

BAB 2

LANDASAN TEORI

2.1 Penelitian Terdahulu

Dalam konteks optimasi jalur instalasi kabel pada proyek konstruksi industri, permasalahan pencarian jalur terpendek umumnya dimodelkan dalam bentuk *graf* berbobot. Algoritma Dijkstra merupakan salah satu algoritma berbasis *graf* yang dirancang untuk menentukan jalur terpendek pada *graf* dengan bobot non-negatif secara deterministik. Karakteristik tersebut menjadikan algoritma Dijkstra dapat digunakan dalam perencanaan jalur instalasi listrik terpendek.

Rekonstruksi model tiga dimensi dari denah dua dimensi telah dibahas dalam beberapa penelitian sebagai cara untuk membentuk model bangunan dari gambar denah. Feist et al. menjelaskan bahwa informasi pada denah dua dimensi dapat dianalisis untuk mengenali elemen bangunan dan kemudian digunakan dalam pembentukan model tiga dimensi. Selain itu, Poku-Agyemang dan Reiterer menunjukkan bahwa gambar denah dua dimensi dapat dimanfaatkan sebagai dasar untuk membangun representasi tiga dimensi dari struktur bangunan [18] [19].

Pada bidang algoritma pencarian jalur, Cormen et al. menjelaskan algoritma Dijkstra sebagai metode untuk menentukan jalur terpendek pada *graf* dengan bobot non-negatif. Millington juga menyatakan bahwa algoritma Dijkstra sesuai digunakan pada lingkungan berbasis *grid* dan mampu menghasilkan solusi optimal secara deterministik [20]. Oleh karena itu, algoritma Dijkstra sering dijadikan dasar dalam permasalahan pencarian jalur yang membutuhkan ketepatan perhitungan jarak, termasuk pada optimasi jalur instalasi kabel dalam bangunan.

Representasi ruang tiga dimensi untuk keperluan navigasi dan pencarian jalur dapat dilakukan menggunakan pendekatan voxel, di mana ruang dibagi menjadi unit-unit kecil yang merepresentasikan kondisi lingkungan [21]. Pendekatan ini mendukung pemodelan *graf* tiga dimensi yang sesuai untuk pencarian jalur pada ruang bangunan. Untuk keperluan visualisasi, digunakan Unity sebagai game engine lintas platform yang mendukung pengembangan aplikasi 3D secara prosedural. Unity memungkinkan pemodelan objek dan simulasi spasial yang berjalan secara mandiri pada perangkat pengguna.

Tabel 2.1. Perbandingan Penelitian Terdahulu dengan Penelitian Ini

Deskripsi	Rahman (2023)	Hasan (2021)	Pratama (2022)	Penelitian Ini
Rekonstruksi Model 3D	✓	-	-	✓
Algoritma Dijkstra	-	✓	-	✓
Visualisasi Interaktif 3D	-	-	✓	✓
Integrasi Sistem	-	-	-	✓

Tabel 2.1 menunjukkan perbandingan antara penelitian terdahulu dan penelitian yang dilakukan dalam skripsi ini. Berdasarkan tabel tersebut, dapat dilihat bahwa penelitian ini mengintegrasikan rekonstruksi model 3D, algoritma Dijkstra, visualisasi interaktif, serta implementasi WebGL dalam satu sistem yang utuh. Hal ini membedakan penelitian ini dari penelitian sebelumnya yang umumnya hanya berfokus pada satu atau dua aspek saja.

2.2 Rekonstruksi Model 3D dari Denah 2D

Rekonstruksi model tiga dimensi dari gambar dua dimensi merupakan proses penting dalam bidang grafika komputer. Proses ini melibatkan segmentasi citra, vektorisasi, serta ekstrusi untuk memberikan dimensi kedalaman pada bentuk dua dimensi. Pendekatan yang digunakan dalam penelitian ini adalah metode berbasis piksel (*pixel-based*), di mana piksel berwarna gelap diasumsikan sebagai dinding dan piksel terang sebagai ruang kosong. Pendekatan ini digunakan untuk membangun struktur 3D secara prosedural.

2.3 Algoritma Dijkstra

Algoritma Dijkstra bekerja dengan cara menginisialisasi jarak semua simpul ke nilai tak hingga, kecuali simpul sumber yang diberi jarak nol. Pada setiap iterasi, algoritma memilih simpul dengan jarak terpendek yang belum dikunjungi, kemudian memperbarui jarak ke simpul tetangganya jika ditemukan jalur yang lebih pendek. Proses ini diulang hingga seluruh simpul telah diproses atau simpul tujuan telah ditemukan, sehingga jalur terpendek dapat ditentukan secara optimal. [22]

Proses pembaruan jarak pada algoritma Dijkstra mengikuti aturan matematis sebagaimana ditunjukkan pada Persamaan.

$$\text{dist}(v) = \begin{cases} \text{dist}(u) + w(u, v), & \text{jika } \text{dist}(u) + w(u, v) < \text{dist}(v) \\ \text{dist}(v), & \text{lainnya} \end{cases} \quad (2.1)$$

Keterangan:

- $\text{dist}(u)$ menyatakan jarak terpendek dari simpul sumber ke simpul u .
- $\text{dist}(v)$ menyatakan jarak terpendek dari simpul sumber ke simpul v .
- $w(u, v)$ merupakan bobot sisi antara simpul u dan v .

Dalam penelitian ini, struktur bangunan direpresentasikan sebagai *graf* tiga dimensi berbasis voxel, di mana setiap voxel merepresentasikan satu simpul dalam ruang tiga dimensi. Representasi berbasis voxel memungkinkan penyederhanaan model lingkungan dan pemrosesan jalur karena semua simpul dan hubungan antar simpul bisa dipetakan secara sistematis. Algoritma Dijkstra dinilai sesuai untuk lingkungan berbasis *grid*, karena algoritma ini mampu menjamin solusi jalur terpendek yang optimal pada *graf* berbobot positif, sebagaimana ditunjukkan dalam studi yang mengevaluasi Dijkstra dalam konteks *pathfinding* pada *grid* [23].

2.4 Unity

Unity merupakan *game engine* lintas *platform* yang mendukung pengembangan aplikasi interaktif dua dan tiga dimensi menggunakan bahasa pemrograman C-Sharp. Unity digunakan untuk memvisualisasikan hasil rekonstruksi denah dua dimensi ke dalam model tiga dimensi serta menampilkan jalur terpendek hasil perhitungan algoritma Dijkstra secara interaktif. Unity mendukung pemrosesan grafis tinggi dan manajemen memori yang stabil, sehingga aplikasi dapat menangani perhitungan graf berukuran besar. [24]