

## **Jalur Evakuasi Tsunami pada Sistem Operasi Android Menggunakan Algoritma Dijkstra**

**Defri Ahmad<sup>1\*</sup>, Rara Sandy Winanda<sup>2</sup>, Hafis Alrizal<sup>3</sup>**

<sup>1,2,3</sup> Program Studi Matematika, Universitas Negeri Padang

Jl. Prof. Dr. Hamka, Air Tawar Barat, Padang

Email: rarawinanda89@gmail.com, hafisalrizal@gmail.com,

\*Korespondensi penulis : defri\_math@fmipa.unp.ac.id

### **Abstrak**

Padang adalah salah satu kota yang rentan terhadap tsunami, akan tetapi masyarakat yang tinggal/ bekerja di tepi pantai sangat ramai sehingga evakuasi secara horizontal tak mungkin dilakukan. Karena itu, dalam penelitian ini evakuasi dilakukan secara vertikal. Untuk menentukan jalur evakuasi tsunami terpendek di Kota Padang dalam tulisan ini digunakan Algoritma Dijkstra. Dalam Algoritma Dijkstra penentuan jalur terpendek dapat dicari untuk semua pasangan simpul. Penelitian ini adalah penelitian terapan yang dimulai dengan mempelajari dan menganalisis teori-teori yang relevan dengan masalah, kemudian dilanjutkan dengan pengambilan data. Penelitian ini dimulai dengan mengubah peta yang diunduh dari Google Map menjadi graf berbobot, kemudian graf tersebut diubah menjadi matriks ketetanggaan dan diproses menggunakan Algoritma Dijkstra. Hasil pemrosesan aplikasi menggunakan Algoritma Dijkstra memperoleh jalur evakuasi tsunami terpendek yang dapat dilewati ke tempat penampungan terdekat berdasarkan posisi pengguna perangkat Android.

**Kata Kunci:** Gempa, Tsunami, Dijkstra, Lintasan Terpendek

### **Abstract**

Padang is one of city that is susceptible against tsunami, but the density of people near offshore fails the horizontally evacuate process. Therefore, in this research the evacuation is done vertically. To determine the shortest path of tsunami evacuation in Padang City is using Dijkstra Algorithm. In the Dijkstra Algorithm the shortest path determination can be searched for all pairs of vertices. This research is an applied research that begins with studying and analyzing theories relevant to the problem, then proceed with data retrieval. This research begins by transforming Maps downloaded from Google Map into a weighted graph, then the graph is transformed into an adjacent matrix and processed using the Dijkstra Algorithm. The results of application processing using Dijkstra Algorithm obtained the shortest path of tsunami evacuation that can be passed to the nearest shelter based on the position of Android device users.

**Keywords:** Earthquake, Tsunami, Dijkstra, The Shortest Path

## 1. Pendahuluan

Salah satu bencana alam yang sering terjadi di Indonesia yaitu gempa bumi. Provinsi di Indonesia yang rawan akan terjadi gempa bumi diantaranya adalah provinsi Sumatera Barat. Hal ini disebabkan karena letaknya yang berada pada jalur patahan Semangko dan pertemuan lempeng Indo-Australia dengan lempeng Eurasia, yang sering menjadi sumber gempa. Selain itu, gempa di Sumatera Barat juga disebabkan oleh aktivitas gunung api. Gempa bumi yang terjadi di dasar laut dengan magnitudo lebih dari 6,0 skala Richter dengan kedalaman pusat gempa kurang dari 60 km akan dapat menjadi penyebab terjadinya tsunami [1].

Tsunami adalah perpindahan badan air yang disebabkan oleh perubahan permukaan laut secara vertikal dengan tiba-tiba. Perubahan permukaan laut tersebut bisa disebabkan oleh gempa bumi yang berpusat di bawah laut, letusan gunung berapi bawah laut, longsor bawah laut, atau hantaman meteor di laut. Kebanyakan Tsunami berawal dari gangguan seismic bawah laut [5].

