasi yang memperlihatkan proses eksplorasi jalur.

10. Pengujian dan Evaluasi Kinerja

Proses pengujian dilakukan dengan beberapa skenario labirin:

- **Labirin sederhana:** Untuk menguji apakah **BFS** dapat menemukan solusi optimal.
- **Labirin kompleks:** Untuk mengamati kinerja **BFS** dalam menangani jumlah simpul yang lebih besar.
- **Analisis waktu eksekusi:** Untuk mengevaluasi efisiensi **BFS** dalam pencarian jalur.

11. Kesimpulan dan Rekomendasi

Proyek AI Maze Solver ini telah berhasil mengimplementasikan:

- Pencarian jalur terpendek dalam labirin menggunakan **BFS**.
- Visualisasi labirin dengan pygame untuk mempermudah pemahaman algoritma.
- Interaksi pengguna untuk menambah/menghapus rintangan serta melihat solusi jalur.

Sebagai rekomendasi pengembangan lebih lanjut:

- Menggunakan algoritma pencarian lainnya seperti **A*** untuk membandingkan efisiensi pencarian jalur.
- Mengembangkan fitur tambahan seperti pembuatan labirin secara otomatis berdasarkan input pengguna.