

PENERAPAN ALGORITMA DIJKSTRA UNTUK PENENTUAN JALUR TERBAIK EVAKUASI TSUNAMI – STUDI KASUS: KELURAHAN SANUR BALI

Eka Ismantohadi¹, Iryanto²

^{1,2}Teknik Informatika Politeknik Negeri Indramayu
Email: ¹ekaismatohadi@gmail.com, ²iryanto.math@gmail.com

Abstrak

Tsunami cukup sering terjadi di Indonesia. Hal ini tidak terlepas dari fakta letak geografis Indonesia yang terletak pada tiga lempeng tektonik utama. Bali termasuk salah satu wilayah yang memiliki potensi tsunami. Mengingat hal tersebut, adanya jalur evakuasi terbaik sebagai sarana mempercepat tindakan untuk menjauhkan warga setempat dari bahaya yang ditimbulkan (evakuasi) menjadi sangat penting. Dalam penelitian ini, dipilih Kelurahan Sanur Denpasar Bali sebagai objek penelitian tersebut. Penelitian ini bertujuan untuk mencari rute/jalur evakuasi terbaik di kelurahan tersebut. Pencarian jalur evakuasi terbaik tersebut dicari dengan algoritma Dijkstra. Dalam artikel ini, hasil jalur evakuasi terbaik dikelompokkan berdasarkan letak tempat evakuasi dan area aman. Selain itu, perbandingan hasil penelitian ini dengan metode pencarian rute lain dari referensi menunjukkan kesesuaian yang baik.

Kata Kunci: evakuasi tsunami, jalur evakuasi tsunami, algoritma Dijkstra

Abstract

Tsunami occurs quite often in Indonesia. It is inseparable from the geographical facts of Indonesia located on three main tectonic plates. Bali is one of the regions that have tsunami potential. Considering these matters, existence of the best evacuation path as a way to evacuate citizen becomes so important. In this research, Kelurahan Sanur Bali is chosen as the research object. Aim of this research is to find the best evacuation path in the district. The best evacuation path is searched using Dijkstra algorithm. In this article, results of the best evacuation path are grouped based on location of evacuation and safe zone. Further, comparison of results of this research and another method from refence is in a good agreement.

Keywords: tsunami evacuation, tsunami evacuation path, Dijkstra algorithm

I. PENDAHULUAN

Tsunami cukup sering terjadi di Indonesia. Tsunami ini merupakan fenomena perambatan gelombang air laut ke segala arah yang terjadi akibat adanya gangguan impulsif pada dasar laut. Tsunami sering terjadi tidak terlepas dari letak Indonesia yang berada pada pertemuan 3 lempeng tektonik utama yaitu Lempeng Samudera India-Australia di sebelah selatan, Lempeng Samudera Pasifik di sebelah timur, Lempeng Eurasia di sebelah utara (dimana sebagian besar wilayah Indonesia berada). Selain itu ditambah Lempeng Laut Philipina (BMKG, 2012).

Hal utama yang menjadi penyebab terjadinya gangguan impulsif tersebut tidak lain adalah perubahan bentuk struktur geologis dasar laut secara vertikal dan peristiwa ini terjadi secara singkat. Adapun penyebab terjadinya perubahan bentuk tersebut adalah longsor, letusan gunung api, dan gempa bumi tektonik yang terjadi di dasar laut. Bahkan disebutkan dalam referensi tersebut bahwa bencana tsunami dapat terjadi sekitar 30 menit setelah gempa bumi terjadi.

Oleh karena itu kajian tentang tsunami merupakan hal yang penting untuk dilakukan. Salah satu kajian ini adalah kajian tentang evakuasi tsunami, tindakan untuk menjauhkan orang-orang dari bahaya tsunami (ketika tsunami terjadi).

Salah satu daerah atau wilayah yang memiliki potensi tsunami adalah Bali. Peta potensi/tingkat resiko tsunami untuk wilayah Bali dapat dilihat pada situs resmi Badan Penanggulangan Bencana Daerah (BPBD) Provinsi Bali (DPPD-Bali, 2017). Tingkat resiko tsunami khusus untuk wilayah Denpasar Bali dapat dilihat di situs resmi Tsunami Kit (Tsunami-Kit, 2010). Dalam penelitian ini, penulis memilih Kelurahan Sanur Denpasar Bali sebagai objek penelitian yang akan dicari jalur terbaik evakuasi tsunami di kelurahan tersebut. Dengan adanya jalur evakuasi terbaik tersebut diharapkan proses evakuasi dapat berjalan dengan cepat dan efisien.

