

## IMPLEMENTASI ALGORITMA *DIJKSTRA* UNTUK PENENTUAN JALUR TERPENDEK PADA APLIKASI EVAKUASI BENCANA UNTUK PENYANDANG DISABILITAS

Bahraen Folaimam<sup>1</sup>, Rosihan<sup>2</sup>, Abdul Mubarak<sup>3</sup>

Program Studi Teknik Informatika, Fakultas Teknik, Universitas Khairun

Jl. Jati Metro, Kota Ternate Selatan

E-mail : bahraenfolaimam88@gmail.com<sup>1</sup>, rosihan.unkhair@outlook.com<sup>2</sup>,

abdulmubarak029@gmail.com<sup>3</sup>

*Abstract -- Persons with disabilities are people who have physical and socio-economic limitations so that they are very vulnerable when a disaster occurs, for this reason an application is needed that can provide location information and access roads to the evacuation team so that the evacuation process can be done quickly. One algorithm that can show the shortest path is the dijkstra algorithm. The workings of the dijkstra algorithm use a greedy strategy, where in each step the side with the smallest weight that connects a selected node with another node has not been selected. The purpose of this study is to implement the dijkstra algorithm for determining the shortest path in disaster evacuation applications and web-based dijkstra calculation system. The system is created using the Java programming language and PHP with prototype as a method of system development. The application is made using GPS technology to show the user's location. From the results of the comparison between disaster evacuation applications and web-based systems that both show the same shortest route only the distance weights are different. This is because the web-based system only uses the coordinates available in the database, while the disaster evacuation application is taken directly from the maps server. Thus the application of the dijkstra algorithm on the system will have a precise and accurate path if more specified coordinates.*

*Keyword : Dijkstra, Prototype, GPS, Maps Server*

*Abstrak -- Penyandang disabilitas merupakan orang yang memiliki keterbatasan fisik maupun sosial ekonomi sehingga sangat rentan ketika terjadi bencana, untuk itu dibutuhkan sebuah aplikasi yang dapat memberikan informasi lokasi dan akses jalan ke tim evakuasi agar proses evakuasi bisa dilakukan dengan cepat. Salah satu algoritma yang dapat menunjukkan jalur terpendek adalah algoritma *dijkstra*. Cara kerja algoritma *dijkstra* memakai strategi *greedy*, dimana pada setiap langkah dipilih sisi dengan bobot terkecil yang menghubungkan sebuah simpul yang sudah terpilih dengan simpul lain yang belum terpilih. Tujuan dari penelitian ini adalah mengimplementasikan algoritma *dijkstra* untuk penentuan jalur terpendek pada aplikasi evakuasi bencana dan sistem perhitungan *dijkstra* berbasis *web*. Sistem dibuat dengan menggunakan bahasa pemrograman *java* dan PHP dengan *prototype* sebagai metode pengembangan sistemnya. Aplikasi dibuat dengan memanfaatkan teknologi GPS untuk menunjukkan lokasi *user*. Dari hasil perbandingan antara aplikasi evakuasi bencana dan sistem berbasis *web* bahwa keduanya menunjukkan *route* terpendek yang sama hanya saja bobot jarak yang di tampilkan berbeda. Hal ini di karenakan pada sistem berbasis *web* hanya menggunakan koordinat yang tersedia di *database*, sedangkan pada aplikasi evakuasi bencana koordinatnya diambil langsung dari *maps server*. Dengan demikian penerapan algoritma *dijkstra* pada sistem akan memiliki jalur yang tepat dan akurat jika koordinat yang ditentukan semakin banyak.*

*Kata Kunci : Dijkstra, Prototype, GPS, Maps Server*

### I. PENDAHULUAN

Indonesia merupakan salah satu negara dengan tingkat kerawanan bencana yang tinggi dengan resiko korban yang tinggi pula [1]. Salah satu contoh daerah yang memiliki tingkat kerawanan bencana yang tinggi yaitu Ternate. Ternate memiliki gunung berapi Gamalama yang termasuk dalam kategori aktif, bencana yang ditimbulkan seperti gunung meletus, gempa, dan banjir lahar dingin [2].

Adanya bencana alam menyebabkan banyak korban baik luka maupun meninggal. Korban bisa siapa saja termasuk penyandang disabilitas.

Penyandang disabilitas sangat rentan saat terjadi bencana dikarenakan keterbatasan fisik maupun sosial ekonomi. Ketika bencana terjadi sering penyandang disabilitas tidak tanggap terhadap situasi darurat yang sedang terjadi. Perlu adanya pertolongan dari pihak yang berkompeten untuk membantu penyandang disabilitas dalam usaha evakuasi. Muncul permasalahan ketika penyandang disabilitas harus memberi informasi secara cepat dan akurat mengenai posisi mereka ke tim evakuasi karena keterbatasan yang dimiliki. Tidak menutup kemungkinan karena terlambatnya pemberian informasi tersebut dapat mengakibatkan korban luka atau sampai meninggal dunia.

## **Implementasi Algoritma Dijkstra Untuk Penentuan Jalur Terpendek Pada Aplikasi Evakuasi Bencana Untuk Penyandang Disabilitas**

Untuk itu dibutuhkan suatu teknologi yang dapat memberikan informasi lokasi serta akses jalan yang harus di lalui pihak yang melakukan evakuasi sehingga kegiatan evakuasi dapat dilakukan secara cepat. Salah satu teknologi yang bisa digunakan adalah *Location Based Service* (LBS) yang mampu menyediakan layanan berbasis lokasi kepada pengguna *mobile smartphone* yang menerapkan sistem *Global Positioning Satellite* (GPS). Melalui teknologi LBS ini, maka perlu dikembangkan sebuah aplikasi yang mampu menyediakan informasi jalur evakuasi dengan menunjukkan *route* terpendek.

