

**NASKAH PUBLIKASI**  
**PERANCANGAN JALUR EVAKUASI MENGGUNAKAN**  
**ALGORITMA DIJKSTRA**  
**(Studi kasus Universitas Teknologi Yogyakarta Kampus I)**

Program Studi Informatika

Disusun oleh  
**ILAM NOVITA SARI**  
**5150411156**

**PROGRAM STUDI INFORMATIKA**  
**FAKULTAS TEKNOLOGI INFORMASI DAN ELEKTRO**  
**UNIVERSITAS TEKNOLOGI YOGYAKARTA**  
**YOGYAKARTA**  
**2020**

PERANCANGAN JALUR EVAKUASI MENGGUNAKAN ALGORITMA DIJKSTRA

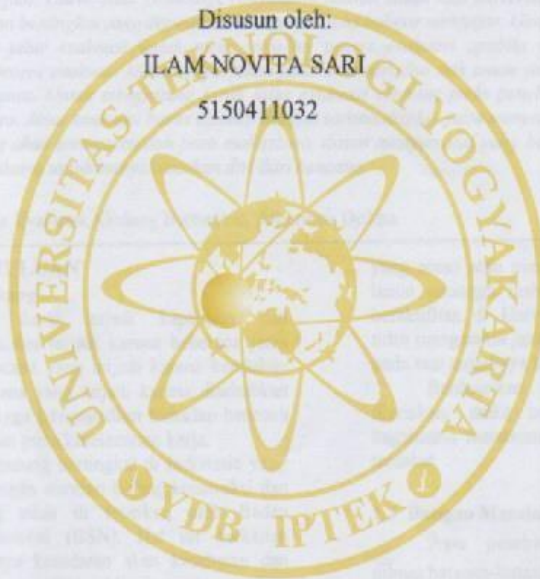
**NASKAH PUBLIKASI**

(Studi Kasus Universitas Teknologi Yogyakarta Kampus I)

**PERANCANGAN JALUR EVAKUASI MENGGUNAKAN ALGORITMA DIJKSTRA**

**(Studi kasus Universitas Teknologi Yogyakarta Kampus I)**

Disusun oleh:  
**ILAM NOVITA SARI**  
5150411032



Pembimbing

**Muhammad Fachrie.S.T.M.Cs.**

Tanggal: 30-09-2020

# PERANCANGAN JALUR EVAKUASI MENGGUNAKAN ALGORITMA DIJKSTRA (Studi Kasus Universitas Teknologi Yogyakarta Kampus I)

**Ilam Novita Sari<sup>1</sup>, Muhammad Fachrie<sup>2</sup>**

*Program Studi Informatika, Fakultas Teknologi Informasi dan Elektro*

*Universitas Teknologi Yogyakarta*

*Jl. Ringroad Utara Jombor Sleman Yogyakarta*

*E-mail : [ilamnov3@gmail.com](mailto:ilamnov3@gmail.com), [muhammad.fachrie@staff.utv.ac.id](mailto:muhammad.fachrie@staff.utv.ac.id)*

## ABSTRAK

*Bencana dapat terjadi kapan saja dan di mana saja. Bencana juga dapat terjadi dari berbagai faktor yaitu bencana alam seperti gempa bumi dan tanah longsor, ada pula bencana dari kesalahan pengguna seperti kebakaran dan lain sebagainya. Apalagi bencana yang dapat terjadi pada gedung bertingkat yang belum memenuhi standar bidang konstruksi dan bangunan maka sangat berbahaya apabila belum dilengkapi dengan jalur alternatif evakuasi pada saat bencana terjadi. Universitas Teknologi Yogyakarta adalah salah satu universitas di Yogyakarta yang memiliki struktur bangunan bertingkat yang digunakan sebagai fasilitas belajar mengajar. Gedung-gedung bertingkat tersebut belum memiliki jalur evakuasi untuk mempermudah proses evakuasi apabila terjadi suatu bencana. Untuk mempermudah proses evakuasi korban diperlukan jalur evakuasi dan titik aman yang mudah diakses oleh korban maupun penyelamat. Untuk menghitung jarak jalur evakuasi terdekat pada penelitian ini penulis menggunakan algoritma Dijkstra. Jalur evakuasi harus disertai dengan sarana display yaitu menyediakan rute jalur evakuasi yang mendukung yang akan mempermudah para mahasiswa, dosen maupun staf yang berada dilingkungan Universitas Teknologi Yogyakarta untuk menyelamatkan diri dari bencana.*

Kata kunci : Jalur Evakuasi, Gedung Bertingkat, Algoritma Dijkstra

## 1. PENDAHULUAN

### 1.1 Latar Belakang

Bencana dapat terjadi kapanpun dan dimanapun. Bencana terjadi karena beberapa jenis faktor yaitu bencana yang terjadi karena kesalahan teknis dan bencana yang terjadi karena disebabkan oleh alam, sehingga kewaspadaan terhadap bencana harus ditingkatkan guna keselamatan kerja.