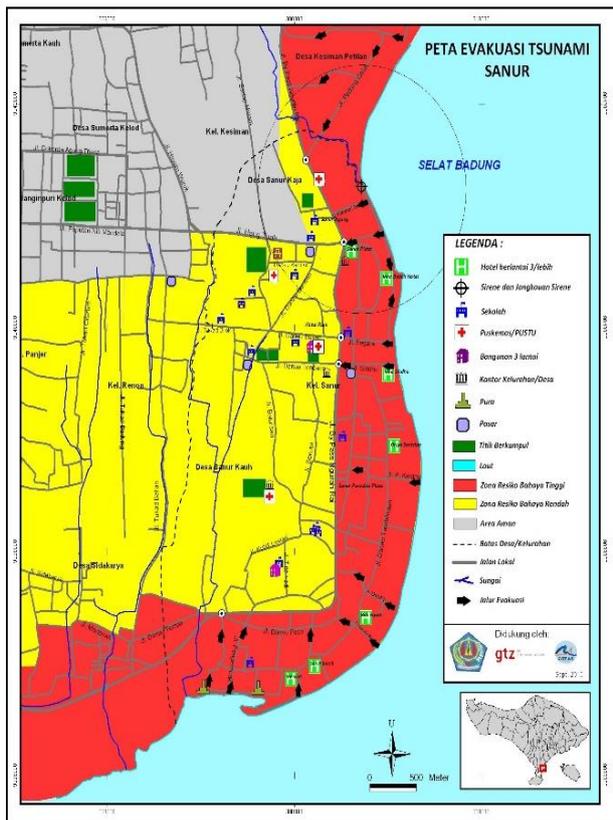
Pemilihan jalur evakuasi tsunami di daerah rawan tsunami telah cukup banyak dikaji oleh para peneliti sebelumnya dua diantaranya adalah pemilihan jalur evakuasi di Sanur Bali (lihat (Sani dkk, 2013) dan Kota

Padang (lihat (Anggria dkk, 2017) dan (Zulfa dan Saputra, 2015)). Di sini Penulis melakukan penelitian dengan objek yang sama dengan (Sani dkk, 2013) namun dengan pendekatan yang berbeda dan rekomendasi jalur terbaik yang berbeda.

Pemilihan jalur evakuasi terbaik dapat dikelompokkan dalam kajian optimasi, sebuah proses untuk mencapai hasil terbaik. Banyak metode yang digunakan dalam menyelesaikan masalah tersebut, salah satunya adalah dengan menggunakan algoritma evolusi, algoritma genetika (Iryanto dan Ismantohadi, 2017). Namun dalam optimasi pemilihan jalur atau lintasan terpendek yang umum digunakan adalah algoritma Dijkstra.

Pemanfaatan algoritma Dijkstra dalam pencarian rute/jalur evakuasi bencana telah banyak dilakukan dalam beberapa tahun sekarang. Sebagai contoh (Pramudya, 2015) menggunakan algoritma tersebut untuk perencanaan rute evakuasi bencana longsor di Kota Semarang. (Hapsari, 2016) memanfaatkan algoritma tersebut untuk optimasi jalur evakuasi letusan gunung merapi. (Herdiyatomoko, 2015) mengimplementasikan algoritma tersebut untuk mensimulasikan jalur evakuasi kebakaran hutan. Algoritma Dijkstra juga tidak hanya digunakan untuk mencari jalur evakuasi bencana namun juga digunakan untuk mencari jalur evakuasi akibat kebakaran sebagaimana yang tertulis di (Dewi, 2018).

Dari beberapa referensi yang telah disebutkan pemanfaatan algoritma Dijkstra banyak digunakan dalam pencarian rute/jalur evakuasi sehingga dalam penelitian ini, Penulis menggunakan algoritma Dijkstra ini sebagai metode mencari jalur evakuasi terbaik tsunami di Kelurahan Sanur Bali.



Gambar 1. Peta Evakuasi Tsunami Sanur Bali [2]

II. METODE

Algoritma Dijkstra

Metode atau algoritma yang umum digunakan dalam menyelesaikan masalah pencarian jalur atau lintasan terpendek adalah metode atau algoritma Dijkstra. Algoritma ini dinamakan seperti nama penemunya Edsger Dijkstra, seorang ilmuwan dari Belanda. Algoritma ini dikembangkan di tahun 1956 dan dipublikasikan secara umum untuk pertama kalinya di tahun 1959 (Dijkstra, 1959).

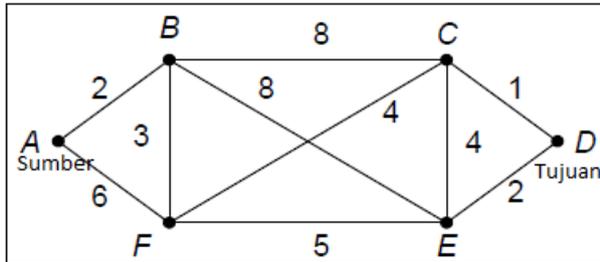
Algoritma Dijkstra menyelesaikan masalah pencarian jalur/lintasan terpendek, sebuah masalah untuk mencari jalur/lintasan antara dua simpul dalam sebuah graf berbobot dengan jumlah total bobot (misal berupa jarak) terkecil, dengan cara mencari jarak terpendek antara simpul awal dengan simpul-simpul lainnya sehingga lintasan yang terbentuk dari simpul awal sampai simpul tujuan memiliki jumlah bobot terkecil.