Untuk menentukan *route* terpendek diterapkan algoritma *Dijkstra*. Dari beberapa penelitian yang telah dilakukan sebelumnya terkait perbandingan algoritma penentuan jalur terpendek, algoritma *Dijkstra* memiliki kecepatan yang lebih cepat dalam memproses dari pada algoritma yang lain seperti algoritma *A\** (*A Star*), Algoritma Semut, dan Algoritma *Bellman-Ford* [3].

Berdasarkan latar belakang tersebut penulis melakukan sebuah penelitian dengan judul Implementasi Algoritma *Dijkstra* Untuk Penentuan Jalur Terpendek pada Aplikasi Evakuasi Bencana untuk Penyandang Disabilitas. Adanya aplikasi ini diharapkan dapat memberikan informasi lokasi penyandang disabilitas dengan teknologi GPS serta informasi akses jalan atau jalur terpendek menggunakan algoritma *Dijkstra*.

## **II. TINJAUAN PUSTAKA**

### **2.1 Android**

*Android* merupakan sistem operasi yang *open source* yang dibangun di atas *Kernel Linux* 2.6. Salah satu keuntungan dari sistem operasi *open source* ini yaitu aplikasi pihak ketiga (*third-party application*) dapat mengakses seluruh *resource* yang dimiliki oleh *smartphone* tersebut, tanpa membedakannya dengan aplikasi inti dari *smartphone* [4].

Pengembangan aplikasi *android* menggunakan bahasa pemrograman *java* yang mana konsep-konsep pemrograman *java* berhubungan dengan *Object Oriented Programming* (OOP). Selain itu pula dalam pengembangan aplikasi *android* membutuhkan *Software Development Kit* (SDK) yang disediakan *android*, SDK ini memberi jalan bagi *programmer* untuk mengakses *Application Programming Interface* (API) pada *android* [5]. Salah satu editor yang digunakan untuk pembuatan aplikasi berbasis *android* yaitu *android studio*.

### **2.2 Disabilitas**

Istilah disabilitas secara terus menerus berkembang, baik itu pandangan maupun pendekatan pengembangannya. Terminologi yang digunakan juga berbeda pada tiap Negara dan wilayah. Ekspresi yang tidak sesuai atau bahkan menghina harus di hindari, meskipun jika hal tersebut masih digunakan pada instansi pemerintah. Disabilitas diartikan sebagai hasil dari interaksi antara orang dengan malfungsi organ tubuh, sikap, dan batasan lingkungan yang

menghalangi mereka untuk secara penuh dan efektif berpartisipasi dalam masyarakat setara dengan orang lain. Malfungsi organ tubuh atau *impairment* adalah masalah pada fungsi tubuh atau struktur yang secara signifikan terganggu atau bahkan hilang, misalnya fungsi tubuh, fungsi mental, fungsi sensor dan rasa sakit, fungsi suara dan kemampuan berbicara, fungsi *kardiovaskular*, amputasi, ataupun penyakit-penyakit lainnya [1].

Berdasarkan hasil pengambilan data penyandang disabilitas yang dilakukan di Dinas Sosial Kota Ternate terdiri dari 840 orang, sedangkan di sekolah SLB Negeri Sasa Ternate dan sekolah SLB YPAC Ternate terdiri 208 orang.

### **2.3 Google Maps**

*Google Maps* adalah layanan oleh *Google* yang menawarkan teknologi pemetaan yang *userfriendly*. Layanan ini dapat diakses melalui situs <http://maps.google.com> atau <http://maps.google.co.id> untuk *Google Maps* Indonesia. *Google Maps* juga menawarkan peta yang dapat diseret dan gambar satelit untuk seluruh dunia, serta menawarkan *route* perjalanan. *Google Maps* dibuat dengan menggunakan kombinasi dari gambar peta, *database*, serta objek-objek interaktif yang dibuat dengan bahasa pemrograman HTML, *Javascript*, dan AJAX, serta beberapa bahasa pemrograman lainnya. Seluruh citra yang ada diintegrasikan ke dalam suatu *database* pada *google server* [6].

### **2.4 Global Positioning System (GPS)**

*Global Positioning System* (GPS) merupakan sebuah alat atau sistem yang dapat digunakan untuk menginformasikan penggunaannya dimana lokasinya berada (secara global) di permukaan bumi yang berbasiskan satelit. Data di kirim dari satelit berupa sinyal radio dengan data digital. Dimanapun pengguna tersebut berada, maka GPS bisa membantu menunjukkan arah. Awalnya GPS hanya digunakan hanya untuk kepentingan militer, tapi pada tahun 1980-an dapat digunakan untuk kepentingan sipil. GPS dapat di gunakan di manapun juga dalam 24 jam. Posisi unit GPS akan ditentukan berdasarkan titik-titik koordinat *latitude* dan *longitude* [7].

### **2.5 Algoritma Dijkstra**

Algoritma ini dinamakan sesuai dengan nama penemunya, yaitu seorang ilmuwan komputer berkebangsaan Belanda yang bernama *Edsger Dijkstra*. Algoritma *Dijkstra* termasuk kedalam pembahasan teori *graf* pada matematika diskrit yang berhubungan dengan *graf* berbobot dan lintasan terpendek (*shortest path*). Algoritma ini digunakan untuk mencari lintasan terpendek pada sebuah *graf* berarah. Cara kerja algoritma *Dijkstra* memakai strategi *greedy*, dimana pada setiap langkah dipilih sisi dengan bobot terkecil yang menghubungkan sebuah simpul yang sudah terpilih dengan simpul lain yang belum terpilih [8].