Banyak gedung bertingkat di Indonesia yang belum sesuai dengan standart bidang konstruksi dan bangunan yang telah di tetapkan oleh Badan Standarisasi Nasional (BSN). Hal ini berkaitan dengan kurangnya kesadaran akan kesehatan dan keselamatan kerja (K3) dilingkungan tersebut.

Disetiap tempat kerja atau tempat umum seperti kantor, kampus, tempat perbelanjaan dan tempat yang memiliki gedung-gedung bertingkat maupun area gedung yang luas sebaiknya memiliki jalur evakuasi.

Jalur evakuasi digunakan untuk membantu para pengguna diarea gedung maupun ditempat yang rawan bencana untuk menyelamatkan diri ke tempat yang lebih aman atau titik kumpul. Jalur evakuasi biasanya berupa rute atau denah yang disertai dengan display.

Universitas Teknologi Yogyakarta adalah salah satu Universitas swasta yang ada di Yogyakarta yang memiliki tiga kampus yang letaknya berbeda-beda dan memiliki gedung-gedung bertingkat namun belum memiliki jalur evakuasi. Seperti kampus utama Universitas Teknologi Yogyakarta yang terletak di Jombor. Adapun permasalahan yang ada di Universitas Teknologi Yogyakarta tepatnya kampus utama yaitu tidak adanya prasarana yang menunjang jalur evakuasi, seperti display petunjuk jalur evakuasi

yang aman atau pun denah jalur evakuasi disetiap lantai sehingga mahasiswa ataupun petugas yang beraktifitas di Universitas Teknologi Yogyakarta tidak mengetahui jalur mana yang aman untuk dilalui pada saat terjadinya bencana.

Berdasarkan latar belakang yang telah diuraikan, maka pokok permasalahannya yaitu bagaimana merancang jalur evakuasi dengan jarak terdekat.

### 1.2 Batasan Masalah

Agar pembahasan lebih terarah, maka dibuat batasan-batasan pembahasan masalah yaitu:

- Perancangan jalur evakuasi ini digunakan untuk menentukan titik kumpul dan jalur evakuasi tercepat ketika terjadi bencana.
- Perancangan jalur evakuasi ini hanya di kampus 1 Universitas Teknologi Yogyakarta wilayah Sleman yaitu di gedung rektorat dan gedung perkuliahan.
- Perancangan jalur evakuasi menggunakan sistem sederhana yang dapat menampilkan jalur evakuasi.

### 1.3 Tujuan Penelitian

Tujuan dari penelitian ini yaitu membuat rancangan jalur evakuasi terdekat. Ketika terjadi bencana di Universitas Teknologi Yogyakarta kampus utama.

## 2. KAJIAN PUSTAKA DAN TEORI

### 2.1 Kajian Hasil Penelitian

Penelitian oleh Siregar, E. S. dkk., (2015), membahas tentang perancangan pembuatan jalur evakuasi yang disertai dengan display jalur evakuasi guna mempermudah pengguna untuk menemukan jalur evakuasi dengan menggunakan algoritma Dijkstra. Selain itu peneliti membahas tentang perancangan assembly point dengan melihat beberapa pertimbangan apa saja yang dibutuhkan. Hasil akhir dari penelitian tersebut memberikan informasi jalur terdekat untuk proses evakuasi saat terjadi suatu bencana yaitu melalui jalur depan gudang serbaguna.[1].

Penelitian oleh Kurniasari, L. dkk., (2018), membahas tentang pencarian gedung pada kompleks kampus terpadu Universitas Muhammadiyah Yogyakarta. Algoritma yang digunakan dalam penelitian ini sama dengan penelitian sebelumnya yaitu menggunakan algoritma Dijkstra.[2].

Peneliti oleh Iftadi, I. dkk., (2011), membahas tentang jalur evakuasi, sama seperti penelitian sebelumnya namun dalam penelitian ini berada di Rumah Sakit Umum Daerah Kebumen, yang membedakan penelitian ini dengan penelitian sebelumnya adalah algoritma yang dipakai oleh peneliti. Algoritma yang digunakan adalah algoritma flound-warshall. [3].

Penelitian oleh Rumondor, A. G. dkk., (2019), membahas tentang daerah rawan bencana di Sulawesi Utara. Sulawesi Utara terletak dekat dengan daerah sumber gempa bumi dan pembangkit tsunami baik darat maupun laut yang terbentuk akibat dari proses tektonik. Adapun sumber gempa di laut berasal dari penujaman sublempeng Sulawesi Utara yang terletak di utara Pulau Sulawesi, lempeng punggung mayu dan lempeng sangihe yang terletak di timur Sulawesi Utara. Sedangkan sumber gempa di laut juga merupakan sumber pembangkit tsunami, kota Manado misalnya. Dengan demikian peneliti merancang jalur terpendek evakuasi bencana gempa dan tsunami menggunakan algoritma Dijkstra agar dapat menunjang mitigasi bencana dalam bentuk penyediaan informasi jalur terpendek serta dapat mengetahui cara penyelamatan diri jika terjadi bencana gempa dan tsunami.[4].