Terdapat beberapa langkah dalam algoritma ini dalam menyelesaikan masalah pencarian rute terpendek sebagai mana dijelaskan di (Deng dkk, 2012) dan (Cormen dkk, 2009). Adapun langkah-langkah tersebut adalah:

1. Menetapkan jarak pada semua simpul. Jarak simpul sumber (awal) dibuat nol sedangkan jarak simpul lainnya diberikan nilai yang cukup besar (tak hingga).
2. Labeli semua simpul dengan label belum dikunjungi. Simpul-simpul ini dikelompokkan menjadi satu. Simpul awal (sumber) ditetapkan sebagai simpul sekarang.
3. Untuk simpul sekarang, hitung jarak dari semua simpul tetangga yang terhubung dengan node ini. Simpan jarak ini sebagai jarak sementara. Bandingkan jarak sementara ini dengan jarak yang tersimpan sebelumnya. Jika jarak sementara lebih kecil maka jarak sementara ini yang disimpan. Misal node sekarang, S, memiliki jarak tersimpan, JS, bernilai 7 sedangkan terdapat simpul lain yang terhubung dengan simpul sekarang, misal simpul B, dengan jarak 3 sehingga simpul S terhubung dengan B dengan jarak 10. Jika sebelumnya jarak tersimpan bernilai 11 maka jarak tersimpan diperbaharui dengan jarak sementara tersebut.
4. Setelah setelah menghitung jarak dari simpul sumber (awal) dengan simpul tetangga selanjutnya ubah label simpul-simpul tersebut menjadi sudah dikunjungi. Dengan kata lain simpul tersebut telah dihapus dari kumpulan simpul-simpul yang belum dikunjungi. Simpul yang berlabel sudah dikunjungi tidak akan dikunjungi atau dicek lagi.
5. Jika semua telah berlabel sudah dikunjungi atau jarak sementara terkecil bernilai tak hingga maka berhenti melakukan pencarian simpul lain. Algoritma tersebut telah selesai.

Jika tidak demikian, masih ada simpul yang berlabel belum dikunjungi atau jarak sementara terkecil berhingga, pilih simpul dengan jarak/bobot terkecil (dari simpul sumber/awal) sebagai simpul sekarang dan lanjutkan proses dari langkah ke tiga.

Sebagai ilustrasi perhatikan graf berbobot $G = \{V, E\}$ dengan kumpulan simpul V , kumpulan sisi E , pada Gambar. 2. Misalkan simpul-simpul pada graf tersebut menggambarkan daerah evakuasi dan bobot tersebut menggambarkan jarak antara dua wilayah tersebut.



Gambar 2. Ilustrasi Algoritma Dijkstra pada Graf

Jika ditetapkan simpul A sebagai simpul sumber (awal) dan simpul D sebagai simpul tujuan. Dengan menerapkan langkah-langkah algoritma Dijkstra diperoleh:

1. Semua jarak pada selain simpul A bernilai tak hingga.
2. Semua simpul berlabel belum dikunjungi. Simpul sekarang adalah simpul A.
3. Terdapat dua simpul yang terhubung dengan simpul A yaitu simpul B dan F. Jarak A – B lebih kecil dari pada jarak A – F sehingga jarak tersimpan bernilai 2 dan jarak sementara bernilai 6.
4. Labeli simpul B dikunjungi.
5. Tetapkan simpul B sebagai simpul sekarang (karena jarak dari simpul awal, A terkecil)
6. 1. Terdapat tiga simpul yang terhubung dengan simpul B yaitu simpul F, C, dan E. A – B – C berjarak 10, A – B – F berjarak 5, dan A – B – E berjarak 10. Dengan membandingkan ketiga jarak tersebut, jarak A – B – F terpilih sebagai jarak tersimpan dengan nilai 5.
2. Labeli simpul F dikunjungi.
3. Tetapkan simpul F sebagai simpul sekarang
4. Terdapat dua simpul yang terhubung dengan simpul F (dari kumpulan simpul berlabel belum dikunjungi) yaitu E dan C. Jarak F – E bernilai 5 sedangkan F – C bernilai 4. Maka F – C terpilih. A – B – F – C berjarak 9.
5. Labeli simpul C dikunjungi.
6. Tetapkan C sebagai simpul sekarang.
7. Simpul E dan D terhubung dengan C. C – E berjarak 4 dan C – D berjarak 1. Sehingga C – D terpilih. A – B – F – C – D berjarak 10.

8. Labeli simpul D dikunjungi.
9. Tetapkan simpul D sebagai simpul sekarang.
10. Hanya ada simpul E dari daftar simpul yang belum dikunjungi sehingga tidak ada pilihan lain. D – E terpilih. Jarak A – B – F – C – D – E berjarak 12.
11. Labeli simpul E dikunjungi.
12. Tetapkan simpul E sebagai simpul sekarang.
13. Semua label telah dikunjungi sehingga proses selesai.