Algoritma yang ditemukan oleh *Dijkstra* untuk mencari *path* terpendek merupakan algoritma yang

**Implementasi Algoritma Dijkstra Untuk Penentuan Jalur Terpendek Pada Aplikasi Evakuasi Bencana Untuk Penyandang Disabilitas**

lebih efisien dibandingkan algoritma *Warshall*, meskipun implementasinya juga lebih sukar. Misalkan  $G$  adalah *graf* berarah berlabel dengan titik-titik  $V(G) = \{v_1, v_2, \dots, v_n\}$  dan *path* terpendek yang dicari adalah dari  $v_1$  ke  $v_n$ . Algoritma *Dijkstra* dimulai dari titik  $v_1$ . Dalam iterasinya, algoritma akan mencari satu titik yang jumlah bobotnya dari titik 1 terkecil. Titik-titik yang terpilih dipisahkan dan titik-titik tersebut tidak diperhatikan lagi dalam iterasi berikutnya. Misalkan:

- $V(G) =$  = Himpunan titik-titik  $\in V(G)$  yang sudah terpilih dalam jalur *path* terpendek.
- $D(j)$  = Jumlah bobot *path* terkecil dari  $v_1$  ke  $v_j$ .
- $w(i,j)$  = Bobot garis dari titik  $v_i$  ke  $v_j$ .
- $w^*(1,j)$  = Jumlah bobot *path* terkecil dari  $v_1$  ke  $v_j$

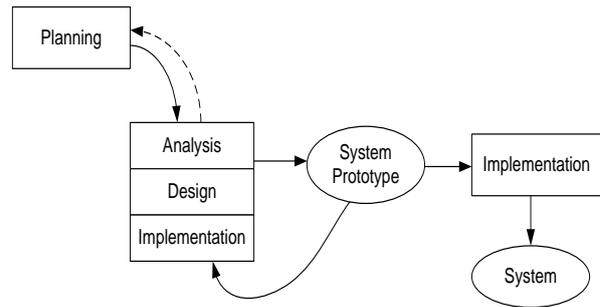
Menurut algoritma di atas, *path* terpendek dari titik  $v_1$  ke  $v_n$  adalah melalui titik-titik dalam  $L$  secara berurutan, dan jumlah bobot *path* terkecilnya adalah  $D(n)$ . Untuk menghitung *weight* (bobot) nilai minimum menggunakan persamaan berikut [8].  
 $Weight(w) = \min(DstValue, MarkValue + EdgeValue) \dots \dots \dots (1)$   
 Dimana :  $DstValue$  = Nilai dalam *node* tujuan  
 $MarkValue$  = Nilai dalam *node* awal  
 $EdgeValue$  = Bobot dari sisi yang menghubungkan *node*

**2.6 Metode Prototype**

*Prototype* merupakan metodologi pengembangan *software* yang menitik beratkan pada pendekatan aspek desain, fungsi dan *user-interface*. *Developer* dan *user* fokus pada *user-interface* dan bersama-sama mendefinisikan spesifikasi, fungsi, desain dan bagaimana *software* bekerja. *Developer* dan *user* bertemu dan melakukan komunikasi dan menentukan tujuan umum, kebutuhan yang diketahui dan gambaran bagian-bagian yang akan dibutuhkan. [9].

Cakupan aktivitas dari *prototyping* model terdiri dari:

1. Mendefinisikan objektif secara keseluruhan dan mengidentifikasi kebutuhan yang sudah diketahui.
2. Melakukan perancangan secara cepat sebagai dasar untuk membuat *prototype*.
3. Menguji coba dan mengevaluasi *prototype* dan kemudian melakukan penambahan dan perbaikan-perbaikan terhadap *prototype* yang sudah dibuat.



Gambar 1. *Prototyping Model* [9]

**III. METODE PENELITIAN**

Pada penelitian ini metode pengembangan perangkat lunak yang digunakan adalah metode *prototype* yaitu sebagai berikut.

**3.1 Analisis**

Tahapan pengumpulan data yang dilakukan adalah sebagai berikut.

**1.1 Observasi**

Observasi yaitu pengamatan langsung ke beberapa lokasi penyandang disabilitas untuk melihat bagaimana kondisi penyandang disabilitas serta untuk mengetahui langsung seperti apa aktivitas dari penyandang disabilitas.

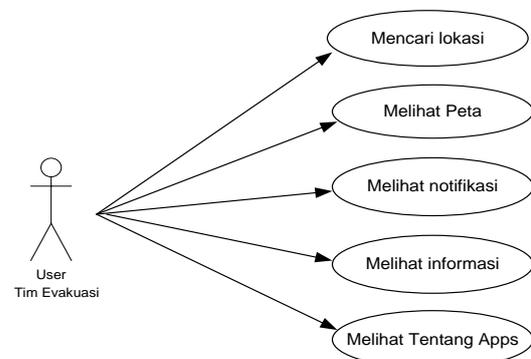
**1.2 Wawancara**

Wawancara (*Interview*), dengan cara tanya jawab dengan Badan Penanggulangan Bencana Daerah (BPBD) Ternate dan Komunitas Relawan Bencana Maluku Utara tentang apa saja kendala yang dihadapi saat proses penanganan yang dilakukan terhadap para penyandang disabilitas ketika akan atau saat terjadi bencana.

**3.2 Desain Sistem**

**1. Diagram UML**

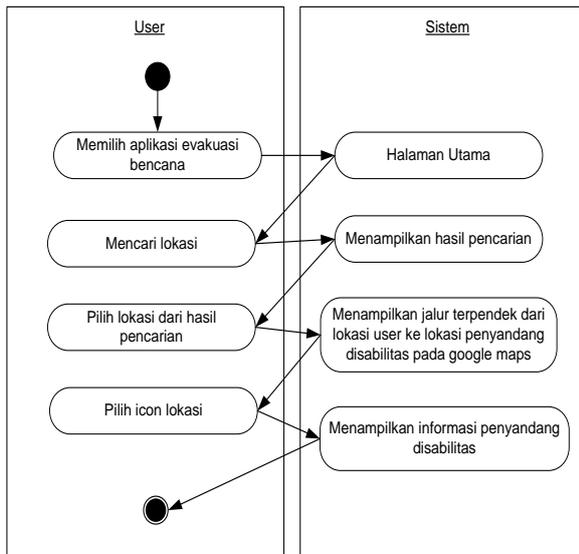
Pada tahap ini akan dilakukan pemodelan sistem untuk keperluan proses dan dokumentasi, perancangan UML seperti diagram *use case* yaitu *use case* untuk tim evakuasi dan *use case* untuk disabilitas atau wali, diagram *activity*, perancangan tabel *database*, dan perancangan antarmuka pengguna dari aplikasi evakuasi bencana berbasis *android*.