Akibat tsunami diantaranya yang pertama adalah dapat merusak bangunan seperti rumah, gedung-gedung, jalan raya dan lain sebagainya yang berada di sekitar pantai maupun bukan sekitar pantai akibat gelombang tsunami yang begitu besar. Kedua timbulnya beberapa penyakit pasca tsunami seperti diare, infeksi, dan ISPA. Ketiga munculnya trauma terutama pada anak-anak, biasanya anak-anak merasakan tekanan psikologi seperti perasaan takut berpisah, takut pada laut setelah terjadinya tsunami. Keempat adalah jatuhnya korban jiwa akibat tsunami yang meyapu mereka.

Pada Gambar 1 terlihat bahwa daerah yang berwarna merah sebelah barat adalah zona rawan tsunami sedangkan daratan yang berwarna kuning sebelah timur adalah zona aman tsunami. Menurut [6], tsunami di Indonesia pada umumnya dapat mencapai pantai sekitar 10-60 menit setelah terjadi gempa. Masyarakat yang sedang melakukan aktifitas ataupun tinggal disekitar pantai ketika tanda-tanda sebelum datangnya tsunami dapat melarikan diri ketempat yang aman dan jauh dari pantai tersebut agar ombak yang dirasakan tidak begitu kencang seperti jika berada disekitar pantai. Waktu yang tersedia untuk melarikan diri ke tempat yang aman tidak lah lama untuk itu masyarakat harus menempuh jalur terdekat agar bisa sampai ke tempat yang jauh dari pantai dan tempat tersebut termasuk tempat yang lebih aman dari pada tempat yang sebelumnya.



Gambar 1. Peta Evakuasi Kota Padang (Sumber : BPBD Kota Padang)

Jalur evakuasi tsunami dibutuhkan dimanapun keberadaan seseorang terlebih jika orang tersebut berada disekitar pantai atau berada pada zona merah. Beberapa algoritma bisa digunakan dalam menentukan rute terpendek salah satunya algoritma Dijkstra. algoritma ini dapat menentukan jalur terpendek dengan waktu yang lebih cepat dari algoritma lainnya. Pada algoritma ini hanya memikirkan solusi terbaik yang akan diambil pada setiap langkah tanpa memikirkan konsekuensi ke depan, dan ini dapat mempercepat pemrosesan atau kalkulasi pada sistem operasi Android [2]. Sedangkan algoritma Floyd-Warshall menggunakan prinsipdinamis yang melakukan pemecahan masalah dengan memandang solusi yang akan diperoleh sebagai suatu keputusan yang saling terkait. Artinya solusi-solusi tersebut dibentuk dari solusi yang berasal dari tahap sebelumnya dan ada kemungkinan solusi lebih dari satu sehingga algoritma ini cenderung membutuhkan waktu yang lebih lama untuk menghasilkan rute terpendek dibandingkan dengan algoritma Djisktra [2], [3]. Sedangkan algoritma Bellmand-Ford adalah algoritma yang digunakan untuk graph yang memiliki sisi dengan bobot negatif sehingga representasi pada kedalam bentuk graph tidak ada yang memiliki sisi dengan bobot negatif [7].

Android adalah Sistem Operasi yang dirancang untuk perangkat berlayar sentuh seperti smartphone dan komputer, atau tablet [6]. Hampir semua vendor saat ini mengembangkan produknya dengan sistem operasi Android, karena peminatnya yang semakin meningkat tajam mulai dari kalangan remaja sampai tua. Seiring berkembangnya zaman tentu manusia butuh peralatan yang canggih dan efisien untuk keperluan sehari-hari.

Penelitian difokuskan pada penentuan lintasan terpendek untuk mencapai tempat yang aman dalam menyelamatkan diri saat akan terjadinya tsunami dengan mengaplikasikan cara kerja algoritma Djisktra kedalam sebuah program android.

## 2. Metode Penelitian

Penelitian yang dilakukan merupakan penelitian dasar (teoritis), dengan menganalisis teori-teori yang relevan terhadap permasalahan yang dibahas berdasarkan pada kajian kepustakaan seperti memahami konsep dasar graf, jalur terpendek dengan graf, dan algoritma Dijkstra. Dalam meninjau permasalahan yang dihadapi, langkah kerja yang dilakukan adalah sebagai berikut :

1. Studi Literatur.
2. Deskripsi Data
3. Analisis Sistem.
4. Perancangan Aplikasi.
5. Pembuatan Program.
6. Simulasi dan Pengujian.
7. Analisis Hasil Simulasi.