Dari hasil tersebut diperoleh jarak terdekat dari simpul sumber (awal) dengan masing-masing simpul. Jika tadi ditetapkan simpul D adalah simpul tujuan sehingga jarak dari simpul sumber A ke simpul tujuan D bernilai 10. Sedangkan jarak dari sumber (awal) ke tiap simpul adalah A – B bernilai 2, A – F bernilai 5, A – C bernilai 9, A – D bernilai 10, dan A – E bernilai 12.

Algoritma Dijkstra dapat digunakan dalam pencarian jalur terpendek pada graf yang memiliki bobot bernilai positif. Jika terdapat bobot graf yang negatif algoritma ini tidak dapat digunakan. Algoritma Dijkstra yang aslinya hanya menghasilkan jarak terpendek dari simpul sumber (awal) ke simpul lain. Algoritma ini tidak memberikan jalur itu sendiri (Yan, 2014). Dalam referensi tersebut dijelaskan bahwa sub lintasan (jalur) dari jarak terpendek adalah lintasan terpendek. Sehingga, misalkan terdapat lintasan/jalur terpendek dari simpul sumber S ke simpul V maka jarak simpul V akan sama dengan jarak lintasan terpendek dari S ke V, $d[V] = d(S, V)$. Pembuktian terkait ini dijelaskan secara rinci di (Yan, 2014) dan (Cormen dkk, 2009). Pada praktiknya, dibuat array `simpulPendahulu[NSimpul]` di dalam program sebagai tempat penyimpanan lintasan dan nilainya diperbaharui pada saat jarak terpendek tersebut diperoleh setiap iterasinya.

Penjelasan yang telah dipaparkan tentang algoritma Dijkstra tersebut dapat dirangkum dalam *pseudocode* algoritma Dijkstra untuk graf $G = \{V, E\}$ dengan simpul sumber s dan bobot w berikut ini:

- $dist[s] := 0$
(jarak ke simpul sumber dibuat nol)
- untuk semua $v \in V - \{s\}$
 $dist[v] := infinity$
(jarak simpul lainnya selain simpul sumber dibuat tak berhingga)
- $S := \emptyset$
(S kumpulan simpul yang telah dikunjungi, di awal dibuat kosong)
- $Q := V$
(Q antrian yang berisi simpul yang akan dicek atau dikunjungi, diawal dibuat sama dengan V)
- **while** $Q \neq \emptyset$ (selama antrian tidak kosong)
 $u := jarakmin(Q, dist)$
 (memilih elemen dari Q dengan jarak yang paling minimum)
 $S := S \cup \{u\}$

(menambahkan simpul u pada kumpulan simpul yang telah dikunjungi)

$$\forall v \in V - \{s\}$$

$$\text{if } (dist[u] + w(u, v) < dist[v]) \\ \text{then } d[v] := dist[u] + w(u, v)$$

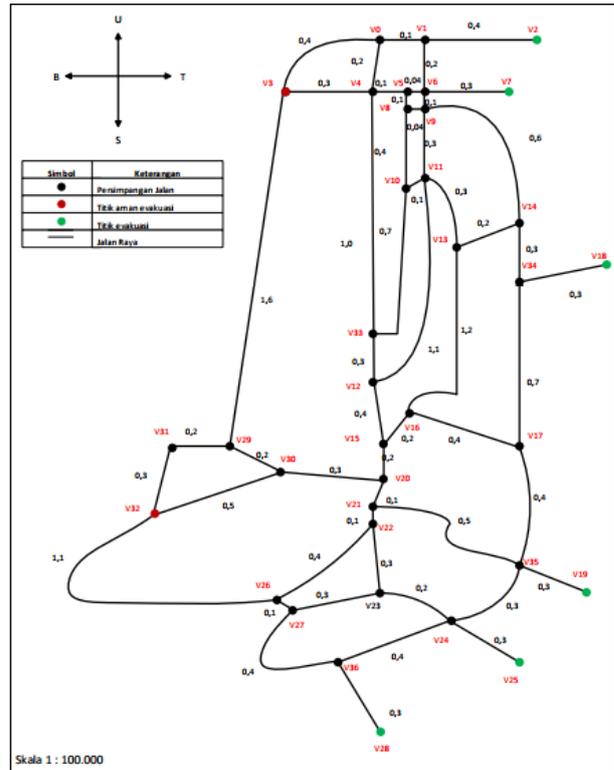
(jika lintasan terpendek yang baru ditemukan, perbaharui jarak terpendek)

$$SimpulPendahulu[v] := u$$

(menambahkan jejak lintasan terpendek)