Gambar 2 *Use Case User (Tim Evakuasi)*

**2. Diagram Activity**

**Implementasi Algoritma Dijkstra Untuk Penentuan Jalur Terpendek Pada Aplikasi Evakuasi Bencana Untuk Penyandang Disabilitas**



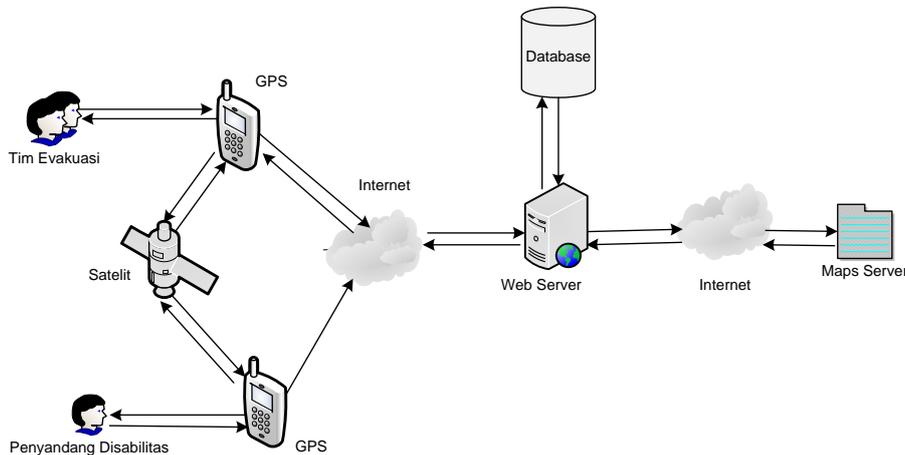
Gambar 3 Diagram Activity User Melihat Lokasi Penyandang Disabilitas

1. User memilih aplikasi evakuasi bencana untuk penyandang disabilitas
2. Sistem menampilkan tampilan menu utama yang terdapat beberapa menu atau *button* seperti *location search, maps, login* dan *about*. User memilih menu pencarian koordinat lokasi
3. Sistem merespon pencarian lokasi dengan menampilkan lokasi sesuai pilihan *user* dalam bentuk peta *google maps*, sekaligus menampilkan jalur terpendek dari lokasi *user* ke lokasi penyandang disabilitas.
4. Setelah itu jika *user* ingin melihat informasi mengenai penyandang disabilitas, *user* dapat memilih *icon* lokasi penyandang disabilitas pada *maps*.

3. Diagram Blok

Berikut ini merupakan diagram *blok* aplikasi evakuasi bencana untuk penyandang disabilitas berbasis *android*.

Keterangan :



Gambar 3 Diagram Blok Sistem Evakuasi Bencana Untuk Penyandang Disabilitas

Sistem yang dibuat terdiri dari aplikasi evakuasi bencana, *web server* dan *database* yang saling terhubung. *Database* yang digunakan yaitu *MySQL* yang berisi data *user* (penyandang disabilitas). Perangkat *android* akan berkomunikasi dengan *database* untuk memanggil maupun menyimpan data. Selain itu perangkat *android* juga akan terhubung dengan *google maps* serta terhubung dengan satelit *GPS*. *GPS* berguna sebagai *tracking* dan memberi tahu lokasi *user* (penyandang disabilitas) ke *user* (tim evakuasi). Untuk penentuan *route* terpendek diambil langsung dari *server google maps* yang algoritma *default* penentuan jalurnya menggunakan algoritma *dijkstra*. Nantinya pada aplikasi evakuasi akan dibuat sebuah *class java* yang akan menerima *input-an* berupa dua buah koordinat, titik awal dan titik tujuan. Untuk kemudian di *decode* menggunakan *Google Maps APIs*, yang menghasilkan sebuah file *XML*. Kemudian *XML* tersebut akan diubah menjadi sebuah *ArrayList* berisi *LatLng* (titik koordinat *latitude* dan *longitude*).

IV. HASIL DAN PEMBAHASAN

Pada bab ini menjelaskan tentang hasil implementasi dari perancangan aplikasi pada bab sebelumnya. Hasil yang di bahas berupa fitur aplikasi dan *interfaces* yang terdapat di dalam aplikasi penentuan jalur terpendek untuk evakuasi bencana menggunakan algoritma *dijkstra*. Selain itu pada bab ini juga akan dijelaskan mengenai hasil uji coba terhadap aplikasi penentuan jalur terpendek untuk evakuasi bencana menggunakan algoritma *dijkstra* yang telah dibuat.

1. Implementasi Interfaces

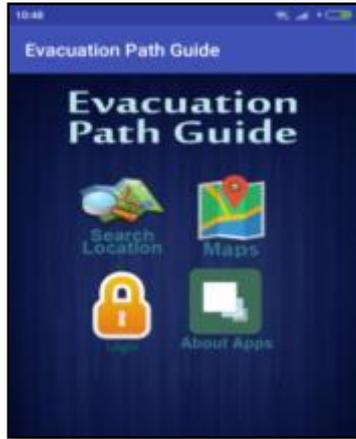
Dalam aplikasi penentuan jalur terpendek untuk evakuasi bencana menggunakan algoritma *dijkstra* ini mengimplemtasikan beberapa *interfaces*.

a. Halaman Utama

Setelah itu aplikasi akan menampilkan halaman menu utama, pada halaman menu utama terdapat beberapa pilihan *button* yaitu *Login, Search Location, Maps* dan *About Apps*. Pada gambar 4 merupakan

## Implementasi Algoritma Dijkstra Untuk Penentuan Jalur Terpendek Pada Aplikasi Evakuasi Bencana Untuk Penyandang Disabilitas

tampilan aplikasi.