## 3. Hasil dan Pembahasan

Penentuan lintasan terpendek untuk mencapai tempat yang aman dalam menyelamatkan diri saat akan terjadinya tsunami dengan mengaplikasikan algoritma Dijkstra kedalam sebuah program android dimulai dengan deskripsi data, analisis data, perancangan aplikasi, pembuatan program, simulasi dan pengujian, dan analisis hasil simulasi tentang masalah penentuan jalur terpendek evakuasi tsunami menggunakan Algoritma *Dijkstra*.

### 3.1 Deskripsi Data

Dalam menyelesaikan permasalahan penentuan jalur terpendek evakuasi tsunami di kota padang diasumsikan bahwa :

1. Simpul pada graf diasumsikan dengan persimpangan jalan, shelter dan zona aman.
2. Sisi pada graf diasumsikan dengan ruas jalan yang menghubungkan persimpangan.
3. Bobot sisi adalah jarak atau panjang ruas jalan yang menghubungkan persimpangan.

Untuk meningkatkan ke akuratan perhitungan (karena permukaan bumi yang lengkung) jarak statis antara dua titik dihitung dengan menggunakan persamaan Harvesine [4] yaitu :

$$\text{jarak} = 2r \cdot \arcsin \left\{ \sqrt{A + B} \right\}$$

dengan

$$A = \sin^2 \left( \frac{Lat_1 - Lat_2}{2} \right)$$

$$B = \cos(Lat_1) \cdot \cos(Lat_2) \cdot \sin^2 \left( \frac{Long_1 - Long_2}{2} \right)$$

$Lat_1$  Latitude titik pertama

$Long_1$  Longitude titik pertama

$Lat_2$  Latitude titik kedua

$Long_2$  Longitude titik kedua

### 3.2 Analisis Data

Langkah pertama yang dilakukan adalah merepresentasikan peta kota padang ke dalam suatu graf berbobot yang diunduh dari Google Maps. Formula yang digunakan untuk menghitung jarak statis perangkat ke shelter adalah sebagai berikut :

Misalkan  $G$  adalah graph yang didapat dari Google Maps,  $V$  himpunan titik dari graph  $G$  dan  $S$  adalah himpunan sisi dari graph. Misalkan  $u, v$  adalah vertex pada graph  $G$  dan maka himpunan sisi  $S$  dari graf ini adalah pasangan vertices  $(u,v)$  yang melambangkan hubungan dari vertex  $u$  ke vertex  $v$  untuk setiap  $u,v \in V$ . Bobot dari semua sisi dihitung dengan fungsi :  $w: S \rightarrow [0, \infty)$ , sehingga  $w(u,v)$  adalah jarak tak-negatif dari vertex  $u$  ke vertex  $v$ .

Misalkan posisi pengguna perangkat atau titik awal adalah  $a$  dan posisi shelter atau titik tujuan adalah  $b$ , maka langkah-langkah untuk mendapatkan rute terpendek dari titik  $a$  ke titik  $b$  adalah sebagai berikut

1. Tentukan panjang sisi atau jarak antar titik yang terhubung dengan fungsi  $w(u,v)$ ,  $u,v \in V$  dan  $u, v$  terhubung. Lalu set nilai 0 pada titik awal  $a$  dan nilai  $\infty$  pada titik lain pada graph  $G$ .
2. Beri label "FALSE" untuk semua titik dan beri label "START" untuk titik awal.
3. Dari titik berlabel "START", pertimbangkan titik tetangga yang berlabel "FALSE" dan periksa jaraknya dari titik yang berlabel "START".
4. Setelah mempertimbangkan setiap jarak terhadap titik tetangga, tandi titik tetangga tadi yang berlabel "FALSE" dengan "TRUE". Titik yang berlabel "TRUE" tidak akan pernah di cek kembali, jarak yang disimpan adalah jarak terakhir dan yang paling minimal bobotnya.
5. Beri label "START" pada titik yang berjarak minimal dari titik berlabel "START" sebelumnya. Kembali ke langkah ke 3 hingga semua titik berlabel "TRUE" dan kalkulasi *dijkstra* dinyatakan selesai.

### 3.3 Perancangan Aplikasi

Perancangan aplikasi ini tergambar pada gambar 2 mengenai arsitektur aplikasi.