Penelitian oleh Batmetan, J. R., (2016), membahas tentang pemilihan rute dalam evakuasi bencana dan memperhitungkan jalur mana saja yang dapat dijangkau oleh masyarakat ketika terjadi bencana. Penelitian tersebut masih sama dalam lokasi yang sama yaitu di Sulawesi Utara, yang membedakan penelitian ini dengan penelitian sebelumnya adalah objek dan algoritma yang digunakan. Penelitian ini mengambil objek gunung lokon sedangkan algoritma yang digunakan dalam penelitian ini adalah algoritma *Ant Colony Optimization* (ACO).[5].

### 2.2 Dasar Teori

#### 2.2.1 Jalur Evakuasi

Jalur Evakuasi adalah jalur yang menghubungkan ruang kelas atau titik kumpul dengan tempat evakuasi sementara dan jalur yang menghubungkan tempat evakuasi sementara dengan tempat evakuasi akhir. Jalur evakuasi bencana dapat berupa berbagai kelas, jalan, mulai dari jalan lingkungan, jalan lokal hingga jalan kolektor. Sebaiknya jalur ini dibuat dengan rute yang semaksimal mungkin menjauhi atau menghindari area yang mungkin dilalui atau dampak dari bencana secara langsung maupun tidak langsung. [ ].

#### 2.2.2 Titik Kumpul

Titik Kumpul adalah area terbuka di dekat pusat-pusat lingkungan gedung yang apabila terjadi bencana maka menjadi titik pertemuan yang hendak diungsikan ke tempat yang lebih aman, yakni tempat evakuasi sementara. Titik kumpul sebagian besar merupakan lapangan olah raga, sebagian kecil berupa area terbuka yang memungkinkan dilakukan kegiatan pengungsian seperti halaman kantor desa, sekolah atau tempat ibadah. Adapun kriteria titik kumpul yaitu sebagai berikut:

- Ketersediaan area atau ruang terbuka yang cukup memadai.
- Mudah diakses oleh korban bencana maupun penolong.
- Cukup terlindung dari jangkauan bahaya langsung atau tidak langsung dari bencana.
- Ketersediaan tempat naungan atau ruang sementara terutama bagi kelompok rentan seperti lansia, bayi, ibu hamil, dan difabel.
- Adanya kemudahan akses mobilisasi secara cepat.
- Ketersediaan sarana komunikasi memadai yang terhubung dengan struktur organisasi kedaruratan.
- Ketersediaan sarana pertolongan pertama atau *emergency kits*.
- Ketersediaan akses transportasi memadai seperti mobilisasi transportasi yang akan membawa ke tempat yang lebih aman secara cepat dan aman.

#### 2.2.3 Algoritma Dijkstra

Algoritma Dijkstra ditemukan oleh Edsger Dijkstra seorang ilmuwan komputer dari Belanda pada tahun 1959. Algoritma Dijkstra digunakan untuk pencarian grafik untuk memecahkan masalah jalur terpendek menuju sumber tunggal untuk grafik dengan tepi tidak negatif atau nonnegative. Sebagai contoh, jika simpul grafik mewakili kota dan biaya jalur tepi mewakili jarak mengemudi antara pasang kota yang terhubung dengan jalan langsung, algoritma Dijkstra bisa digunakan untuk menemukan rute terpendek antara satu kota dan kota lainnya (C. Yi-zhou, 2014). Metodologi yang digunakan untuk menemukan jalur terbaik dengan menggunakan algoritma Dijkstra adalah sebagai berikut:

- Tetapkan ke setiap node dengan nilai jarak dan tetapkan nol ke simpul awal dan ke tak terhingga untuk semua node lainnya.
- Tandai semua node sebagai simpul awal yang belum dikunjungi dan setel sebagai arus.
- Untuk node saat ini, pertimbangkan semua tetangga yang belum dikunjungi dan hitung nilainya jarak tentatif atau dari simpul awal. Misalnya, jika current node (A) mempunyai jarak 6, dan edge yang menghubungkannya dengan node lain (B) adalah 2, yaitu jarak ke B melalui A akan menjadi  $6 + 2 = 8$ . Jika jarak ini kurang dari sebelumnya tercatat jarak (tak terhingga di awal, nol untuk awal node), menimpa jarak.
- Saat mempertimbangkan semua tetangga simpul saat ini, tandai seperti yang dikunjungi. Sebuah node yang dikunjungi tidak akan diperiksa lagi, jarak yang direkam sekarang bersifat final dan minimal.
- Jika semua node telah dikunjungi, selesai. Jika tidak, setel node yang belum dikunjungi jarak terkecil (dari simpul awal) sebagai "node arus" berikutnya dan lanjutkan dari langkah ke 3.