Gambar 5 Halaman Utama

### b. Halaman Login

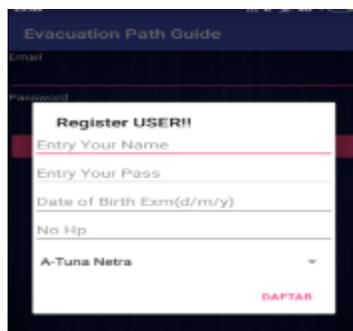
Pada gambar 6 dapat di tampilan aplikasi yang menunjukkan *user* (penyandang disabilitas/wali) dapat melakukan *login* dan registrasi.



Gambar 6 Halaman Login

### c. Halaman Register

Jika *user* melakukan registrasi maka aplikasi akan menampilkan *alert dialog* untuk memasukkan biodata *user* lalu klik daftar seperti gambar 7.

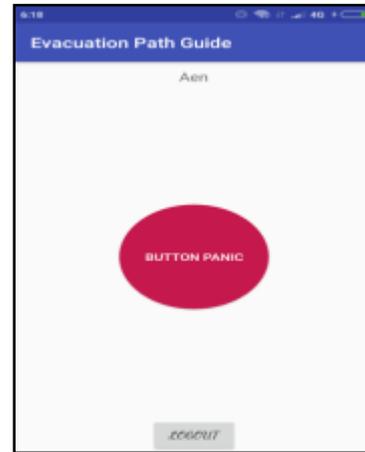


Gambar 7 Halaman Register

### d. Halaman Button Panic

Jika pengguna sudah berhasil melakukan *register* maka akun tersebut dapat digunakan untuk *login* ke aplikasi, jika berhasil *login* maka pengguna akan menampilkan halaman *butto panic* yang fungsinya koordinat jalur lokasi.

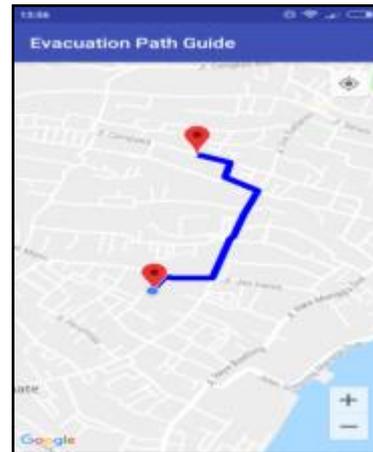
jika *user* (penyandang disabilitas) berada dalam keadaan darurat pengguna dapat menekan tombol tersebut maka secara otomatis aplikasi akan membagikan lokasi keberadaannya ke tim evakuasi. Seperti pada gambar 8.



Gambar 8 Halaman Button Panic

### e. Menampilkan Rute Terpendek

Apabila pengguna disabilitas berada dalam kondisi darurat dan sudah menekan *button panic* maka tim evakuasi secara otomatis menerima koordinat lokasi disabilitas dan secara langsung menampilkan jalur terpendek pada *maps*, seperti pada gambar 9.



Gambar 9 Rute Terpendek pada Maps

## 2. Implementasi Interfaces Pada Sistem Perhitungan Dijkstra Berbasis Web

Pada gambar 10 merupakan rancangan titik-titik koordinat lokasi yang di ambil dari *google maps* dan diterapkan di dalam sistem. Titik koordinat jalur yang diterapkan bobot jarak masing-masing dalam satuan meter. Untuk koordinat jalur beserta bobot jarak diambil dengan memanfaatkan sebuah situs maps yaitu *graplatcoding.com* yang dibuat khusus untuk mempermudah dalam mendapatkan data-data

# Implementasi Algoritma Dijkstra Untuk Penentuan Jalur Terpendek Pada Aplikasi Evakuasi Bencana Untuk Penyandang Disabilitas



Gambar 10 Titik-titik Koordinat Lokasi

Tabel 1 Tabel Jalur Pada Database

Id	No_Trayek	Simpul
1	J01	0-1,1-3,3-4,4-6,6-7,7-8,8-10,10-8,8-7,7-6,6-4,4-3,3-1,1-0,
2	J02	1-2,2-1,
3	J03	3-5,5-6,6-5,5-3,
4	J04	5-4,4-5,
5	J05	8-9,9-8,
6	J06	7-10,10-7,

Pada tabel 1 menunjukkan simulasi jalur-jalur yang sudah di sediakan di dalam database. Pada kolom "No\_Trayek" menunjukkan nama jalur yang terdiri dari 6 jalur yaitu J01, J02, J03, J04, J05 dan J06. Dari setiap jalur terdapat "Simpul" yang ditandai dalam bentuk angka-angka. Angka-angka tersebut menunjukkan nama jalan yang di lalui, diberi simbol dengan angka agar dalam implementasi ke sistem lebih mudah.

### Pseudocode algoritma Dijkstra

```

function Dijkstra(Graph, source):
for each vertex v in Graph: //
  Initializations
  dist[v] := infinity
  previous[v] := undefined
  dist[source] := 0 //
  Distance from v to v
Q := semua node dalam Graph // Set
of all vertices
while Q is not empty: // The
algorithm itself
  u := node dalam Q dengan jarak terkecil
  hapus u dari Q
  for each tetangga v dari u:
    alt := dist[u] + dist_between(u, v)
    if alt < dist[v] //
      Relax (u,v)
      dist[v] := alt
      previous[v] := u
return previous[ ]
    
```

### 3. Pengujian

Pada tahap ini aplikasi yang sudah dibuat akan dilakukan ujicoba untuk memastikan bahwa aplikasi berjalan sesuai perancangan dan algoritmanya pun berjalan dengan benar. Pengujian yang dilakukan terdiri dari 2 yaitu sebagai berikut.