Implementasi dan Perancangan

Penelitian ini dilakukan dengan beberapa tahapan. Tahapan pertama dimulai dengan pencarian sumber pustaka terkait pencarian jalur terpendek dengan algoritma Dijkstra dan informasi terkait tsunami, dan evakuasi tsunami. Tahapan kedua dilanjutkan dengan pengumpulan data. Dalam hal ini, penulis melakukan pencarian data daerah atau wilayah yang memiliki potensi tinggi terkena tsunami. Pencarian data berujung pada situs web Badan Penanggulangan Bencana Daerah (BPBD) Provinsi Bali (DPPD-Bali, 2017). Dari referensi tersebut dapat dikatakan bahwa Bali termasuk kedalam wilayah yang memiliki potensi tinggi tsunami. Salah satu wilayah yang penulis ambil dari referensi tersebut adalah Kelurahan Sanur. Potensi setiap daerah pada wilayah tersebut diberikan pada Gambar. 1. Jalur evakuasi tsunami juga diberikan pada gambar tersebut namun jalur dari titik-titik berkumpul menuju area (zona) aman belum tersajikan. Selanjutnya penulis melakukan pencarian data yang diperlukan. Dalam hal ini, mencari jarak antara titik-titik evakuasi, daerah lain, dan area (zona) aman. Penulis menggunakan data yang terdapat di referensi (Sani dkk, 2013). Data jarak titik-titik tersebut diberikan pada Gambar. 3. Pada gambar tersebut terdapat enam buah titik evakuasi yang ditandai dengan warna hijau, dua zona aman yang ditandai warna merah dan jarak antar titik (persimpangan jalan) yang ditandai dengan warna hitam.



Gambar 3. Graf Representasi Kelurahan Sanur Bali (Satuan Jarak dalam Kilometer)

Tahapan selanjutnya adalah pembuatan program untuk pencarian jalur terbaik evakuasi tsunami pada Kelurahan Sanur Bali. Program dibuat mengikuti *pseudocode* yang telah dipaparkan pada bahasan algoritma Dijkstra. Program ini dibuat dalam bahasa pemrograman C++. Titik-titik yang tertera pada Gambar. 3 dijadikan sebagai simpul-simpul graf dan jalan-jalan yang menghubungkan sebagai *edge*. Titik-titik evakuasi dijadikan sebagai simpul awal dan zona aman dijadikan sebagai simpul tujuan. Nilai yang tertera diantara dua titik dalam gambar tersebut yang menyatakan jarak antara dua titik (daerah) disimpan kedalam suatu *adjacency matrix*, $adjMatrix[Nsimpul][Nsimpul]$. Di dalam program, nilai *adjacency matrix* dari dua buah simpul (titik) yang tidak terhubung dibuat bernilai sangat besar (tak hingga). Mengingat data-data jarak antara dua simpul/titik berkisar dari 0 – 1.5 sehingga dalam penelitian ini nilai tersebut ditetapkan 999.

Telah dikatakan sebelumnya bahwa implementasi perekaman jalur terpendek di program dilakukan dengan membuat array $simpulPendahulu[Nsimpul]$ sebagai tempat penyimpanan lintasan dan nilainya diperbaharui pada saat jarak terpendek tersebut diperoleh setiap iterasinya. Gambar. 4 adalah penggalan *source code* program tersebut yang menjelaskan perhitungan jarak dan perekaman jalur terpendek.

```

void Dijkstra::hitungjarak(){
    inisialisasi();
    double minjarak = INFINITY;
    int belum; //penanda simpul yang belum dikunjungi
    int hitung = 0; int i;

    while(hitung < Nkota && !terkunjungi[kotaTujuan]) {
        belum = simpulTerdekat();
        kunjungi[belum] = true;
        for(i=0;i<Nkota;i++) {
            if(!terkunjungi[i] && (adjMatrix[belum][i]>0)) {
                if(jarak[i] > jarak[belum]+adjMatrix[belum][i]) {
                    jarak[i] = jarak[belum]+adjMatrix[belum][i];
                    simpulPendahulu[i] = belum;
                }
            }
        }
        hitung++;
    }
}
    
```

Gambar 4. Penggalan Kode Program Perekaman Jalur Terpendek dengan Algoritma Dijkstra.

Tahapan terakhir berupa pengujian program dengan data yang diperoleh dan validasi hasil dengan hasil peneliti lain dan solusi analitik (manual) dari metode yang digunakan. Perbaikan program terus dilakukan guna mendapatkan hasil yang benar, sesuai dengan hasil peneliti lain dan solusi analitik (manual).

III. HASIL DAN PEMBAHASAN

Berdasarkan data yang ada, Kelurahan Sanur memiliki dua titik aman evakuasi dan enam titik evakuasi. Kedua titik aman evakuasi tersebut adalah Puskesmas III Denpasar Selatan (simpul tiga, V3) dan SMK Negeri III Denpasar (simpul tiga puluh dua, V32). Sedangkan titik-titik evakuasi tersebut adalah Segara (V2), Shindu (V7), Karang (V18), Duyung (V19), Semawang (V25), dan Cemara (V28). Sehingga pencarian jalur evakuasi dimulai dari titik-titik evakuasi menuju titik aman evakuasi.

