

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Menurut Undang-Undang terkait Bangunan Gedung Nomor 28 Tahun 2002 Pasal 5 Ayat 1, Bangunan gedung merupakan wujud fisik dari aktivitas konstruksi yang terletak di dalam atau di atas permukaan tanah ataupun air, yang secara utama berfungsi sebagai tempat manusia melakukan aktivitas, termasuk tempat tinggal, ibadah, bisnis, kegiatan sosial, budaya, dan tujuan khusus lainnya. Berdasarkan aspek hukum mengenai K3 (Kesehatan dan Keselamatan Kerja) yang telah ditetapkan pada Undang-Undang terkait Bangunan Gedung Nomor 28 Tahun 2002, diperoleh bahwa kenyamanan, keamanan, kesehatan, dan kemudahan merupakan aspek penting yang harus dimiliki oleh sebuah bangunan gedung.

Salah satu aspek yang krusial dalam menjaga keamanan dan keselamatan di dalam gedung adalah adanya sistem evakuasi darurat yang efektif. Menurut Peraturan Menteri Pekerjaan Umum Nomor 26/PRT/M/2008, pada saat proses evakuasi, penghuni atau pengguna bangunan harus memiliki waktu yang cukup untuk menyelamatkan diri dengan aman dan meninggalkan gedung. Dalam penelitian yang dilakukan oleh Andhika, Kasim, dan Hawibowo (2013), waktu optimal untuk proses evakuasi dipengaruhi oleh berbagai faktor, termasuk desain bangunan dan jalur evakuasi yang tersedia, jumlah orang yang perlu dievakuasi, serta elemen-elemen pendukung yang sesuai dengan standar terkait proses evakuasi. Sehingga, salah satu upaya untuk menjaga keamanan dan keselamatan pada bangunan gedung adalah adanya jalur evakuasi yang efektif untuk mengoptimalkan waktu pada proses evakuasi saat situasi darurat terjadi.

Masalah penentuan jalur evakuasi tercepat dapat dipandang sebagai masalah pencarian lintasan terpendek. Algoritma Dijkstra merupakan salah satu algoritma yang populer digunakan untuk mencari lintasan terpendek, termasuk perencanaan jalur evakuasi (Afandi dan Mayasari, 2021). Berdasarkan Patel dan Bagar (2014) juga Harahap dan Khairina (2017), Algoritma Dijkstra bekerja dengan cara mengidentifikasi dan memperbarui lintasan terpendek dari simpul awal ke setiap

simpul lain dalam graf berbobot dengan bobot *non-negatif*. Berdasarkan *mini review* yang dilakukan oleh Al Hakim, Satria, Arief, Pangestu, Jaenul, Hertin, dan Nugraha (2021). Al Hakim dkk. (2021) Algoritma Dijkstra dapat dimanfaatkan untuk berbagai keperluan seperti penyelesaian *biobjective shortest path* (BSP), penentuan jalur *multi-objective*, Graf Dijkstra, evakuasi darurat, permasalahan *fuzzy*, integrasi dengan fitur *location-based service*, dan distribusi rute optimal.

Menurut Handaka (2011), Algoritma Dijkstra mampu melakukan pencarian lintasan terpendek dalam waktu yang cepat dan ruang atau memorinya yang kecil jika dibandingkan dengan Algoritma Bellman-Ford dan Algoritma Floyd-Warshall. Selain itu, menurut Patel dan Bagar (2014), pada masalah pencarian lintasan terpendek Algoritma Dijkstra dapat memberikan hasil yang lebih cepat dan lebih cocok untuk masalah jaringan yang besar dibandingkan dengan Algoritma Bellman-Ford. Sehingga dipilih Algoritma Dijkstra berdasarkan pertimbangan kecepatan dan efisiensi yang telah terbukti pada penelitian sebelumnya dalam konteks pencarian jalur evakuasi terpendek.

Penentuan jalur evakuasi tercepat biasanya didasarkan oleh beberapa faktor, diantaranya adalah panjang jalan, lebar jalan, dan jumlah pengungsi. Karena penggunaan Algoritma Dijkstra memerlukan bobot tunggal, maka agar algoritma dapat digunakan pada penentuan jalur evakuasi diperlukan metode untuk mengonversi berbagai macam parameter jalur evakuasi menjadi sebuah nilai saja. Salah satu metode yang dapat digunakan untuk keperluan tersebut adalah logika *fuzzy*. Logika *fuzzy* memungkinkan untuk mengatasi situasi di mana parameter seperti panjang jalan, lebar jalan, dan jumlah pengungsi yang tidak memiliki kondisi pasti. Tidak memiliki kondisi pasti dalam hal ini merujuk pada situasi dimana nilai-nilai dari parameter tidak dapat ditentukan dengan pasti atau memiliki tingkat ketidakpastian. Sebagai contoh, misalkan parameter panjang jalan dibagi menjadi tiga kategori himpunan yakni pendek (jika panjang jalan  $< 5$  meter), sedang (jika  $5 \text{ meter} \leq \text{panjang jalan} \leq 8 \text{ meter}$ ), dan panjang (jika panjang jalan  $> 8$  meter). Selanjutnya apabila terdapat sebuah jalan dengan panjang 4,99 meter, maka dikatakan pendek. Sedangkan, apabila terdapat sebuah jalan dengan panjang 5 meter, maka dikatakan tidak pendek. Dari sini bisa dikatakan bahwa pemakaian himpunan pasti untuk menyatakan panjang jalan tidak selalu adil, karena perbedaan

