

Pencarian Rute Terdekat Lokasi Evakuasi Tsunami Dengan Metode Pencarian Dijkstra Dan Pendekatan MADM

¹⁾ M Ramdhan Syahputra

Program Studi Teknik Informatika Universitas Muhammadiyah Sukabumi
Jl. R. Syamsudin, S.H. No. 50, Jawa Barat, Indonesia
E-Mail: ramdhanjr690@ummi.ac.id

²⁾ Asriyanik

Program Studi Teknik Informatika Universitas Muhammadiyah Sukabumi
Jl. R. Syamsudin, S.H. No. 50, Jawa Barat, Indonesia
E-Mail: asriyanik263@ummi.ac.id

ABSTRACT

Tsunami is one of the natural disasters that threatens Indonesia, where one of the major factors is caused by an earthquake that occurred on the seabed and has been recorded to have claimed many lives. Behind this incident, there are problems faced by the society when evacuating themselves, one of which is the ignorance of access to evacuation routes, so that people fail to take advantage of their time to evacuate. In this study applying *Dijkstra* method with *SAW* and *Haversine* approaches at Pelabuhan Ratu, Sukabumi Regency, the final weight results are grouped based on 3 main parameters, namely road width, distance traveled and estimated arrival time, also at the end of this study resulted in an Android application that implements the model that has been created.

Keyword : Dijkstra, Saw, Haversine, Nearest route, Tsunami

PENDAHULUAN

Indonesia merupakan sebuah negara yang memiliki kekayaan alam yang begitu banyak, salah satunya adalah banyaknya pulau yang tersebar di Indonesia, terhitung 17.504 pulau yang tersebar, dimana sebagian besar pulau Indonesia dikelilingi oleh laut yang membuat Indonesia memiliki banyak pantai-pantai yang begitu indah, tidak heran Indonesia menjadi salah satu incaran wisatawan asing maupun lokal untuk berlibur.

Akan tetapi dibalik itu semua Indonesia merupakan salah satu negara yang terkena dampak bencana alam tsunami, terdapat beberapa faktor yang membuat Indonesia menjadi potensial terhadap bencana alam, yaitu letaknya yang berada diantara lempengan besar dunia yakni Lempengan Benua Eurasia, Lempeng Samudera Pasifik dan Lempeng Benua Samudera Indo-Australia [1], sehingga apabila sewaktu-waktu lempengan-lempengan tersebut terjadi pergerakan maka kemungkinan besar Indonesia akan terkena dampak dari efek pergerakan tersebut. Berdasarkan UU Nomor 24 Tahun 2007 bencana alam merupakan sebuah kejadian atau peristiwa yang diakibatkan oleh alam berupa, gempa bumi, gunung meletus, banjir, tsunami, tanah longsor dan angin topan yang menimbulkan kerugian di darat, udara maupun laut.

Adapun bencana alam tsunami umumnya diakibatkan oleh pergerakan tanah pada dasar laut

secara vertikal dan gunung berapi aktif [2], sebagian besar tsunami yang terjadi di Indonesia banyaknya diakibatkan oleh gempa bumi yang terjadi di dasar laut dengan angka persentase sebesar 90%, gunung merapi aktif yang meletus dibawah laut sebesar 9% dan tanah longsor sebesar 1%, data ini diambil berdasarkan RBI (Risiko Bencana Indonesia) [3], bencana tsunami yang terjadi di Indonesia telah banyak merenggut korban jiwa antara lain adalah kejadian yang terjadi di Palu-Donggala pada tanggal 28 september 2018 dengan jumlah 3.879 korban jiwa dan kawasan Selat Sunda yang terjadi di Banten dan Lampung pada tanggal 22 Desember 2018 dengan jumlah 126 korban jiwa dan lebih dari 4000 jiwa yang tercatat luka-luka.

Melihat riwayat kejadian tsunami yang telah terjadi sebelumnya Pelabuhan Ratu Kabupaten Sukabumi merupakan salah satu wilayah yang memiliki potensi akan tsunami pada Selat Sunda yang berada pada zona *megathrust* Jawa Barat. Oleh karena itu pentingnya untuk dilakukan kajian mengenai evakuasi tsunami, yakni sebuah tindakan yang harus dilakukan untuk menjauhkan masyarakat dari ancaman tsunami (evakuasi) untuk meminimalisir jumlah kerugian yang akan menimpa.