Hasil pencarian jalur evakuasi dengan jarak terpendek untuk setiap titik evakuasi dapat dilihat pada Gambar. 5 dan Gambar. 6. Hasil jalur evakuasi dari setiap titik evakuasi menuju Puskesmas III Denpasar Selatan disajikan di Tabel 1. Sedangkan jalur evakuasi dari setiap titik evakuasi menuju SMK Negeri III Denpasar diberikan di Tabel 2.

Jalur evakuasi menuju zona aman pertama, Puskesmas III Denpasar Selatan, memiliki jarak terpendek 0.74 (dengan skala dan satuan yang tertera di peta Gambar. 3) dengan jalur evakuasi dimulai dari titik evakuasi tujuh, Shindu (V7) – V6 – V5 – V4 – Puskesmas III Denpasar Selatan (V3). Jika diurutkan berdasarkan jarak jalur evakuasi menuju zona aman pertama dari yang terpendek diperoleh urutan Shindu (V7), Segara (V2),

Karang (V18), Duyung (V19), Semawang (V25), dan Cemara (V28).

Jalur evakuasi menuju zona aman kedua, SMK Negeri III Denpasar, memiliki jarak terpendek 1.7 (dengan skala dan satuan yang tertera di peta pada Gambar. 3) dengan jalur evakuasi dimulai dari titik evakuasi sembilan belas, Duyung (V19) – V35 – V21 – V20 – SMK Negeri III Denpasar (V32). Jika diurutkan berdasarkan jarak jalur evakuasi menuju zona aman kedua dari yang terpendek diperoleh urutan Duyung (V19), Semawang (V25), Cemara (V28), Karang (V18), Shindu V(7), dan Segara (V2).

Jalur Evakuasi dari U2 ke U3 2 1 0 3 Jarak : 0.9
Jalur Evakuasi dari U7 ke U3 7 6 5 4 3 Jarak : 0.74
Jalur Evakuasi dari U18 ke U3 18 34 14 9 8 5 4 3 Jarak : 1.94
Jalur Evakuasi dari U19 ke U3 19 35 21 20 30 29 3 Jarak : 3
Jalur Evakuasi dari U25 ke U3 25 24 23 22 21 20 30 29 3 Jarak : 3.1
Jalur Evakuasi dari U28 ke U3 28 36 24 23 22 21 20 30 29 3 Jarak : 3.5

Gambar 5. Keluaran Program Jalur Terpendek Tiap Titik Evakuasi ke Zona Aman Puskesmas III

Tabel 1. Jalur Evakuasi dari Titik Evakuasi ke Zona Aman Puskesmas III Denpasar Selatan

Zona Aman	Titik Evakuasi	Jalur Evakuasi	Jarak (Km)
Puskesmas III Denpasar Selatan (V3)	Segara (V2)	V2 V1 V0	0,9
	Shindu (V7)	V7 V6 V5	0,74
		V4 V3	
	Karang (V18)	V18 V34	1,94
		V14 V9 V8	
		V5 V4 V3	
Duyung (19)	V19 V35	3	
	V21 V20		
	V30 V29		
	V3		
Semawang (V25)	V25 V24	3,1	
	V23 V22		
	V21 V20		
	V30 V29		
	V3		
Cemara (V28)	V28 V36	3,5	
	V24 V23		
	V22 V21		
	V20 V30		
	V29 V3		

Jalur Evakuasi dari U2 ke U32 2 1 0 3 29 31 32 Jarak : 3
Jalur Evakuasi dari U7 ke U32 7 6 5 4 3 29 31 32 Jarak : 2.84
Jalur Evakuasi dari U18 ke U32 18 34 17 16 15 20 30 32 Jarak : 2.6
Jalur Evakuasi dari U19 ke U32 19 35 21 20 30 32 Jarak : 1.7
Jalur Evakuasi dari U25 ke U32 25 24 23 22 21 20 30 32 Jarak : 1.8
Jalur Evakuasi dari U28 ke U32 28 36 27 26 32 Jarak : 1.9

Gambar 6. Keluaran Program Jalur Terpendek Tiap Titik Evakuasi ke Zona Aman SMK Negeri III Denpasar

Tabel 2. Jalur Evakuasi dari Titik Evakuasi ke Zona Aman SMK Negeri III Denpasar

Zona Aman	Titik Evakuasi	Jalur Evakuasi	Jarak (Km)
SMK Negeri III Denpasar (V32)	Segara (V2)	V2 V1 V0 V29 V31 V32	3
	Shindu (V7)	V7 V6 V5 V4 V3 V29 V31 V32	2,84
	Karang (V18)	V18 V34 V17 V16 V15 V20 V30 V32	2,6
	Duyung (19)	V19 V35 V21 V20 V30 V32	1,7
	Semawang (V25)	V25 V24 V23 V22 V21 V20 V30 V32	1,8
	Cemara (V28)	V28 V36 V27 V26 V32	1,9