Algoritma ini merupakan salah satu algoritma yang digunakan untuk menyelesaikan jalur terpendek dan tidak memiliki cost yang negatif. Strategi *greedy* yang digunakan dalam algoritma ini adalah setiap simpul dalam *graph* akan mencari nilai yang minimum. Pseudocode dari algoritma Dijkstra adalah sebagai berikut:

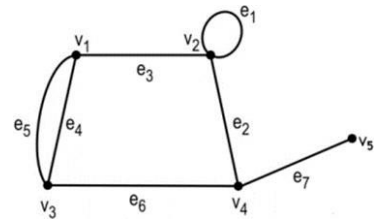
- Tentukan titik awal dan titik tujuan.
- Cari *node* yang berhubungan dengan node titik awal.
- Cari jarak terpendek dari node titik awal ke node-node yang berhubungan dengan node titik awal.
- Jika *node* tersebut bukan node tujuan maka ulangi langkah ke-2.

Jika *node* tersebut adalah *node* tujuan maka pencarian selesai.

## 2.2.4 Graf

Definisi dari suatu graf adalah himpunan benda benda yang disebut verteks atau node yang terhubung oleh sisi (edge atau arc). Biasanya graf digambarkan sebagai kumpulan titik-titik (melambangkan verteks) yang dihubungkan oleh garis-garis atau melambangkan sisi. Sebuah *graph* linier atau secara sederhana disebut *graph* yaitu  $G = (V, E)$  terdiri atas sekumpulan objek  $V = \{v_1, v_2, \dots\}$  yang disebut himpunan titik, dan sebuah himpunan lain  $E = \{e_1, e_2, \dots\}$  yang merupakan himpunan sisi sedemikian hingga tiap sisi  $e$  dikaitkan dengan suatu pasangan tak terurut  $k$  ( $v_i, v_j$ ). Titik  $v_i, v_j$  yang berkaitan dengan  $e_k$  disebut titiktitik ujung sisi. permodelan basis data relasional.

Cara merepresentasikan sebuah *graph* yang paling umum adalah berupa diagram. Dalam diagram tersebut, titik-titik dinyatakan sebagai noktah dan tiap sisi dinyatakan sebagai segmen garis yang menghubungkan tiap dua titik. Dalam sebuah *graph*, seperti terlihat pada gambar 2.1 dimungkinkan



**Gambar 2.1** Graf Sederhana

adanya suatu sisi yang dikaitkan dengan pasangan ( $v_1, v_2$ ).

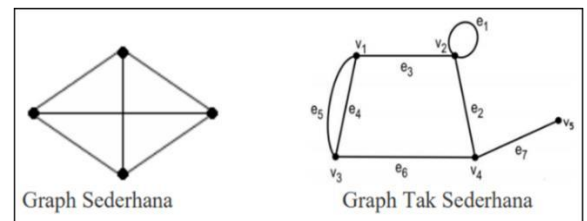
Sisi yang dua titik ujungnya sama disebut *loop*. Dalam graf pada Gambar 2.1, simbol  $e$  merupakan sebuah *loop*.

Dari contoh di atas juga perlu dicatat bahwa dalam sebuah graf dimungkinkan adanya lebih dari satu sisi yang dikaitkan dengan sepasang titik. Sebagai contoh  $e_4$  dan  $e_5$  pada graf di atas dikaitkan dengan pasangan titik ( $v_1, v_3$ ).

Pasangan sisi semacam ini disebut sisi-sisi paralel atau sejajar. Sebuah *graph* yang tidak memiliki *loop* dan sisi paralel disebut graf sederhana.

Dalam beberapa literatur teori *graph*, pembahasan hanya dibatasi pada graf sederhana, akan tetapi dalam banyak aplikasi teknik, penggunaan *loop* dan sisi paralel sangat diperlukan. Untuk membedakan antara *graph* yang memuat *loop* dan sisi paralel dengan *graph* yang tidak memuat kedua hal tersebut, sebagian penulis menggunakan istilah *graph* sederhana untuk *graph* yang tidak memuat *loop* dan sisi paralel, serta *graph* umum untuk yang lainnya.

Dalam sebuah *graph* mungkin terdapat dua sisi atau lebih yang menghubungkan dua titik yang berlainan. Kedua sisi tersebut dinamakan sisi rangkap atau sisi ganda. *Graph* yang mengandung *loop* atau sisi rangkap dinamakan *graph* ganda.



**Gambar 2.2** Graf Sederhana dan Tidak Sederhana

Dalam menggambar sebuah *graph*, bentuk sisi dapat berupa garis lurus atau garis lengkung. Demikian pula ukurannya, dalam penggambaran sebuah *graph* tidaklah diperhatikan. Hal yang penting untuk diperhatikan dalam sebuah *graph* adalah insidensi antara titik-titik dengan sisi-sisinya.