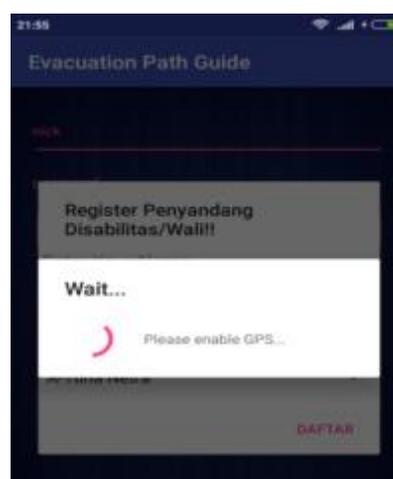
#### a. Pengujian Sistem

Pengujian sistem dilakukan menggunakan metode yang *blackbox* untuk mengamati dan mengetahui

keluaran dari berbagai masukan, jika aplikasi evakuasi bencana telah sesuai dengan perancangan untuk variasi data, maka sistem tersebut dinyatakan baik.

Tabel 2 Tabel Pengujian Register Aplikasi

Hasil Uji Data Normal			
Data Masukkan	Yang Diharapkan	Pengamatan	Kesimpulan
Input data registrasi	Menampilkan data konfirmasi	Hasil sesuai dengan yang diharapkan	Diterima
GPS Aktif saat registrasi	Menampilkan koordinat Lokasi	Hasil sesuai dengan yang diharapkan	Diterima
Hasil Uji (Data Tidak Normal)			
GPS tidak aktif saat registrasi	Muncul pesan "Please enable GPS"	Hasil sesuai dengan yang diharapkan	Diterima



Gambar 11 GPS Tidak Aktif Saat Registrasi

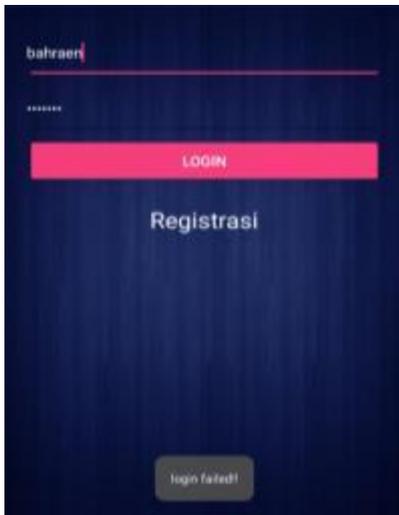
Tabel 3 Tabel Pengujian Login Aplikasi

Hasil Uji Data Normal			
Data Masukkan	Yang Diharapkan	Pengamatan	Kesimpulan
Username : aen Password	Menampilkan halaman button panic	Hasil sesuai dengan yang diharapkan	Diterima

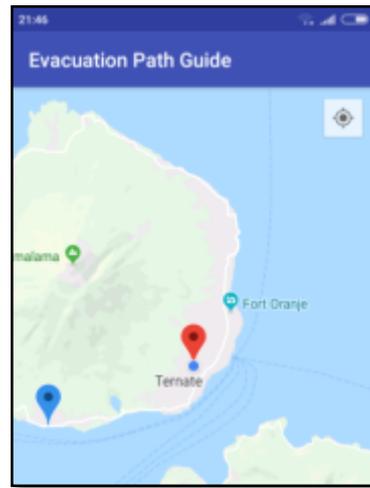
**Implementasi Algoritma Dijkstra Untuk Penentuan Jalur Terpendek Pada Aplikasi Evakuasi Bencana Untuk Penyandang Disabilitas**

Hasil Uji Data Normal			
Data Masukkan	Yang Diharapkan	Pengamatan	Kesimpulan
: aen			
Hasil Uji (Data Tidak Normal)			
Username : bahraen Password : bahraen	Muncul pesan "login failed"	Hasil sesuai dengan yang diharapkan	Diterima
User dan password belum dikonfirmasi oleh admin	Muncul pesan "login failed"	Hasil sesuai dengan yang diharapkan	Diterima
GPS tidak aktif saat login	Muncul pesan "Please enable GPS"	Hasil sesuai dengan yang diharapkan	Diterima

Hasil Uji Data Normal			
Data Masukkan	Yang Diharapkan	Pengamatan	Kesimpulan
Pilih lokasi di listview	Menampilkan maps dan lokasi sesuai pilihan	Hasil sesuai dengan yang diharapkan	Diterima
Koneksi Internet & GPS aktif	Menampilkan rute terpendek pada maps	Hasil sesuai dengan yang diharapkan	
Hasil Uji (Data Tidak Normal)			
Koneksi Internet Buruk	Rute di maps tidak tampil	Hasil sesuai dengan yang diharapkan	Diterima



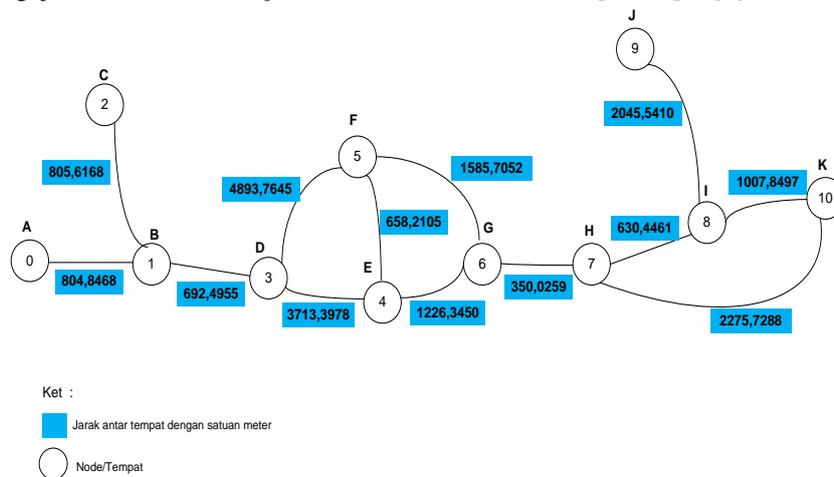
Gambar 12 Login Failed



Gambar 13 Rute Tidak Tampil Saat Koneksi Internet Buruk

Tabel 4 Tabel Pengujian Listview dan Maps

b. Pengujian Algoritma Dijkstra Pada Maps  
Berikut ini merupakan titik-titik koordinat lokasi beserta jaraknya yang terdapat pada gambar 14 kemudian di konversi ke dalam bentuk *grap* agar lebih mudah dalam proses pengujian.