Namun pada saat masyarakat saat ingin melakukan evakuasi terdapat beberapa masalah yang menimpa, berdasarkan data pada "Keterbatasan dan tantangan sistem Peringatan dini Studi Kasus Tsunami Palu-Donggala 28 September

2. Tahap Pengolahan Data SAW

Tahap ini melakukan analisis terhadap data yang telah didapat untuk diolah dengan metode SAW yang memberikan hasil akhir berupa bobot yang akan dihitung dengan algoritma *Dijkstra*, adapun data yang menjadi kriteria pada penelitian ini adalah lebar jalan, jarak tempuh dan estimasi waktu tiba.

3. Tahap Pengolahan Data Haversine

Tahap pengolahan data dengan metode haversine pada penelitian ini adalah untuk mencari jarak pada masing-masing *vertex* yang saling terhubung dan mencari jarak pada 3 *vertex* lokasi evakuasi guna menjadikan lokasi akhir yang akan dikunjungi oleh pengguna.

4. Tahap Evaluasi Model

Tahap ini melakukan evaluasi terhadap model yang telah dibuat dengan menghitung semua kemungkinan jalan yang akan dilalui menuju lokasi evakuasi yang dipilih dan memastikan nilai akhir dari model adalah yang terkecil dari semua kemungkinan jalan yang telah dihitung.

5. Tahap Implementasi Model

Pada tahap terakhir ini, peneliti menerapkan model yang telah dibuat kedalam aplikasi Android dengan menggunakan Bahasa pemrograman Kotlin dan Android Studio sebagai text editor yang digunakan.

HASIL DAN PEMBAHASAN

Langkah awal adalah mencari data-data jalan yang akan digunakan dan persimpangan jalan yang terhubung, terdapat 22 jalan raya yang diambil pada penelitian ini yaitu :

Tabel 1 Jalan raya Pelabuhan Ratu Kabupaten Sukabumi

No	Nama Jalan
1.	Jl. Nasional III
2.	Jl Raya Pangsor
3.	Jl. Pasar Pelabuhanratu
4.	Jl. Gumelar
5.	Jl. Sirgalih
6.	Jl. Kaum Raya
7.	Jl. Pemda
8.	Jl. Kenari 2
9.	Jl. Siliwangi
10.	Jl. Soka
11.	Jl. Simantang
12.	Jl. Empang Raya 1
13.	Jl. Cagak
14.	Jl. Bhayangkara
15.	Jl. Raya Cisolok
16.	Jl. Pelita Cipatuguran
17.	Jl. Majelis
18.	Jl. Dewi Sartika
19.	Jl. Penegak I

No	Nama Jalan
20.	Jl. Penegak II
21.	Jl. Raya Cangehgar
22.	Jl. Otto Iskandar Dinata

Adapun berikut adalah data vertex dan koordinat yang digunakan.

Tabel 2 *Vertex-vertex* lokasi persimpangan

No	Vertex	Koordinat
1	V1	-6.984021136912581, 106.54242422961052
2	V2	-6.985831502332914, 106.54349711310161
3	V3	-6.985305087423851, 106.54417958070525
4	V4	-6.984990935926289, 106.5443190555591
5	V5	-6.9854807991863845, 106.54517736235199
6	V6	-6.984783276345959, 106.54517199793452
7	V7	-6.984304061577873, 106.54500033657595
8	V8	-6.983510693810596, 106.54458191201958
9	V9	-6.9861574453642605, 106.5450917651763
10	V10	-6.986129946893477, 106.54662894605842
11	V11	-6.986779546479836, 106.5465699374664
12	V12	-6.986070309301696, 106.54736519601954
13	V13	-6.985859870053963, 106.54894852885835
14	V14	-6.9848524468450055, 106.54776215697427
15	V15	-6.985103183489061, 106.54959810117344
16	V16	-6.986013507304345, 106.54990031008936
17	V17	-6.986569929786441, 106.55078166445675
18	V18	-6.986900053983113, 106.55167802974897
19	V19	-6.987966999908298, 106.55059672077293
20	V20	-6.988128466288221, 106.55136862865379
21	V21	-6.988733173226074, 106.5505967207707
22	V22	-6.988815731810449, 106.55118485786983
23	V23	-6.988311217196974, 106.55209333318624
24	V24	-6.98893734804612, 106.5519752669342

