



Penerapan Algoritma Dijkstra dengan metode SAW dan Haversine Pada Pencarian Rute Terdekat Menemukan Titik Pemberhentian Angkot Kota Sukabumi

Zaldi Arif Mulkan*, Iwan Rizal Setiawan, Fathia Frazna

Fakultas Sains Dan Teknologi, Teknik Informatika, Universitas Muhammadiyah, Sukabumi

Jl. R. Syamsudin, S.H. No. 50, Cikole, Kec. Cikole, Kota Sukabumi, Jawa Barat, Indonesia

Email: ^{1,*}zaldospens@gmail.com, ²myfrank5150@gmail.com, ³fathiafrazna@gmail.com

Email Penulis Korespondensi: zaldospens@gmail.com

Submitted: 16/06/2023; Accepted: 30/07/2023; Published: 31/07/2023

Abstrak—Angkot (Angkutan Kota) di Kota Sukabumi adalah sistem transportasi umum yang populer dan menjadi salah satu sarana transportasi utama bagi warga lokal dan pengunjung untuk bergerak di dalam kota dengan begitu banyak jalur masing-masing yang dilewati angkot itu sendiri, 19 trayek jalur yang begitu banyak memunculkan permasalahan pada masyarakat dan pendatang luar kota Sukabumi dalam menemukan titik pemberhentian angkot sehingga memperlambat waktu dan tidak efisien. Oleh karena itu untuk mempermudah dalam mencari titik pemberhentian angkot di pilihlah algoritma Dijkstra. Algoritma dijkstra yaitu algoritma pencarian rute terpendek dengan mengandalkan bobot nilai terkecil dan menghitung semua simpul tetangga untuk menemukan nilai yang paling kecil. Peneliti menggunakan 2 Metode yang digunakan yaitu metode SAW dan Haversine. Metode SAW (Simple Additive Weighting Method). Metode ini untuk melakukan penjumlahan terbobot berdasarkan 3 kriteria lebar jalan, jarak tempuh dan estimasi waktu lalu melakukan normalisasi data hingga perangkingan. metode haversine yaitu untuk menghitung jarak berdasarkan lattitude dan lontitude. Melakukan inisiasi dari titik awal ke titik tujuan dengan graf menggunakan algoritma Dijkstra sehingga menghasilkan titik fokus 30 iterasi yang di hasilkan oleh algoritma Dijkstra, lalu melakukan evaluasi model dengan menjangkau jalur lain sampai ke titik tujuan. Didapatkan hasil dalam pencarian rute terdekat dengan menggunakan dua metode SAW dan Haversine, setelah menghitung semua jalur - jalur yang dilewati dari titik awal V117 (Jalan Bayangkara) - V82 (Jalan R.E Martadinata) didapatkan hasil rute terpendek yang didapat yaitu V117 - V118 - V131 - V130 - V132 - V133 - V134 - V138 - V137 - V110 - V82. Mencari rute terdekat dari titik awal menuju lokasi tujuan yang di tentukan oleh peneliti dengan hasil akhir berupa aplikasi berbasis Web.

Kata Kunci: Algoritma Dijkstra; Metode SAW; Haversine; Rute Terpendek; Angkutan Kota

Abstract—Angkot (City Transportation) in Sukabumi City is a popular public transportation system and is one of the main means of transportation for local residents and visitors to move around in the city with so many lanes each that the angkot itself passes, 19 route routes which cause so many problems for the community and outsiders of Sukabumi in finding angkot stop points so that it slows down time and is inefficient. Therefore, to make it easier to find angkot stopping points, Dijkstra's algorithm was chosen. Dijkstra's algorithm is the shortest route search algorithm by relying on the smallest value weight and calculating all neighboring nodes to find the smallest value. Researchers used 2 methods used, namely the SAW and Haversine methods. SAW method (Simple Additive Weighting Method). This method is for carrying out weighted sums based on 3 criteria for road width, distance traveled and estimated time and then normalizing the data to ranking. the haversine method is to calculate distances based on lattitude and lontitude. Initiate from the starting point to the destination point with a graph using Dijkstra's algorithm to produce a 30 iteration focus point generated by Dijkstra's algorithm, then evaluate the model by reaching other paths to the destination point. The results obtained in the search for the closest route using the two SAW and Haversine methods, after calculating all the paths passed from the starting point V117 (Bayangkara Street) - V82 (R.E Martadinata Street) obtained the results of the shortest route obtained, namely V117 - V118 - V131 - V130 - V132 - V133 - V134 - V138 - V137 - V110 - V82. Looking for the closest route from the starting point to the destination location determined by the researcher with the final result in the form of a Web-based application.