Hasil yang dipaparkan dalam artikel ini terdapat sedikit perbedaan dengan refensi [4]. Semua hasil keluaran program jalur evakuasi sama dengan hasil yang diperoleh Sani dkk, yang disajikan di [4] kecuali untuk jalur evakuasi dari titik evakuasi Duyung (19) menuju zonan aman kedua SMK Negeri III Denpasar (V32) dalam hal ini keluaran program yang penulis buat adalah **Duyung (19) – V35 – V21 – V20 – V30 – SMK Negeri III Denpasar (V32)** dengan **jarak 1,7**. Sedangkan hasil yang dipaparkan oleh Sani dkk. didalam artikelnya [4] adalah **Duyung (19) – V35 – V21 – V20 – V30 – SMK Negeri III Denpasar (V32)** dengan **jarak 2**. Terlihat jelas terdapat perbedaan jarak yang dihasilkan namun dengan jalur yang sama.

Perhatikan Gambar. 3, akan dilakukan perhitungan manual untuk menentukan jarak terpendek dari titik evakuasi Duyung (V19) – zona aman SMK Negeri III Denpasar (V32). Jalur terpendek dari titik evakuasi tersebut adalah **V19 – V35 – V21 – V20 – V30 – V32**. Perhitungan jarak dilakukan dengan menjumlahkan jarak

antar simpul-simpul tersebut. Adapun jarak **V19 – V35 = 0,3**, **V35 – V21 = 0,5**, **V21 – V20 = 0,1**, **V20 – V30 = 0,3**, **V30 – V32 = 0,5**. Adapun jarak total jalur tersebut adalah **Jarak = 0,3 + 0,5 + 0,1 + 0,3 + 0,5 = 1,7**. Hal ini membuktikan bahwa keluaran program yang dibuat adalah benar sesuai dengan perhitungan manual algoritma Dijkstra (solusi analitik) dan untuk jalur-jalur lainnya keluaran program juga sesuai dengan referensi (Sani dkk, 2013) yang menggunakan algoritma Floyd Warshall.

Jika melihat hasil yang diperoleh, perhatikan Gambar. 5 dan Gambar. 6, jarak dari titik evakuasi yang sama ke masing-masing zona aman evakuasi adalah berbeda, misalnya jarak V2 – V3 (jarak 0.9) dan V2 – V32 (jarak 3). Tidak hanya titik evakuasi tersebut yang memiliki perbedaan jarak menuju zona aman namun titik evakuasi yang lainnya juga. Jika dirinci jarak titik-titik evakuasi lainnya menuju zona aman evakuasi adalah V7 – V3 (jarak 0,74), V7 – V32 (jarak 2,84), V18 – V3 (jarak 1,94), V18 – V32 (jarak 2,6), V19 – V3 (jarak 3), V19 – V32 (jarak 1,7), V25 – V3 (jarak 3,1), V25 – V32 (jarak 1,8), V28 – V3 (jarak 3,5), dan V28 – V32 (jarak 1,9).

Melihat hal tersebut, jalur evakuasi tsunami pada Kelurahan Sanur dapat dibuatkan berdasarkan lokazi zona aman terdekat. Dalam hal ini, berdasarkan keluaran program yang dibuat penulis menyarankan bahwa jalur evakuasi tsunami terbaik untuk Kelurahan Sanur adalah sebagaimana yang tertera di Tabel 3.

Tabel 3. Jalur Evakuasi Terbaik Tiap Titik Evakuasi ke Zona Aman Kelurahan Sanur Bali

Zona Aman	Titik Evakuasi	Jalur Evakuasi	Jarak (Km)
Puskesmas III Denpasar Selatan (V3)	Segara (V2)	V2 V1 V0 V3	0,9
	Shindu (V7)	V7 V6 V5 V4 V3	0,74
	Karang (V18)	V18 V34 V14 V9 V8 V5 V4 V3	1,94
SMK Negeri III Denpasar (V32)	Duyung (19)	V19 V35 V21 V20 V30 V32	1,7
	Semawang (V25)	V25 V24 V23 V22 V21 V20 V30 V32	1,8
	Cemara (V28)	V28 V36 V27 V26 V32	1,9

Tabel 3 memaparkan pembagian jalur evakuasi berdasarkan jarak terdekat dari titik-titik evakuasi dengan zona aman evakuasi. Titik-titik evakuasi Segara (V2), Shindu (V7), dan Karang (V18) diarahkan menuju zona aman pertama Puskesmas III Denpasar Selatan (V3) dengan rute masing-masing seperti yang tertera di tabel tersebut. Tiga titik evakuasi yang lainnya Duyung (19), Semawang (V25), dan Cemara (V28) diarahkan menuju zona aman yang kedua SMK Negeri III Denpasar (V32) dengan jalur evakuasi yang disajikan pada Tabel 3. Dengan pembagian tersebut, diharapkan evakuasi tsunami

akan lebih cepat dan efisien prosesnya jika dibandingkan dengan jalur-jalur evakuasi yang diberikan di Tabel 1 dan Tabel 2.