Gambar 2. Rancangan Aplikasi

### 3.4 Pembuatan Aplikasi

Proses pembangkitan rute ini dimulai dari inisialisasi koordinat asal dan tujuan. Koordinat asal yaitu koordinat perangkat yang telah didapatkan sebelumnya dengan GPS. Koordinat tujuan yaitu koordinat shelter-shelter terdekat yang telah disaring pada proses sebelumnya. Selanjutnya aplikasi mengubah peta dari Google Maps kedalam bentuk graph dan menentukan bobot setiap sisi (ruas jalan). Formula yang digunakan untuk menentukan bobot sisi sama dengan formula yang digunakan untuk menentukan jarak statis pada proses penyaringan shelter.

### 3.5 Simulasi

Untuk simulasi dan pengujian akan diambil titik awal (posisi perangkat) yaitu gedung dekanat FMIPA. Gedung dekanat FMIPA (titik awal) berada pada posisi geografis dengan nilai latitude dan longitude yaitu  $-0.896616$  dan  $100.346743$ . Selanjutnya menghitung jarak statis dari titik awal ke semua shelter yang ada dikota padang. Dari koordinat awal  $(-0.896616, 100.346743)$ , dipilih 5 shelter yang memiliki jarak statis terkecil dari titik awal yaitu shelter Perpustakaan UNP, FIK, LPMP, MKU dan PASCA UNP. Setelah itu satu persatu dari 5 shelter tersebut akan diinputkan kedalam algoritma djikstra untuk mendapatkan jalur terpendek masing-masing nya. seperti terlihat pada Lampiran 2, shelter Perpustakaan UNP memiliki jalur yang jaraknya paling minimal dari 4 shelter lainnya.

Setelah didapatkan jalur terpendek selanjutnya system mewarnai jalur terpendek tersebut pada Maps yang ditampilkan pada layar perangkat android seperti yang terlihat pada Lampiran 2.

Berikut hasil jalur terdekat dari titik awal  $(-0.896616, 100.346743)$  menuju shelter Perpustakaan UNP dari aplikasi android yang telah dibuat.



Gambar 3. Rute terpendek Ke *shelter* Perpustakaan Pusat UNP

Terlihat bahwa algoritma Dijkstra menghasilkan Perpustakaan Pusat UNP sebagai shelter yang memiliki panjang rute terpendek dari semua shelter lainnya.

Jika pengguna perangkat menginginkan rute menuju ke shelter lainnya yang tersedia, system akan membangkitkan rute terpendek menuju shelter yang dipilih oleh pengguna perangkat Android. Gambar berikut adalah tampilan pada layar perangkat Android jika pengguna memilih rute terpendek menuju shelter LPMP dan gedung MKU UNP.