Keywords: Dijkstra's Algorithm; SAW Method; Haversine; Shortest Route; City Transport

1. PENDAHULUAN

Angkutan merupakan sebuah sarana untuk memindahkan barang ataupun manusia dari suatu tempat ke tempat lain dengan mudah. Berdasarkan [1]. Angkutan terbagi menjadi 2 macam diantaranya yaitu, angkutan umum dan angkutan barang, sebagaimana dalam segi fungsi nya, angkutan berperang penting untuk membantu manusia memudahkan segala aktifitas dalam menjangkau pada berbagai tempat yang akan di hendaki [2].

Transportasi merupakan salah satu sarana yang paling sering digunakan oleh masyarakat dalam menempuh perjalanan baik itu jalur darat, laut dan udara. Pada jalur darat transportasi banyak tersedia serta berbagai macam jenis dan tarif kendaraan seperti kereta api, bus, serta yang paling banyak di gunakan dan menjadi kendaraan alternatif yaitu Angkutan Kota (Angkot) [3].

Kota Sukabumi adalah salah satu Kota yang menggunakan alat transportasi angkot. Kota Sukabumi memiliki luas 4.800,231 Ha, untuk melakukan perjalanan baik itu di dalam Kota atau menuju ke luar Kota angkot menjadi salah satu alat transportasi alternatif [4]. Angkot (Angkutan Kota) yang berada di Kota Sukabumi terdiri dari 19 jalur trayek dan memiliki banyak macam warna seperti warna kuning, merah, putih dan lain-lain. Dari setiap jenis warna pada kendaraan terdapat perbedaan rute tujuan yang sudah di tetap kan oleh Dinas Perhubungan Kota Sukabumi dari kode trayek kendaraan yang bisa di lihat pada setiap Angkot yang berbeda jurusan, tujuan nya agar mempermudah dalam mengenali rute tujuan sesuai dengan warna dan kode trayek dari sekian banhyk nya.

Trayek kendaraan merupakan lintasan kendaraan yang telah memiliki rute tersendiri dan tetap secara flexible ” [5]. Sampai saat ini penggunaan angkot masih banyak digunakan oleh masyarakat Kota Sukabumi. Namun, masyarakat yang berasal dari luar kota sukabumi atau masyarakat kota yang belum mengetahui jalur trayek Angkot masih kesulitan dalam menemukan titik menunggu pemberhentian angkot.

Jarak adalah panjang pendeknya lintasan dari satu titik awal ke suatu titik akhir. Sedangkan lebar jalan adalah penampang badan jalan baik sempit atau lebar sehingga lebar jalan dapat mempengaruhi waktu yang dituju [6]. Agar dapat meminimalisir waktu dalam menemukan titik menunggu pemberhentian angkot sesuai dengan rute dapat memanfaatkan Algoritma Dijkstra. Algoritma Dijkstra adalah salah satu algoritma pencarian yang termasuk algoritma yang rakus dan tidak memikirkan konsekuensi kedepannya sehingga di golongan sebagai algoritma yang sangat cepat dalam mengambil keputusan [7]. Algoritma ini menghitung bobot paling kecil dalam melakukan perhitungannya. Tidak hanya itu ada beberapa algoritma optimasi yang serupa diantaranya Depth First Search (DFS), Breadth First Search (BFS), A-Star, Prim, Steepest Ascent Hill Climbing, Floyd-Warshall dan Kruskal. Beberapa peneliti pada sebelumnya melakukan penelitian yang hampir sama menggunakan algoritma Dijkstra, yang diteliti oleh [8]. untuk menemukan rute terpendek menggunakan Algoritma Dijkstra dan Haversine Formula.

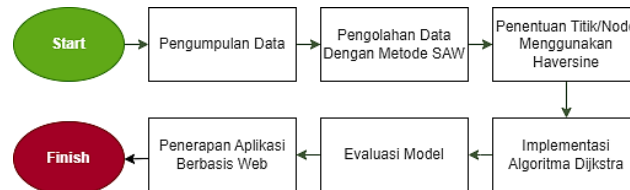
Perbandingan Algoritma Dijkstra dengan algoritma pencarian sejenis untuk optimasi waktu tercepat mengenai perbandingan algoritma Dijkstra, algoritma AStar dan algoritma Floyd Warshall. Hasilnya Algoritma Dijkstra lebih unggul dalam mencari rute dengan hasil yang di dapatkan yaitu algoritma Dijkstra di angka 0,0060 detik, algoritma AStar dengan 0,0067 detik dan Floyd Warshall sebesar 0,0433 detik [9].