No	Vertex	Koordinat
25	V25	-6.98813067673753, 106.55251975536034
26	V26	-6.988362295374446, 106.55247684001687
27	V27	-6.9889932559066255, 106.55234004735958
28	V28	-6.988979944511787, 106.55286576031702
29	V29	-6.9889613085610405, 106.55329223155792
30	V30	-6.9874471188206675, 106.54435427841207
31	V31	-6.987827388051684, 106.54539361967763
32	V32	-6.989905685753024, 106.5437435705893
33	V33	-6.990886473401127, 106.54308924075019
34	V34	-6.991805136344499, 106.54316510509037
35	V35	-6.9917938413194225, 106.5439161619173
36	V36	-6.991703481108996, 106.54490619137098
37	V37	-6.989933923453875, 106.5464614100786
38	V38	-6.99038196102259, 106.54654486083714
39	V39	-6.990216300115652, 106.54712142973551
40	V40	-6.989922628376356, 106.54746661241862
41	V41	-6.989161523087814, 106.5490314513003
42	V42	-6.988458681175001, 106.54911728198721
43	V43	-6.990116490632656, 106.54913012117592
44	V44	-6.990285916590598, 106.5500746320544
45	V45	-6.990372512055328, 106.55049947228218
46	V46	-6.990406397232819, 106.55089775999345
47	V47	-6.99067371354682, 106.5509508650216
48	V48	-6.990737718843625, 106.55118604444802
49	V49	-6.99118199062832, 106.55136053239772
50	V50	-6.989992244899929, 106.55167157613413
51	V51	-6.9903122717379516, 106.55279057495824
52	V52	-6.989984714853351, 106.55260470735966

No	Vertex	Koordinat
53	V53	-6.989717398145217, 106.55253263624999
54	V54	-6.989566797115537, 106.55328369307695
55	V55	-6.989043458151655, 106.55355680466168
56	V56	-6.986869113389685, 106.54736727736591
57	V57	-6.987151190888207, 106.54743945208287
58	V58	-6.986998958608556, 106.54767853083278
59	V59	-6.987446700466026, 106.54811609005434
60	V60	-6.988315318446525, 106.54811157913454
61	V61	-6.987747593094858, 106.55061465577771

Selanjutnya adalah mencari nilai perhitungan dari 3 parameter yang berbeda yaitu lebar jalan, jarak tempuh dan estimasi waktu tiba, dimana data pada masing-masing parameter tersebut didapatkan dengan observasi secara tidak langsung melalui google maps dan penggunaan metode *haversine* dalam pencarian jarak antar *vertex*-nya.

Setelahnya maka dilakukan analisa terhadap jenis kriteria yang akan digunakan terhadap 3 parameter tersebut yang ditampilkan pada Tabel 3 berikut.

Tabel 3 Jenis kriteria lebar jalan, jarak tempuh dan estimasi waktu tiba

Kode	Kriteria	Atribut	Bobot
C1	Lebar Jalan	Benefit	40
C2	Jarak Tempuh	Cost	30
C3	Estimasi Waktu Tiba	Cost	30

Perhatikan pada Tabel 3 dimana terdapat 2 atribut yaitu *benefit* dan *cost* yang artinya apabila nilai bobot pada atribut lebih besar maka semakin besar keuntungan untuk pengguna dan adapun *cost* adalah sebaliknya.

Setelah diketahui data kriteria pada Tabel 3 maka langkah berikutnya adalah melakukan normalisasi terhadap jenis kriteria dengan menggunakan persamaan berikut.

$$r_{ij} = \begin{cases} \frac{X_{ij}}{\text{Max}_{ij} X_{ij}}, & \text{jika } j \text{ adalah atribut keuntungan} \\ \frac{\text{Min}_i X_{ij}}{X_{ij}}, & \text{jika } j \text{ adalah atribut biaya (cost)} \end{cases}$$

Dimana,
 Rij : rating kinerja ternormalisasi
 Xij : nilai standar kriteria pada baris ke-i, kolom ke -j

Max Xij : nilai maksimum dari setiap baris dan kolom

Min Xij : nilai minimum dari setiap baris dan kolom

A : alternatif

C : kriteria

m : kriteria ke-m

n : kriteria ke-n

benefit : nilai kriteria yang menguntungkan pada pengguna ketika nilai semakin tinggi.

perhitungan dilakukan berdasarkan atribut benefit dan cost, berikut adalah 3 sampel data hasil perhitungan terhadap jenis kriteria.