kecil dalam nilai dapat menyebabkan perubahan kategori yang cukup signifikan. Himpunan *fuzzy* digunakan untuk mengantisipasi hal tersebut, dimana sebuah jalan dapat masuk dalam dua himpunan yang berbeda. Hal tersebut dikarenakan, dalam logika *fuzzy* parameter-parameter tersebut memiliki derajat keanggotaan dalam himpunan *fuzzy* yang menggambarkan tingkat ketidakpastian. Seberapa besar eksistensinya dalam himpunan tersebut dapat dilihat pada derajat keanggotaannya. Logika *fuzzy* ini bekerja dengan mengambil data masukan yang berupa nilai pasti dan mengubahnya menjadi bentuk kabur, mengevaluasi aturan-aturan *fuzzy*, menggabungkan hasilnya, dan kemudian mengubah hasil tersebut kembali menjadi nilai pasti yang dapat digunakan dalam pengambilan keputusan. Dengan demikian, untuk mengefisienkan pencarian lintasan terpendek dengan Algoritma Dijkstra diperlukan konsep logika *fuzzy* sebagai penentuan bobot sisi atau dapat disebut Algoritma *Fuzzy Dijkstra*.

Penggunaan Algoritma *Fuzzy Dijkstra* pernah diteliti oleh beberapa peneliti. Deng, Chen, Zhang, dan Mahadevan (2012) meneliti masalah jalur terpendek atau *Shortest Path Problem* dalam lingkungan yang tidak pasti. Ichsan, Yudaningtyas, Muslim (2012) mengoptimalkan jalur evakuasi di jalan dan Iskierka dan Lipiec (2019) mengoptimalkan jalur antar dua kota. Abdillah, Rakhmatsyah, Putrada (2019) menggunakan Algoritma *Fuzzy Dijkstra* dalam sistem pengangkutan sampah dan Nggufon, Rochmad, dan Mashuri (2019) meneliti masalah pencarian rute terbaik bagi pemadam kebakaran. Afandi dan Mayasari (2021) juga Mayasari dan Afandi (2021) penggunaan Algoritma *Fuzzy Dijkstra* untuk mengoptimalkan jalur evakuasi tsunami.

Penelitian ini bertujuan untuk mengoptimalkan jalur evakuasi pada gedung dengan menggunakan Algoritma *Fuzzy Dijkstra*. Selanjutnya algoritma tersebut akan diimplementasikan untuk menyelesaikan masalah penentuan jalur evakuasi di Gedung FPMIPA A UPI. Penelitian sebelumnya dilakukan oleh Ilham (2023) yang menggunakan model *Maximum Dynamic Flow Problem* (MDFD) dengan pendekatan *Temporally Repeated Flow* pada pencarian jalur evakuasi di Gedung FPMIPA A UPI. Berbeda dengan penelitian tersebut yang bertujuan untuk memaksimalkan jumlah pengungsi dalam waktu minimum dengan asumsi bahwa panjang dan lebar jalan adalah sama, penelitian ini bertujuan untuk meminimalkan

waktu evakuasi dengan memperhatikan panjang jalan, lebar jalan, dan jumlah pengungsi yang sesungguhnya sehingga diharapkan akan diperoleh jalur evakuasi yang lebih realistis dari penelitian sebelumnya. Hasil dari penelitian ini diharapkan dapat meningkatkan keamanan dan keselamatan penghuni gedung dalam situasi darurat.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, rumusan masalah pada penelitian ini adalah sebagai berikut:

1. Bagaimana model optimasi masalah penentuan jalur evakuasi yang optimal pada sebuah gedung?
2. Bagaimana menerapkan logika *fuzzy* dalam penentuan bobot dari jalur evakuasi?
3. Bagaimana implementasi Algoritma *Fuzzy* Dijkstra dalam penentuan jalur evakuasi di Gedung FPMIPA A UPI?

## 1.3 Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk mengkaji model optimasi berbasis *Fuzzy* Dijkstra dan penyelesaiannya, serta penerapannya untuk perencanaan jalur evakuasi di Gedung FPMIPA A UPI.

## 1.4 Manfaat Penelitian

Manfaat dari penelitian ini adalah:

1. Menambah referensi tentang Algoritma *Fuzzy* Dijkstra pada masalah optimasi rute evakuasi di gedung.
2. Memberikan rujukan terkait perencanaan jalur evakuasi di Gedung FPMIPA A UPI.