Dari hasil uraian di atas maka akan di lakukan penerapan algoritma Dijkstra pada pencarian rute terdekat untuk menemukan jalur trayek angkutan kota (Angkot) dengan menghitung parameter lebar jalan, jarak tempuh dan estimasi waktu untuk melakukan pembobotan di titik yang ditentukan. Untuk menggabungkan beberapa nilai parameter tersebut ke dalam satu bobot maka akan digunakan sebuah metode SAW (Simple Additive Weighting Method) dan Haversine Formula sebagai metode untuk menentukan titik awal hingga akhir. Yang akan di diterapkan kedalam sebuah aplikasi berbasis web.

2. METODOLOGI PENELITIAN

2.1 Tahapan Penelitian

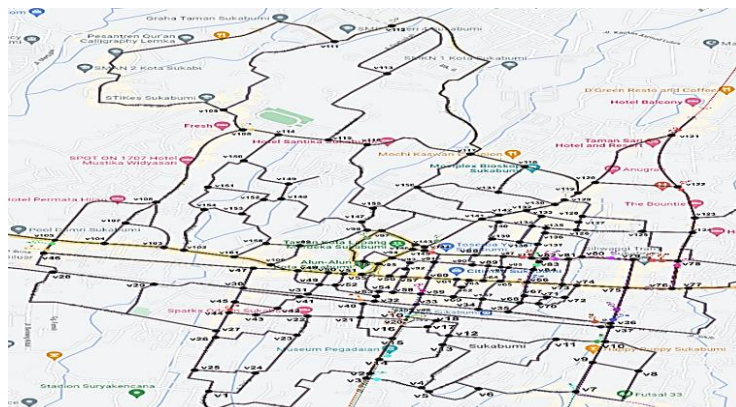
Pada penelitian ini menggunakan dua metode dalam pengambilan data dengan cara studi literatur dan melakukan survey lapangan. Dengan cara ini di dapatkan beberapa informasi dari sumber tertulis seperti jurnal, buku dan sebagainya, serta melakukan survey lapangan untuk memastikan kebenaran sebuah informasi dan data yang di dapatkan. Berikut adalah tahapan penelitian:



Gambar 1. Alur Penelitian

a. Pengumpulan data

Pada tahap ini merupakan langkah awal dalam pengambilan sebuah data, untuk mendapatkan sebuah data penulis melakukan observasi secara tidak langsung menggunakan Google Maps yakni mengumpulkan data jalan dan titik koordinat dari setiap persimpangan jalan yang akan menjadi penghubung untuk ke lokasi tujuan serta menghitung lebar jalan antar titik.



Gambar 2. Persebaran vertex di Lokasi Penelitian



- b. Pengolahan Data Dengan Metode SAW
Setelah data di dapatkan, tahap selanjutnya melakukan pengolahan data dengan 3 jenis-jenis kriteria diantaranya yaitu lebar jalan, jarak tempuh dan estimasi waktu, selanjutnya melakukan normalisasi data terhadap bobot yang di tentukan dan yang terakhir melakukan perangkingan untuk menjadi satu nilai bobot.
- c. Penentuan Titik/Node Menggunakan Haversine
Melakukan perhitungan dari titik awal menuju beberapa titik tujuan tempat menunggu pemberhentian angkot di Kota Sukabumi.
- d. Implementasi Algoritma Dijkstra
Melakukan pencarian menggunakan algoritma Dijkstra, dengan mengandalkan bobot paling kecil lalu kemudian di bandingkan Kembali dengan simpul tetangga dan untuk selanjutnya mendapatkan nilai paling kecil untuk di lewati.
- e. Evaluasi Model
Setelah melakukan pencarian Langkah selanjutnya menyusuri jalur lain untuk menuju ke titik tujuan, lalu di bandingkan kembali dengan jalur yang telah di dapatkan menggunakan algoritma dijkstra.
- f. Penerapan Pada Aplikasi Berbasis Web
Setelah hasil dari perhitungan dan pencarian menggunakan algoritma Dijkstra akurat, maka tahap selanjutnya yaitu menerapkan pada aplikasi berbasis web.