Gambar 14 Titik-titik Koordinat Lokasi

Tabel 5 Pengujian Algoritma Dijkstra

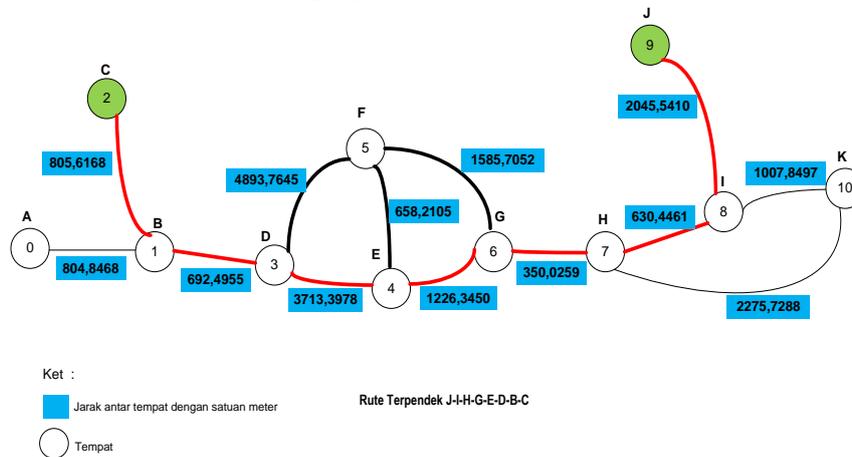
Pengujian 1
Koordinat Awal Kampus Jati (J) : 0.7759705,127.3732416

Koordinat Tujuan SLB Negeri Sasa Kota Ternate (C) : 0.75526, 127.32484	
Relasi Vertex 1 : J-I-H-G-F-D-B-C	11003.5950 meter

## Implementasi Algoritma Dijkstra Untuk Penentuan Jalur Terpendek Pada Aplikasi Evakuasi Bencana Untuk Penyandang Disabilitas

Relasi Vertex 2 : J-I-H-G-E-F-D-B-C	11302,4453 meter
Relasi Vertex 3 : J-I-H-G-E-D-B-C	9463,8681 meter
Relasi Vertex Terpilih adalah 3 = J-I-H-G-E-D-B-C	= 9463,8681 meter

Pada tabel 5 dapat di lihat hasil pengujian



Gambar 15 Rute Terpendek

Tabel 6 Tabel Jalur di Maps

Nama Jalan dalam Bentuk Huruf	Nama Jalan dalam Bentuk Angka di Maps	Nama Jalur di Maps
J - I	9 - 8	J05
I - H	8 - 7	J01
H - G	7 - 6	J01
G - E	6 - 4	J01
E - D	4 - 3	J01
D - B	3 - 1	J01
B - C	1 - 2	J02

Untuk hasil perhitungan *dijkstra* pada sistem berbasis *web* dapat dilihat pada gambar 16.



Gambar 16 Rute Terpendek pada Maps (J05-J01-J02)  
 Sedangkan untuk tampilan *rute* terpendek pada

### 4. Analisis Hasil Pengujian

Berdasarkan pengujian yang telah dilakukan yaitu pengujian sistem dengan metode *blackbox* menunjukkan bahwa aplikasi evakuasi bencana yang dibuat terdiri dari 4 menu yaitu halaman *login*, *maps*, *about*, dan *search location*, selain itu lokasi pengguna yang di tampilkan pada *maps* dapat di tampilkan dengan memanfaatkan GPS *smartphone*, aplikasi dapat terkoneksi dengan *database server* sehingga

algoritma *dijkstra* secara manual (menggunakan *excel*) dengan *rute* terpendek dari J ke C yang diperoleh adalah pada *relasi vertex* 3 = J-I-H-G-E-D-B-C dengan bobot atau jarak 9463,8681 meter. Untuk *rute* terpendeknya dapat dilihat pada gambar 4.33 yang ditandai dengan garis berwarna merah.

dapat melakukan penyimpanan maupun pengambilan data. Dengan demikian maka aplikasi evakuasi bencana sudah berjalan sesuai dengan perancangan sistem.

Untuk pengujian algoritma yang di lakukan pada sistem berbasis *web* dapat di implemtasikan algoritma *dijkstra* dan mampu menunjukkan *rute* terpendek menuju lokasi tujuan, hanya saja pada sistem berbasis *web* lokasi tujuannya sudah ditetapkan di dalam *database* sedangkan lokasi awal di tetapkan sendiri oleh pengguna tanpa menggunakan GPS. Dengan demikian jika di lakukan *zoom out* pada *maps* yang terdapat pada sistem maka terlihat titik koordinat awal yang di ambil untuk perhitungan *dijkstra* tidak sesuai dengan posisi *user* melainkan yang diambil adalah koordinat yang berdekatan dengan posisi *user* kecul jika posisi *user* tepat berda pada jalur yang sudah di tetapkan.