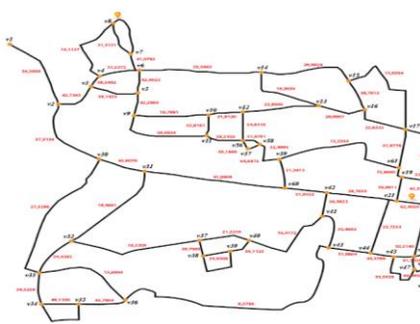
Tabel 4 Sampel hasil perhitungan terhadap jenis kriteria

No	Vertex	Kode Kriteria		
		C1	C2	C3
1	V1 - V2	0,7494	0,1048	0,1048
2	V2 - V3	0,6836	0,2566	0,2566
3	V2 - V30	0,7570	0,1206	0,1206

Tahap selanjutnya adalah melakukan perangkaian terhadap hasil perhitungan jenis kriteria Tabel 4 dengan Tabel 3, dimana akan dilakukan perkalian terhadap semua atribut yang telah diproses sebelumnya pada tiap-tiap kriteria, sebagai contoh perhitungan pada bobot vertex V2-V3.

$$\begin{aligned}
 V_s \text{ (Preferensi)} &= C1(0,6836) * \text{bobot } C1(40) + \\
 &C2(0,2566) * \text{bobot } C2(30) + \\
 &C3(0,2566) * \text{bobot } C3(30) = \\
 &\mathbf{42,7383}
 \end{aligned}$$

sehingga bobot akhir untuk vertex V1 menuju V2 adalah 42,7383. Bobot akhir inilah yang akan digunakan untuk proses selanjutnya yakni mencari lokasi terdekat dengan algoritma Dijkstra. Sehingga ditemukan sebaran bobot yang telah dihitung sebagai berikut.



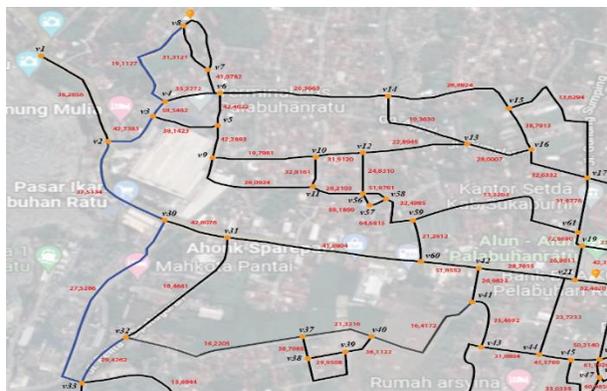
Gambar 3 Sebaran bobot akhir pada peta

Sebelum melakukan pencarian dengan algoritma Dijkstra dilakukan pencarian lokasi awal dan lokasi tujuan (evakuasi) dengan metode haversine, lokasi awal diambil berdasarkan posisi terakhir pengguna dengan memanfaatkan based location service pada pustaka google maps SDK Android lalu dilakukan perbandingan terhadap 3 lokasi tujuan evakuasi yaitu, Jl. Simagilih, Pelabuhanratu yang ditandai dengan V8 pada Gambar 2, Jl. Siliwangi, Pelabuhanratu No.15 yang ditandai dengan V21 pada Gambar 2 dan Jl. Bhayangkara 15, Pelabuhanratu yang ditandai dengan V29 pada Gambar 2.

Titik awal pada penelitian ini diambil pada vertex V32 yang berada di Jl. Pelita Cipatuguran dan lokasi tujuan evakuasi yang terpilih dari 3 lokasi sebelumnya adalah vertex V8. Setelah diketahui titik awal dan titik tujuan maka langkah selanjutnya adalah melakukan perhitungan terhadap semua kemungkinan vertex dengan algoritma Dijkstra. Adapun langkah-langkah untuk menggunakan algoritma Dijkstra ini adalah sebagai berikut.

1. Inialisasi titik awal dan titik tujuan.
2. Inialisasi bobot antar titik.
3. Beri label sementara pada vertex-vertex yang terhubung dengan vertex saat ini.
4. Menentukan jarak vertex paling minimum dari label sementara.
5. Beri label permanen pada titik minimum yang terpilih tersebut.
6. Menghapus titik minimum yang terpilih pada label sementara.
7. Menyimpan titik minimum yang terpilih kedalam label permanen.
8. Membandingkan nilai titik paling minimum pada label permanen.
9. Jika tidak terdapat nilai titik minimum pada label permanen maka ulangi pada langkah ke-5.
10. Lintasan terpendek ditemukan.