Gambar 4. Rute terpendek ke *shelter* LPMP

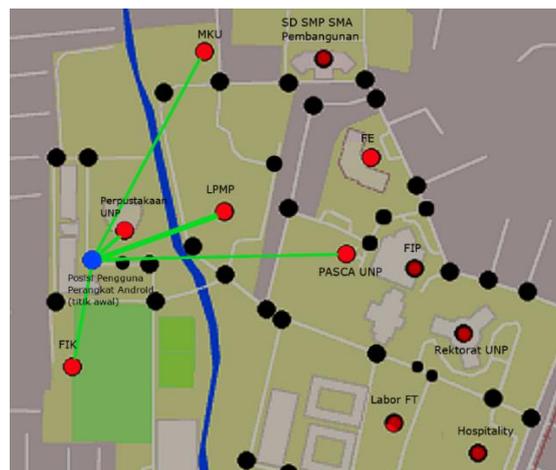


Gambar 5. Rute terpendek ke *shelter* MKU UNP

### 3.6 Analisis Hasil Simulasi

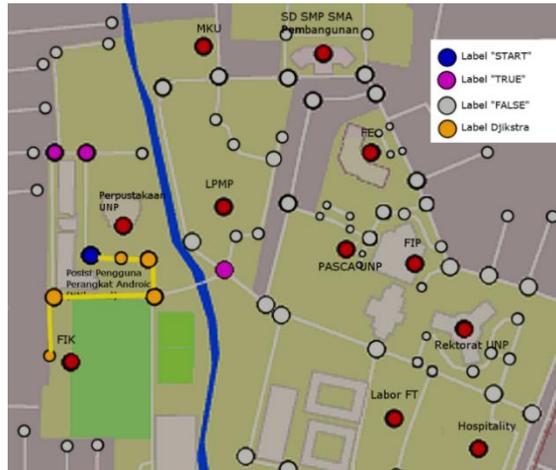
Untuk menguji hasil silmulasi akan dilakukan perhitungan manual untuk dibandingkan hasil simulasi. Pada perhitungan manual akan diinputkan titik awal yang sama dengan titik awal pada saat simulasi.

Dari Gambar 6 berikut terlihat bahwa *shelter* Perpustakaan Pusat UNP, gedung FIK, LPMP, gedung MKU dan gedung pasca memiliki jarak statis terdekat dengan titik awal (-0.896616, 100.346743).

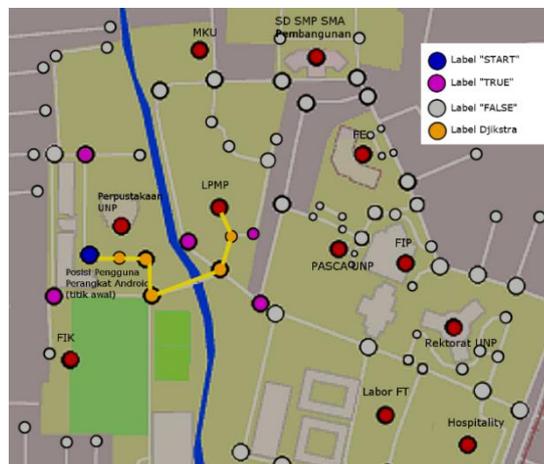


Gambar 6. 5 shelter terdekat

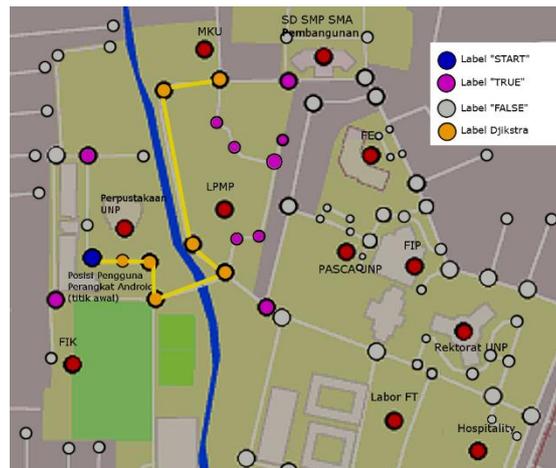
Setelah itu kelima shelter tersebut akan diinputkan kedalam algoritma djikstra untuk mendapatkan rute terdekat ke shelter tersebut. Hasil dari algoritma djikstra untuk kelima shelter tersebut adalah :



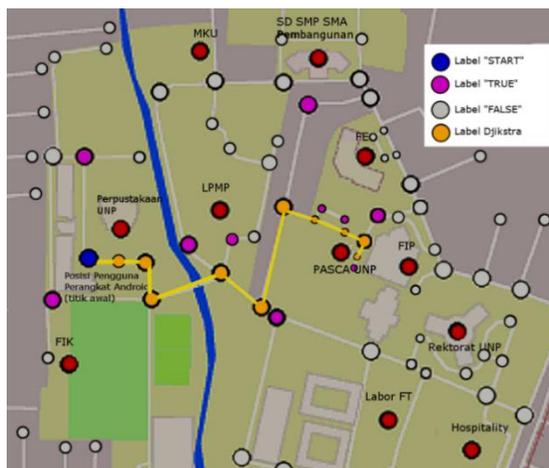
Gambar 7. Rute terpendek menuju *shelter* Gedung FIK UNP