## 3. METODOLOGI PENELITIAN

### 3.1 Bahan atau Data

#### 3.1.1 Data Yang Diperoleh

Dalam pengambilan data ini difokuskan pada gedung rektorat dan gedung perkuliahan. Bagian gedung rektorat terdiri dari 4 lantai yaitu basement gedung rektorat, gedung rektorat lantai 1, gedung rektorat lantai 2, gedung rektorat lantai 3. Berikut adalah data tabel gedung rektorat:

**Tabel 3.2** Tabel Gedung Rektorat

No	Gedung dan Lantai	Ruang dan Kelas	KM	Jalur dengan Tagga
1	Basement rektorat	11	2	4
2	Rektorat lantai 1	11	2	7
3	Rektorat lantai 2	16	2	4
4	Rektorat lantai 3	12	3	4
Total		50	9	19

Seperti pada tabel 3.1 menjelaskan bahwa gedung rektorat yang terdiri dari 4 lantai memiliki total ruang dan kelas sebanyak 50 ruang dan kelas, 9 kamar mandi, dan 19 jalur yang dilalui dengan tangga. Setelah gedung rektorat, data diperoleh dari gedung perkuliahan kampus satu Universitas Teknologi Yogyakarta. Sama dengan gedung rektorat, gedung perkuliahan sendiri juga memiliki 4 lantai yaitu basement gedung perkuliahan, gedung perkuliahan lantai 1, gedung perkuliahan lantai 2, gedung perkuliahan lantai 3. Data tabel gedung perkuliahan dijelaskan dalam tabel 3.2 adalah sebagai berikut:

**Tabel 3.1** Tabel Gedung Perkuliahan

No	Gedung dan Lantai	Ruang dan Kelas	KM	Jalur Tagga
1	Basement perkuliahan	17	4	6
2	Gedung rektorat lantai 1	19	4	7
3	Gedung rektorat lantai 2	24	4	4
4	Gedung rektorat lantai 3	24	4	4
Total		84	16	21

### 3.1.2 Prosedur Pengumpulan Data

Dalam penelitian kali ini objek penelitian yang penulis gunakan adalah gedung rektorat dan gedung perkuliahan Universitas Teknologi Yogyakarta kampus I. Data diperoleh dengan cara meminta ijin kepada pihak kampus untuk mengambil data setelah disetujui penulis mengambil data langsung kelokasi dengan cara mengukur jarak antar kelas dan ruang serta menentukan *node-node* seperti persimpangan dan tangga.

### 3.1.3 Tahap Penelitian

Tahap penelitian ini terdiri dari beberapa tahapan yaitu:

#### a. Identifikasi awal

Identifikasi awal ini dilakukan dengan melihat struktur gedung seperti jumlah ruangan, jalur yang dilalui dengan melewati tangga. Universitas Teknologi Yogyakarta tepatnya di gedung rektorat dan gedung perkuliahan kampus satu memiliki jumlah ruangan sebanyak 134 yang terdiri dari ruangan dan kelas, 25 kamar mandi dan 40 jalur yang dilalui dengan tangga.

#### b. Observasi dan pengukuran jalur evakuasi

Observasi jalur evakuasi ini dilakukan setelah mengetahui struktur gedung yang ada di kampus satu Universitas Teknologi Yogyakarta. Observasi dilakukan dengan cara mengukur jalur yang dilalui dengan menggunakan alat ukur roll meter.

#### c. Perancangan jalur evakuasi

Perancangan jalur evakuasi ini dilakukan setelah data-data diperoleh setelah observasi dilakukan. Dalam perancangan ini dibutuhkan algoritma untuk menghitung jarak jalur tercepat ketika evakuasi dilakukan. Algoritma yang digunakan dalam penelitian ini adalah algoritma Dijkstra.

#### d. Implementasi

Pada tahap ini, data yang telah diperoleh dari kampus satu Universitas Teknologi Yogyakarta diolah menggunakan algoritma Dijkstra kemudian diimplementasikan ke dalam program sederhana yang telah dibuat.

## 4. ANALISIS DAN PERANCANGAN SISTEM

### 4.1 Analisis Sistem yang Diusulkan

Analisis yang diusulkan untuk perancangan jalur evakuasi di Universitas Teknologi Yogyakarta meliputi:

#### a. Analisis fungsional

Analisis fungsional perancangan jalur evakuasi terdiri dari itu yang terdapat pada program sederhana perancangan jalur evakuasi yaitu dapat mencari jalur evakuasi terpendek di gedung rektorat dan gedung perkuliahan kampus 1 Universitas Teknologi Yogyakarta serta kemampuan yang dimiliki sistem sederhana perancangan jalur evakuasi adalah dapat menampilkan jalur evakuasi terpendek dan titik kumpul yang aman pada saat terjadi bencana yang disajikan berupa gambar denah dan keterangan sehingga mempermudah pembaca untuk mengetahui jalur mana saja yang aman untuk dilewati saat berada di gedung rektorat maupun gedung perkuliahan kampus 1 Universitas Teknologi Yogyakarta.