Sehingga diperoleh 34 proses iterasi dari perhitungan titik awal V32 menuju V8 yang menghasilkan rute terdekat yaitu V32 – V33 – V30 – V2 – V3 – V4 – V8, dan apabila digambarkan melalui graf maka akan seperti pada gambar berikut.



Gambar 4 Ilustrasi graf hasil perhitungan dengan Dijkstra

Setelah ditemukan hasil rute terdekat dengan algoritma *Dijkstra* langkah selanjutnya adalah melakukan evaluasi model terhadap hasil yang telah ditemukan dengan beberapa kemungkinan rute yang menuju lokasi evakuasi vertex V_8 , berikut adalah beberapa kemungkinan rute yang menuju lokasi evakuasi V_8 .

Tabel 5 Kemungkinan rute yang menuju lokasi evakuasi

No	Rute	Hasil
1.	V32 – V33 – V30 – V2 – V3 – V4 – V8	241,86
2.	V32 – V33 – V30 – V2 – V3 – V4 – V6 – V7 – V8	303,37
3.	V32 – V33 – V30 – V2 – V3 – V5 – V6 – V7 – V8	290,20
4.	V32 – V31 – V30 – V2 – V3 – V4 – V8	280,46
5.	V32 – V31 – V30 – V2 – V3 – V4 – V6 – V7 – V8	249,14
6.	V32 – V31 – V30 – V2 – V3 – V5 – V6 – V7 – V8	294,52
7.	V32 – V31 – V60 – V59 – V58 – V56 – V11 – V9 – V5 – V6 – V7 – V8	377,52
8.	V32 – V31 – V60 – V59 – V58 – V57 – V56 – V11 – V9 – V5 – V6 – V7 – V8	449,42
9.	V32 – V31 – V60 – V59 – V58 – V56 – V12 – V10 – V9 – V5 – V6 – V7 – V8	399,76
10.	V32 – V37 – V40 – V41 – V42 – V60 – V59 – V58 – V56 – V11 – V9 – V5 – V6 – V7 – V8	449,97

Dari 10 kemungkinan rute yang menuju lokasi evakuasi vertex V_8 , pada Tabel 5 rute terkecil terdapat pada no 1, dengan begitu hasil dari perhitungan dengan algoritma *Dijkstra* telah berhasil diterapkan.

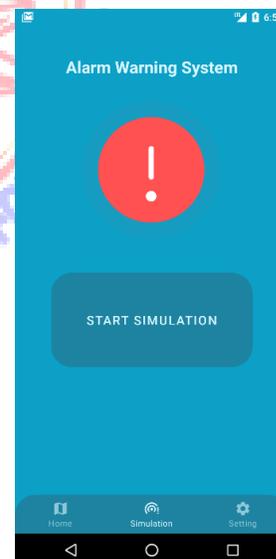
Langkah terakhir adalah menerapkan model tersebut kedalam sebuah aplikasi Android, hasil dari model yang telah dibuat diterapkan didalam fitur *Alarm Warning System*, dimana fitur ini akan berfungsi seperti alarm yang dapat memberitahukan pengguna ketika adanya informasi terkait tsunami pada Pelabuhan Ratu Kabupaten Sukabumi dan terdapat tombol darurat yang dapat pengguna gunakan untuk mencari lokasi evakuasi terdekat dan berdasarkan rute yang telah dihitung dengan model yang telah dibuat, namun pada penelitian ini informasi tersebut digunakan hanya sebagai simulasi pada aplikasi dengan hasil sebagai berikut.

1) Tampilan halaman beranda aplikasi

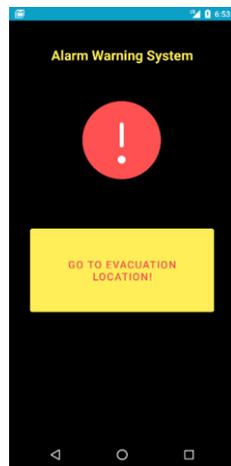


Gambar 5 Halaman Beranda Aplikasi Seakidul

2) Tampilan halaman fitur simulasi dan rute terdekat



Gambar 6 Halaman Simulasi Seakidul

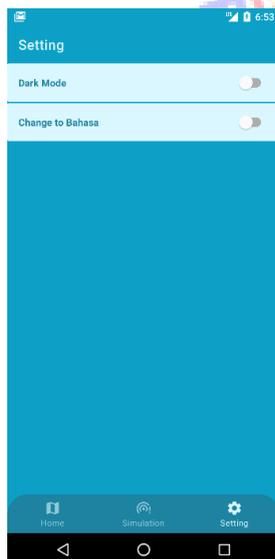


Gambar 7 Tampilan fitur Alarm Warning System Seakidul



Gambar 8 Tampilan fitur rute terdekat Seakidul

3) Halaman Pengaturan



Gambar 9 Halaman Pengaturan Seakidul

KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan maka dapat disimpulkan sebagai berikut.