IV. PENUTUP

Kesimpulan

Pemaparan mengenai pemanfaatan algoritma Dijkstra untuk pencarian jalur terbaik evakuasi tsunami di Kelurahan Sanur Bali telah diberikan. Hasil keluaran program menunjukkan kesesuaian yang baik dengan referensi dan solusi manual (analitik) metode tersebut. Jalur evakuasi dibagi berdasarkan letak (jarak) titik-titik evakuasi dengan zona aman terdekat dan diperoleh titik evakuasi Segara, Shindu, dan Karang diarahkan menuju zona aman Puskesmas III Denpasar Selatan. Sedangkan titik evakuasi Duyung, Semawang, dan Cemara diarahkan menuju zonan aman lainnya yaitu SMK Negeri III Denpasar. Jalur masing-masing titik evakuasi diberikan di Tabel 3.

Saran

Penyusunan jalur terbaik evakuasi tsunami Kelurahan Sanur Bali yang telah dipaparkan memungkinkan untuk dilakukan karena tidak ada batasan kapasitas maksimum daya tampung zona aman evakuasi dan titik-titik evakuasi serta tidak mempertimbangkan jumlah penduduk pada masing-masing daerah. Kapasitas maksimum dan jumlah penduduk ini perlu untuk diperhitungkan sehingga penyusunan jalur terbaik evakuasi tsunami akan lebih akurat.

V. DAFTAR PUSTAKA

- BMKG. "Buku Pedoman Pelayanan Peringatan Dini Tsunami InaTEWS – Edisi Kedua". (2012)
- BPPD-Bali. "Peta Evakuasi Tsunami Bali". Diambil dari <http://balisafety.baliprov.go.id/artikel/peta-evakuasi-tsunami-bali.html>. 2017, Agustus 28. Diakses 3 Februari 2018.
- Tsunami-Kit. "Tsunami Evacuation Map Bali, Denpasar." Diambil dari http://www.gitews.org/tsunami-kit/en/id_tsunami_evacuation_map_denpasar.htm 1. (2010). Diakses 3 Februari 2018.
- Sani, Ajeng Fitrah, Ni Ketut Tari Tastrawati, and I. Made Eka Dwipayana. "Algoritma Floyd Warshall Untuk Menentukan Jalur Terpendek Evakuasi Tsunami di Kelurahan Sanur." *E-Jurnal Matematika* 2.1 (2013): 1-5.
- Anggria, Siska, Syafwan, Mahdhivan, dan Efendi. "ACE 3-001 Pemodelan Optimasi Evakuasi Tsunami di Kota Padang." (2016). pp. 1 – 12.
- Iryanto, Iryanto, and Eka Ismantohadi. "Optimasi Pemilihan Barang Dagangan bagi Pedagang Keliling dengan Algoritma Genetika." *JTT (Jurnal Teknologi Terapan)* 3.1 (2017): 24-28.
- Yan, Melissa. "Dijkstra's algorithm." Massachusetts Institute of Technology. Regexstr (2014). Diakses 3 Februari 2018.
- Deng, Yong, Yuxin Chen, Yajuan Zhang, and Sankaran Mahadevan. "Fuzzy Dijkstra algorithm for shortest path problem under uncertain environment." *Applied Soft Computing* 12, no. 3 (2012): 1231-1237.
- Cormen, Thomas H., Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. "Introduction to algorithms third edition." (2009).
- Dijkstra, Edsger W. "A note on two problems in connexion with graphs." *Numerische mathematik* 1.1 (1959): 269-271.
- Dewi, L. N. "Perancangan Jalur Evakuasi Kebakaran Fakultas Teknik UNS Sesuai SNI 03-1746-2000 Dan ISO 7010 Dengan Metode Algoritma Dijkstra (Studi Kasus: Gedung 1 Fakultas Teknik UNS)" (Skripsi, Universitas Sebelas Maret). (2018).
- Pramudya, R. A., & Subiyanto, S. "Penggunaan Algoritma Dijkstra Dalam Perencanaan Rute Evakuasi Bencana Longsor Di Kota Semarang." *Geoplanning: Journal of Geomatics and Planning*, 2(2) (2015): 93-102.
- Hapsari, R. A., & Budi Hartono, S. T. "Optimasi Penentuan Jalur Evakuasi Bencana Alam Letusan Gunung Merapi Berdasarkan Kehandalan Jalur." (Skripsi, Universitas Gadjah Mada). (2016)
- Herdiyatomoko, H. F. "Seleksi Atribut Dengan Teori Rough Set Untuk Simulasi Evakuasi Bencana Kebakaran Hutan Menggunakan Algoritma Dijkstra." (Tesis, Universitas Gadjah Mada). (2015).
- Zulfa, A. M., & Saputra, R. "Aplikasi Penentuan Rute Evakuasi Bencana Tsunami Kota Padang Menggunakan Algoritma A-Star Berbasis Hybrid Application." (Skripsi, Universitas Diponegoro). (2015).