#### b. Analisis nonfungsional

Analisis nonfungsional sendiri berupa software seperti Google Chrome, XAMPP,



MySQL, Sublime dan hardware yang digunakan untuk mendukung perancangan jalur evakuasi.

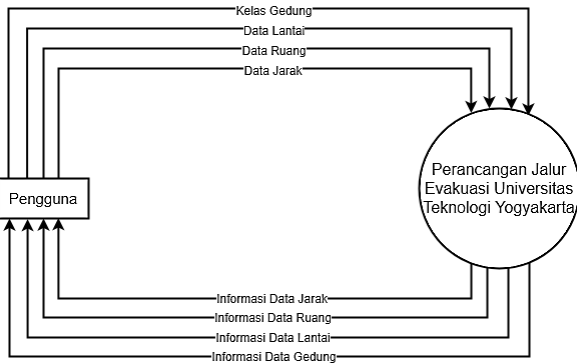
## 4.2 Rancangan Sistem

### 4.2.1 Desain Logik

Untuk lebih memahami perancangan sistem yang akan dibuat, maka diperlukan suatu gambaran sistem dan arus data yang digunakan mulai dari proses input hingga proses output yang akan dibutuhkan oleh pengguna. Berikut ini adalah gambaran sistem dan arus data yang dimodelkan kedalam diagram konteks, diagram jenjang, perancangan graf, DAD level 0 hingga relasi tabel.

### 4.2.2 Diagram Konteks

Diagram konteks merupakan diagram yang memperlihatkan aplikasi sebagai bentuk suatu proses yang terjadi. Tujuannya yaitu untuk memberikan gambaran umum yang terjadi pada perancangan sistem. Aliran diagram memodelkan masukan ke sistem dan keluaran dari sistem seperti pada Gambar

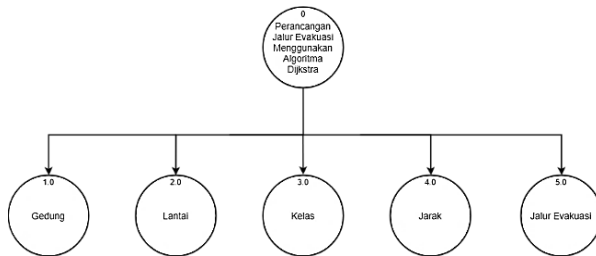


4.1.

Gambar 4.1 Diagram Konteks

### 4.2.3 Diagram Jenjang

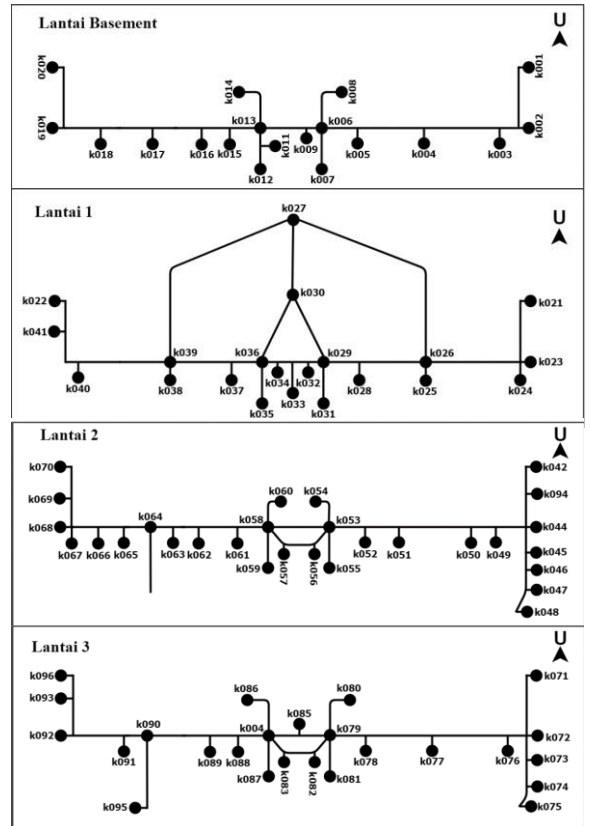
Diagram Jenjang merupakan penjelasan sistem yang dapat menampilkan seluruh proses yang terdapat. Pada Gambar 4.4 menjelaskan rangkaian pada perancangan system sederhana jalur evakuasi yang dapat dilihat pada Gambar 4.4.