1. Penelitian ini telah berhasil menerapkan 2 metode pendukung yaitu SAW (*Simple Additive Weighting*) dan *Haversine* dengan menghasilkan bobot yang digunakan untuk melakukan perhitungan dengan algoritma *Dijkstra*.
2. Penelitian ini telah berhasil menghitung semua kemungkinan vertex yang terhubung menuju lokasi tujuan evakuasi yaitu Jl.Simagilih Pelabuhanratu (V8) dari lokasi awal vertex yaitu V32 yang berlokasi di Jl. Pelita Cipatuguran dengan hasil rute terdekat yaitu V32 – V33 – V30 – V2- V3 – V4 – V8.
3. Penelitian ini telah berhasil menerapkan model yang telah dibuat kedalam sebuah aplikasi Android yang memiliki fitur *Alarm Warning System* yang akan memberitahukan dan mengarahkan pengguna menuju lokasi evakuasi terdekat ketika adanya informasi tsunami di Pelabuhan Ratu Kabupaten Sukabumi.

Dengan berhasilnya penerapan model algoritma *Dijkstra* kedalam sebuah aplikasi Android diharapkan dapat membantu masyarakat Pelabuhan Ratu Kabupaten Sukabumi dalam upaya evakuasi bencana alam tsunami.

Adapun saran untuk penelitian selanjutnya yaitu adanya parameter jumlah penduduk pada lokasi evakuasi, situasi jalan, kondisi kemacetan pada jalur yang akan dipilih dan mengintegrasikan data informasi terkait tsunami dengan badan resmi yang bersangkutan seperti BPBD, BNPB, BMKG agar data informasi yang diterima lebih akurat.

DAFTAR PUSTAKA

- [1] R. K. Apriyadi and R. Amelia, "Tingkat Pengetahuan Kesiapsiagaan Resiko Bencana Tsunami disaat Pandemi Covid-19," *PENDIPA J. Sci. Educ.*, vol. 5, no. 1, pp. 56–62, 2020, doi: 10.33369/pendipa.5.1.56-62.
- [2] E. Ismantohadi and I. Iryanto, "Penerapan Algoritma Dijkstra untuk Penentuan Jalur Terbaik Evakuasi Tsunami – Studi Kasus: Kelurahan Sanur Bali," *JTT (Jurnal Teknol. Ter.)*, vol. 4, no. 2, pp. 72–78, 2018, doi: 10.31884/jtt.v4i2.79.
- [3] BNPB, "Risiko bencana indonesia," 2016.
- [4] Z. Zainuddin, A. L. Arda, and A. Z. Nusri, "Sistem Peringatan Dini Banjir," *Inspir. J. Teknol. Inf. dan Komun.*, vol. 9, no. 2, 2019, doi: 10.35585/inspir.v9i2.2501.
- [5] Surianto and M. H. Adiya, "Perbandingan

- Algoritma Dijkstra dan Algoritma Floyd-Warshall Dalam Pencarian Lokasi Kuliner,” vol. 2, no. 3, pp. 128–133, 2020.
- [6] M. K. Siswandi, “Pemilihan Lokasi Sentra Industri Batu Klawing Purbalingga Dengan Metode AHP,” *Din. Rekayasa*, vol. 12, no. 1, pp. 39–44, 2016.
- [7] U. Kultsum, M. A. Z. Fuad, and A. Isdianto, “Desain Jalur Evakuasi Tsunami di Daerah Pelabuhan Ratu Kabupaten Sukabumi Menggunakan Sistem Informasi Geografis Design of Tsunami Evacuation Routes in Pelabuhan Ratu Area of Sukabumi District using Geographic Information System,” *Semin. Nas. Peninderaan Jauh ke-4*, vol. 24, no. 24, pp. 291–300, 2017, [Online]. Available: <http://repository.ub.ac.id/id/eprint/6423>.