2.2 Algoritma Dijkstra

Algoritma Dijkstra digunakan untuk pencarian rute terpendek dengan mengandalkan sebuah bobot. Algoritma ini di kategorikan sebagai algoritma yang rakus karena menghitung semua simpul tetangga dan melakukan perbandingan Kembali agar mendapatkan nilai bobot terkecil untuk di lewati. Adapun Langkah-langka algoritma Dijkstra sebagai berikut:

- a. Inisialisasi dari titik awal ke titik tujuan yang telah di beri bobot hasil dari perhitungan dengan metode SAW
- b. Pemberian label sementara jarak dari setiap titik antara lain vertex (V1-V2)
- c. Menentukan nilai terkecil dari setiap titik label sementara
- d. Memberikan tanda label permanen dari titik nilai terkecil
- e. Menghapus label sementara dari daftar
- f. Melakukan perbandingan jarak terdekat melalui label permanen
- g. Jika titik terdekat ditemukan lebih dekat/nilai lebih kecil dari label yang dibandingkan maka di beri label permanen, jika tidak ada maka abaikan
- h. Lintasan terdekat di temukan

2.3 Metode SAW (Simple Additive Weighting Method)

Metode SAW digunakan untuk melakukan pengolahan data hingga menjadi sebuah bobot

- a. Menentukan kriteria untuk mencari titik pemberhentian angkot diantaranya, lebar jalan, jarak tempuh dan estimasi waktu.
- b. Penentuan tingkat kecocokan dari setiap alternatif sebuah persimpangan jalan pada kriteria yang di tentukan seperti lebar jalan, jarak tempuh dan estimasi waktu.
- c. Membuat sebuah matriks keputusan yang di dapatkan dari persamaan yang sesuai dengan jenis atribut cost dan benefit.

$$R_{ij} = \begin{cases} \frac{x_{ij}}{\max x_{ij}}, & \text{jika } j \text{ atribut keuntungan (benefit)} \\ \frac{\min x_{ij}}{x_{ij}}, & \text{jika } j \text{ atribut biaya (cost)} \end{cases} \quad (1)$$

Keterangan :

R_{ij} = nilai tingkat kinerja ternormalisasi

X_{ij} = nilai umum kriteria pada baris ke- i dan kolom ke- j

Max X_{ij} = nilai terbesar dari tiap-tiap baris dan kolom

Min X_{ij} = nilai terkecil dari tiap-tiap baris dan kolom

A = nilai alternatif

C = nilai kriteria

m = nilai kriteria ke- m

n = nilai kriteria ke- n

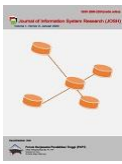
benefit = nilai terbesar yang terbaik

cost = nilai terkecil yang terbaik

R_{ij} merupakan rating kinerja yang telah ternormalisasi dan x_{ij} adalah nilai pada baris i kolom j dari matriks [10].

- d. Melakukan normalisasi data dengan perkalian matriks dengan bobot preferensi

$$V_i = \sum_{j=1}^n W_j R_{ij} \quad (2)$$



Keterangan:

V_i = nilai alternatif akhir,

W_j = bobot yang sudah ditentukan, dan

R_{ij} = rating kinerja ternormalisasi.

2.4 Metode Haversine

Pada penelitian ini metode haversine untuk melakukan perhitungan jarak tempuh dari titik lokasi awal hingga lokasi tujuan, dengan menghitung garis lintang (latitude) dan garis bujur (longitude) [11]. pada perhitungannya diasumsikan bumi itu bulat maka ditetapkan bahwa jari-jari bumi 6,371 km. rumus haversine sebagai berikut :

$$d = r \times \left(2 \operatorname{Arcsin} \left(\sqrt{\sin^2 \left(\frac{\varphi_x - \varphi_y}{2} \right)^2 + \cos \varphi_x \times \cos \varphi_y \times \sin^2 \left(\frac{\lambda_x - \lambda_y}{2} \right)^2} \right) \right) \tag{3}$$

Keterangan :

d = jarak

r = jari-jari bumi 6.371 km

φ_x = (latitude) 1

φ_y = (latitude) 2

λ_x = (longitude) 1

λ_y = (longitude) 2

3. HASIL DAN PEMBAHASAN

Pada pencarian rute terdekat untuk menemukan titik menunggu pemberhentian angkot di Kota Sukabumi terdapat banyak data vertex yang berjumlah 158 dari setiap persimpangan jalan lalu terdapat juga 35 lokasi tujuan dari 17 trayek angkot yang berada di Kota Sukabumi, sehingga algoritma dijkstra digunakan untuk melakukan pencarian berdasarkan bobot terkecil. Dalam penggunaannya sebelum menuju tahapan algoritma dilakukan terlebih dahulu pengolahan data dari 3 jenis kriteria menggunakan metode SAW, lalu melakukan normalisasi data hingga tahap akhir metode SAW yaitu perangkingan sampai menjadi sebuah bobot. Melakukan inisiasi dari titik awal hingga titik tujuan menggunakan metode haversine sehingga melaju ke tahap implementasi algoritma Dijkstra. Dengan menggunakan algoritma Dijkstra dapat menghasilkan pencarian rute dengan cepat dan akurat.