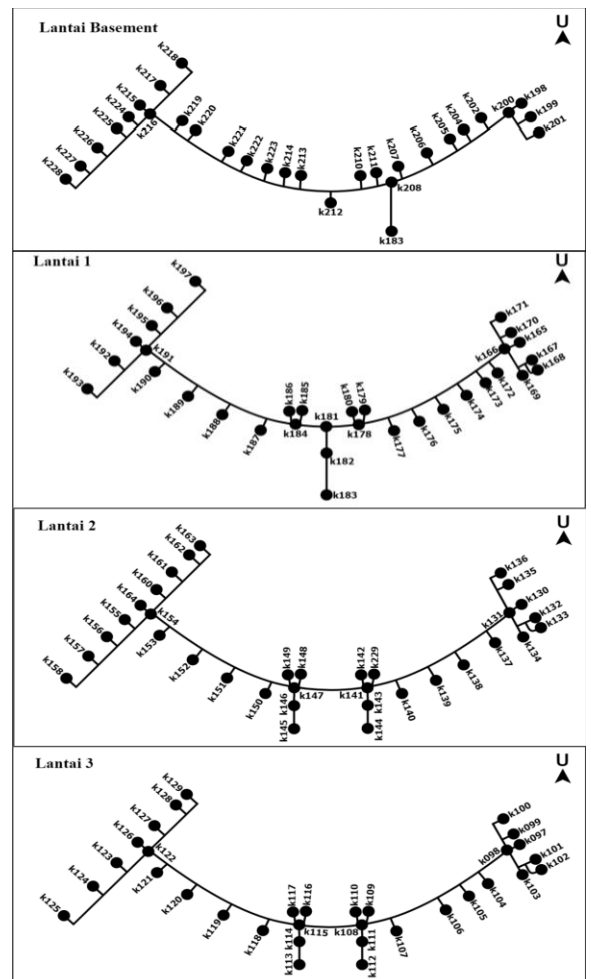
Gambar 4.1 Diagram Jenjang

### 4.2.5 Rancangan Graf

Rancangan Graf menggunakan node-node yang telah dirancang dari gedung rektoran dan gedung perkuliahan lantai basement sampai lantai 3. Berikut adalah gambar rancangan graf gedung rektorat:



Gambar 4.3 Rancangan Graf Gedung Rektorat

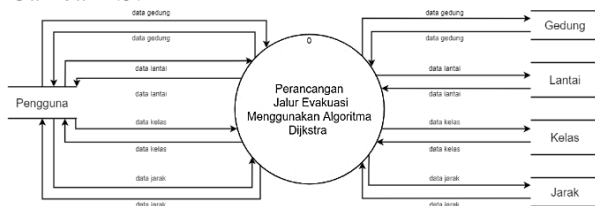


Gambar 4.4 Rancangan Graf Gedung Perkuliahan

Rancangan graf gedung rektorat dan gedung perkuliahan menggambarkan seluruh *node-node* yang telah dibuat oleh penulis sehingga mempermudah dalam perancangan jalur evakuasi yang akan dibuat. Rancangan graf sendiri dibuat dari lantai basement gedung rektorat sampai lantai 3 gedung rektorat yang dapat dilihat pada gambar 4.3 dan gedung perkuliahan yang dapat dilihat pada gambar 4.4.

#### 4.2.6 DAD (Diagram Alir Data) Level 0

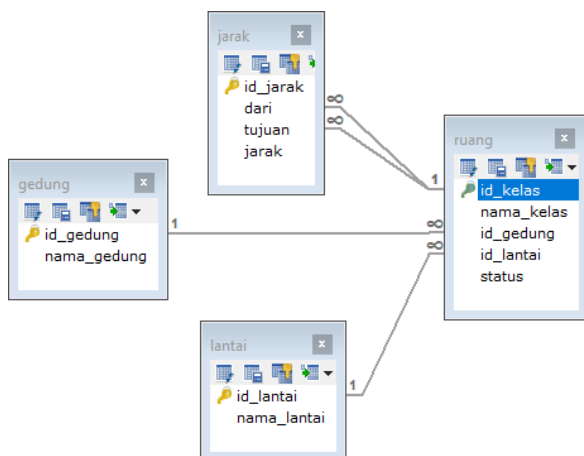
Dalam DAD level 0 dijelaskan mengenai proses data gedung, data lantai, data kelas dan data jarak diolah sehingga menghasilkan informasi data gedung, informasi data lantai, informasi data kelas, dan informasi data jarak yang dapat digunakan untuk perancangan jalur evakuasi dapat dilihat pada Gambar 4.5.



Gambar 4.5 DAD Level 0

#### 4.2.7 Relasi Antar Tabel

Relasi antar tabel menggambarkan hubungan antar tabel dengan adanya *primary key* dan *foreign key* yang terdapat pada setiap tabel dan kemudian digunakan untuk merelasikan antar tabel yang memiliki kaitan pada input dan proses. Diagram relasi antar tabel digambarkan pada gambar seperti dibawah ini:



Gambar 4.6 Relasi Antar Tabel.

## 5. IMPLEMENTASI DAN HASIL

### 5.1 Implementasi

Proses perancangan sistem sederhana yang dilakukan pada bab sebelumnya akan dijelaskan pada bab ini. Implementasi bertujuan untuk menerjemahkan keperluan perangkat lunak ke dalam bentuk sebenarnya yang dimengerti oleh komputer atau dengan kata lain tahap implementasi ini merupakan tahapan lanjutan dari tahap perancangan yang sudah dilakukan.