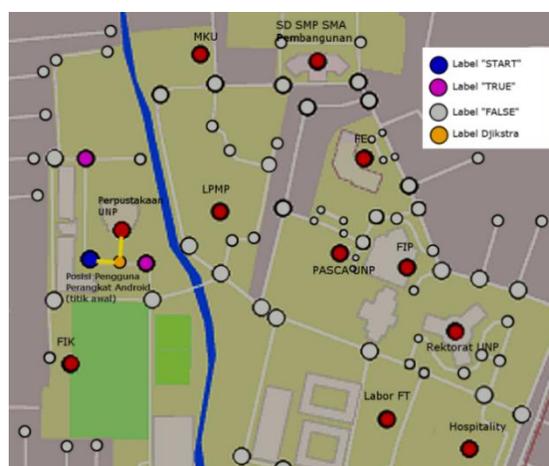
Gambar 8. Rute terpendek menuju *shelter* Gedung LPMP



Gambar 9. Rute terpendek menuju *shelter* Gedung MKU UNP



Gambar 10. Rute terpendek menuju *shelter* Gedung PASCA UNP



Gambar 11. Rute terpendek menuju *shelter* Perpustakaan UNP

Dari hasil penghitungan secara manual dihasilkan rute terpendek yang sama dengan hasil simulasi aplikasi. Dapat disimpulkan bahwa aplikasi jalur evakuasi tsunami dengan algoritma Dijkstra pada sistem operasi android dapat diimplementasikan.

#### 4. Kesimpulan

Berdasarkan pembahasan di atas dapat disimpulkan bahwa :

1. Jalur terpendek evakuasi tsunami dapat dibentuk dengan pemodelan graf menggunakan Algoritma *Dijkstra*. Simpul-simpul pada graf menyatakan persimpangan jalan atau shelter, sedangkan sisi menyatakan ruas jalan yang menghubungkan dua persimpangan.
2. Penggunaan Algoritma *Dijkstra* dalam menentukan jalur terpendek dari lokasi pengguna menuju lokasi yang ditentukan mampu menghasilkan solusi jalur terpendek dengan tepat dan relative cepat untuk di implementasikan kedalam aplikasi android.

Hasil dari aplikasi ini dapat digunakan sebagai alat untuk memudahkan proses evakuasi ketika terjadi tsunami. Selanjutnya, diperlukan penelitian lanjutan untuk

menentukan kecenderungan pilihan tempat evakuasi yang dipilih oleh masyarakat untuk menghindari penumpukan pengungsi. Selain itu juga diperlukan penentuan lokasi shelter yang tepat untuk meratakan jarak rumah masyarakat ke shelter yang tersedia.

### Daftar Pustaka

- [1] I. Iftadi, W. A. Jauhari, and B. Nugroho, "Perancangan Peta Evakuasi Menggunakan Algoritma Floyd- Warshall untuk Penentuan Lintasan Terpendek : Studi Kasus," *PERFORMA Ind. Eng. Sci. Media*, vol. 10, no. 2, pp. 95–104, 2011.
- [2] R. A. D. Novandi, "Perbandingan Algoritma Dijkstra dan Algoritma Floyd-Warshall dalam Penentuan Lintasan Terpendek ( Single Pair Shortest Path )," *IF2251 Strateg. Algorith.*, vol. 1, pp. 1–5, 2013.
- [3] S. Hamdi and Prihandoko, "Analisis Algoritma Dijkstra dan Algoritma Bellman-Ford Sebagai Penentuan Jalur Terpendek Menuju Lokasi Kebakaran (Studi Kasus: Kecamatan Praya Kota)," *J. Ilm. Ilmu-Ilmu Tek.*, vol. 8, no. 1, pp. 26–32, 2018.
- [4] Y. Miftahuddin, S. Umaroh, and F. R. Karim, "Perbandingan Metode Perhitungan Jarak Euclidean, Haversine, Dan Manhattan Dalam Penentuan Posisi Karyawan," *J. Tekno Insentif*, vol. 14, no. 2, pp. 69–77, 2020, doi: 10.36787/jti.v14i2.270.
- [5] BMKG. Pedoman Pelayanan Peringatan Dini Tsunami INATEWS. Jakarta : Badan Meteorologi, Klimatologi dan Geofisika (BMKG) dan GTZ IS-GITEWS. 2012.
- [6] Bryant, Edward. TSUNAMI Bahaya yang Diabaikan. Bandung : Pakar Raya. 1991.
- [7] Retanto, Yudi. Algoritma Dijkstra dan Bellman-Ford dalam Pencarian Jalur Terpendek. Bandung : ITB. 2009.