Dari pengujian tersebut di simpulkan bahwa perhitungan algoritma *dijkstra* hanya dapat dilakukan pada koordinat-koordinat lokasi yang sudah ada di dalam *database* jika ada koordinat lokasi baru yang tidak ada di dalam *database* maka sistem akan mengambil koordinat yang terdekat dengan koordinat baru tersebut. Berbeda dengan aplikasi evakuasi bencana yang menunjukkan *rute* terpendek langsung dari titik awal dan tujuan, hal ini di karenakan koordinat jalur di ambil dari *google maps* API sehingga semua koordinat lokasi sudah tersedia.

## V. PENUTUP

### 1. Kesimpulan

Berdasarkan hasil penelitian yang telah di lakukan dengan mengimplemantasikan algoritma *dijkstra* pada aplikasi evakuasi bencana dan sistem berbasis *web*

## **Implementasi Algoritma Dijkstra Untuk Penentuan Jalur Terpendek Pada Aplikasi Evakuasi Bencana Untuk Penyandang Disabilitas**

untuk pencarian rute terpendek dapat di simpulkan bahwa :

1. Pembuatan aplikasi evakuasi bencana pada *platform android* dapat dilakukan dengan menggunakan bahasa pemrograman *java* dengan memanfaatkan *android studio* sebagai *Integrated Development Environment (IDE)*. Pembuatan basis data juga dapat dilakukan dengan memanfaatkan dua *database* yaitu *android SQLite sever* dan *MySQL* dengan koneksinya menggunakan *PHP*.
2. Metode pengembangan perangkat lunak *prototype* dapat digunakan untuk pengembangan aplikasi evakuasi bencana yang terdiri dari beberapa tahapan yaitu analisis di mana dilakukan analisis tentang proses penanganan penyandang disabilitas ketika proses evakuasi bencana. Desain sistem dilakukan mulai dari perancangan *UML*, basisdata sampai pada perancangan *interfaces*. Pengkodean, dimana desain sistem dikonversi ke bahasa pemrograman dan terakhir yaitu pengujian sistem dengan menggunakan *blackbox*.
3. Pemanfaatan *GPS* pada aplikasi evakuasi bencana dapat diimplementasikan sehingga mempermudah aplikasi untuk mendeteksi titik koordinat lokasi *user*.
4. Berdasarkan hasil perbandingan algoritma *dijkstra* diantara aplikasi evakuasi bencana dan sistem berbasis *web* bahwa keduanya menunjukkan *rute* terpendek yang sama dengan bobot jarak yang berbeda. Hal ini di karenakan pada sistem

### **2. Saran**

Dari pembuatan aplikasi evakuasi bencana sampai dengan hasil dan pengujiannya masih memiliki banyak kekurangan, untuk itu ada beberapa saran yang dapat diberikan penulis yaitu:

1. Dari hasil di atas perlu di adakan penelitian yang lebih lanjut untuk memaksimalkan fungsi aplikasi evakuasi bencana dengan penambahan fitur yang dapat menampilkan jarak dan waktu tempuh dari lokasi awal ke lokasi tujuan.
2. Ada sekitar 11 jenis penyandang disabilitas dengan keterbatasan yang berbeda-beda maka tidak semua penyandang disabilitas dapat menggunakan aplikasi ini sehingga memungkinkan penggunaan aplikasi masih harus digunakan dengan bantuan wali, untuk itu pada penelitian selanjutnya hal ini bisa dijadikan sebagai bahan pertimbangan.
3. Untuk lebih memaksimalkan proses evakuasi aplikasi masih perlu ditambahkan titik-titik jalur evakuasi agar lebih mempermudah tim evakuasi dalam melakukan proses evakuasi.

### **DAFTAR PUSTAKA**

- [1] R. Probosiwi, "Keterlibatan Penyandang Disabilitas Dalam Penanggulangan Bencana," vol. 4, hal. 13–22, 2013.
- [2] S. J. Kesdm, A. Febriani, C. P. Sari, dan D.

- Ester, "Kementerian Energi Dan Sumber Daya Mineral," 2015. [Daring]. Tersedia pada: <http://vsi.esdm.go.id>. [Diakses: 22-Sep-2017].
- [3] M. S. Handaka, "Perbandingan Algoritma Dijkstra ( Greedy ), Bellman-Ford ( BFS-DFS ), dan Floyd-Warshall ( Dynamic Programming ) dalam Pengaplikasian Lintasan Terpendek pada Link-State Routing Protocol," *informatika*, hal. 1–8, 2011.
- [4] B. R. Rompas, "Aplikasi Location-Based Service Pencarian Tempat Di Kota Manado Berbasis Android," no. 1, hal. 1–11, 2013.
- [5] U. Hasanah, N. Safriadi, dan Tursina, "Location Based Service Lokasi Masjid Pontianak Menggunakan Metode Dijkstra Berbasis Android," *J. Sist. dan Teknol. Inf.*, hal. 1–6, 2015.
- [6] N. Muslim dan A. Sunyoto, "Sistem Informasi Geografis Berbasis Pemetaan Potensi Panas Bumi Di Indonesia Menggunakan Google Maps," *J. Dasi*, vol. 13, no. 2, hal. 60–64, 2012.
- [7] F. Mahdia dan F. Noviyanto, "Pemanfaatan Google Maps API untuk Pembangunan Sistem Informasi Manajemen Bantuan Logistik Pasca Bencana Alam Berbasis Mobile Web," *J. Sarj. Tek. Inform.*, vol. 1, no. 1, hal. 162–171, 2013.
- [8] Fitria dan A. Triansyah, "Implementasi Algoritma Dijkstra Dalam Aplikasi Untuk Menentukan Lintasan Terpendek Jalan Darat Antar Kota Di Sumatera Bagian Selatan," *J. Sist. Inf.*, vol. 5, no. 2, hal. 611–621, 2013.
- [9] D. S. Budi, T. A. Y. Siswa, dan H. Abijono, "Analisis Pemilihan Penerapan Proyek Metodologi Pengembangan Rekayasa Perangkat Lunak," *Teknika*, vol. 5, no. November, hal. 24–31, 2016.