### 5.2 Hasil

Hasil merupakan tahapan implementasi mengenai hasil uji coba sistem yang dimasukan. Hasil uji coba ini akan ditampilkan menggunakan screenshot pada layar monitor.

#### 5.2.1 Halaman Home

Halaman *home* merupakan halaman utama dari program sederhana perancangan jalur evakuasi menggunakan algoritma Dijkstra. Adapun tampilan *home* sebagai berikut:



Gambar 5.1 Halaman Home

Halaman pada gambar 5.2 ini berisi button menu home, profil dan jalur evakuasi yang akan mrujuk ke halaman lain apabila salah satu button. Pada halama ini juga tersaji foto kampus 1 Universitas Teknologi Yogyakarta.

#### 5.2.2 Halaman Profil

Halaman profil berisi tentang informasi dari Universitas Teknologi Yogyakarta seperti alamat, e-mail, sejarah dingkat dan lain-lain. Seperti pada gambar dibawah ini.



Gambar 5.2 Halaman Profil

#### 5.2.3 Halaman Jalur Evakuasi Gedung Rektorat

Pada halaman halaman jalur evakuasi gedung rektorat ini digunakan untuk melakukan proses inputan yaitu memilih lokasi serta kelas yang sudah tersedia dalam pilihan. Kemudian ada *button Oke* yang akan mengarah ke halaman selanjutnya. Seperti yang terlihat pada gambar 5.3.



Gambar 5.3 Halaman Profil



Dari gambar 5.3 bagian pilih lokasi anda di input lantai “Basement” kemudian bagian pilih kelas di input “R. Kelas C02” kemudian klik *button Oke*. Maka akan muncul hasil dibawah ini.



**Gambar 5.4** Hasil Pengujian Gedung Rektorat

Dari hasil pengujian jalur evakuasi basement gedung rektorat dapat diketahui bahwa rute jalur evakuasi dari ruang kelas c02 ke titik kumpul adalah melewati kode rute:

k004→k005→k006→k007→k031→k029→k030→k027 adalah rute tercepat untuk dilewati menuju titik kumpul pada saat terjadi bencana.

## 6. PENUTUP

### 6.1 Kesimpulan

Berdasarkan hasil penelitian dan uji coba hasil yang dilakukan penulis pada sistem sederhana perancangan jalur evakuasi Universitas Teknologi Yogyakarta kampus 1, penulis dapat mengambil kesimpulan yaitu: sistem sederhana ini dapat memberikan informasi rute dan jalur mana yang tercepat untuk dilewati menuju titik kumpul pada saat terjadi bencana.

### 6.2 Saran

Berdasarkan hasil dari melakukan penelitian ini, penulis menyadari banyak kekurangan dari sistem yang sudah dibuat, beberapa hal sebagai berikut:

- Sistem sederhana yang dimiliki belum dapat memunculkan titik awal dan titik kumpul digambar rute gedung yang telah dibuat.
- Dapat dikembangkan dengan membuat tampilan rute dengan cara diberi warna yang berbeda dari warna jalur yang dibuat.
- Diharapkan sistem seperti ini ada di dalam sistem kemahasiswaan kampus agar pengguna dapat mengakses dan mengetahui rute evakuasi.

## DAFTAR PUSTAKA

- [1] Batmetan, J.R. (2016), *Algoritma Ant Colony Optimization (ACO) Untuk Pemilihan Jalur Tercepat Evakuasi Bencana Gunung Lokon Sulawesi Utara*, *Jurnal Teknologi Informasi-Aiti*, 14(1), 31–48.
- [2] Iftadi, I., Jauhari, W.A. and Nugroho, B. (2011), *Perancangan Peta Evakuasi Menggunakan Algoritma Floyd-Warshall Untuk Penentuan Lintasan Terpendek (Studi Kasus: Rumah Sakit Umum Daerah Kabupaten Kebumen)*,

*PERFORMA : Media Ilmiah Teknik Industri*, 10(2), 95–104.

- [3] Kurniasari, L., Mayadi and Kusri (2018), *Pencarian Rute Gedung Menggunakan Algoritma Dijkstra*, *Seminar Nasional Teknologi Informasi Dan Multimedia 2018*, 43–48.
- [4] Rumondor, A.G., Sentinuwo, S.R. and Sambul, A.M. (2019), *Perancangan Jalur Terpendek Evakuasi Bencana Di Kawasan Boulevard Manado Menggunakan Algoritma Dijkstra*, *Jurnal Teknik Informatika*, 14(2), 261–268.
- [5] Siregar, E.S., Yuniar and Desrianty, A. (2015), *Usulan Perancangan Jalur Evakuasi Menggunakan Algoritma Dijkstra (Studi Kasus Gedung 21 Itenas)*, *Jurnal Online Institut Teknologi Nasiona*.