3.1 Pengumpulan Data

3.1.1 Pengambilan Data Jalan Yang di Lewati

Dari hasil pengumpulan data jalan yang terhubung dengan 158 vertex berjumlah 66 jalan yang diambil pada penelitian ini

Tabel 1. Jalan Raya Kota Sukabumi

No	Nama Jalan
1.	Gg Ajid 3
2.	Gg Ajid 4
3.	Gg. Arjuna
4.	Gg. Dahlia I
5.	Gg. Irawan
6.	Gg. Rawa Salak
7.	Gg. Tipar
8.	Jl., Bayangkara
9.	Jl. Alun-Alun Utara
10.	Jl. Aminta Azmali
11.	Jl. Arif Rahman Hakim
12.	Jl. Balai Desa
13.	Jl. Benteng
14.	Jl. Cikole Dalam
15.	Jl. Ciwangi
16.	Jl. Dewi Sartika
17.	Jl. Ir H Juanda
18.	Jl. Jendral Ahmad Yani
19.	Jl. Kapten Harun Kabir
20.	Jl. Karamat
21.	Jl. Kelenteng



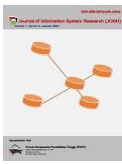
No	Nama Jalan
22.	Jl. Kenari
23.	Jl. Kopeng
24.	Jl. Kuta Pasir
25.	Jl. Lettu Bakri
26.	Jl. Manggis
27.	Jl. Masjid
28.	Jl. Mayawati
29.	Jl. Mayawati Atas
30.	Jl. Nanggaleng
31.	Jl. National
32.	Jl. National Rte 3
33.	Jl. National Rte 4
34.	Jl. Nyomplong
35.	Jl. Otto Iskandardinata
36.	Jl. Pabuaran
37.	Jl. Pajagalan
38.	Jl. Parigi
39.	Jl. Pasundan
40.	Jl. Pelabuhan II
41.	Jl. Pemuda
42.	Jl. Pemuda II
43.	Jl. Pengadilan
44.	Jl. Pengadilan
45.	Jl. Perintis Kemerdekaan
46.	Jl. Perniagaan
47.	Jl. Ps.Timur
48.	Jl. R.A Kosasih
49.	Jl. R.E Martadinata
50.	Jl. Rh. Didi Sukardi
51.	Jl. Rumah Sakit Belakang
52.	Jl. Saniin
53.	Jl. Siliwangi
54.	Jl. Soebarkah
55.	Jl. Sriwijaya
56.	Jl. Stasiun Barat
57.	Jl. Stasiun Timur
58.	Jl. Surya Kencana
59.	Jl. Swidari
60.	Jl. Taman Bahagia
61.	Jl. Tipar
62.	Jl. Tipar Gede
63.	Jl. Veteran I
64.	Jl. Veteran II
65.	Jl. Yulius Usman
66.	Jl. Zaenal Zakse

3.1.2 Pengambilan Data Koordinat

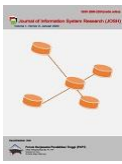
Hasil pengambilan data titik koordinat dari setiap Vertex yang telah di tentukan pada persimpangan jalan yang di lalui oleh masyarakat untuk menemukan titik pemberhentian angkot di Kota Sukabumi

Tabel 2. Vertex Lokasi Persimpangan Jalan

No	Vertex	Koordinat
1	V1	-6.931283724322995, 106.92084939344362
2	V2	-6.929958288262984, 106.92588757998976
3	V3	-6.930268990174118, 106.92586559836228
4	V4	-6.931040294060108, 106.92753556179935
5	V5	-6.931439944276487, 106.92777897686275
6	V6	-6.931110332890111, 106.92938472864752
7	V7	-6.931465406275524, 106.93239618638562
8	V8	-6.92966268597533, 106.93440661071313



No	Vertex	Koordinat
9	V9	-6.929005803645851, 106.93299647376587
10	V10	-6.927404901112504, 106.93332839712664
11	V11	-6.927102277938578, 106.93182211408487
12	V12	-6.926883415307892, 106.92842799214583
13	V13	-6.927629626861619, 106.92823263371429
14	V14	-6.928868777339756, 106.9262058118538
15	V15	-6.927081918254047, 106.92664473661875
16	V16	-6.9260815764981825, 106.92695431780776
17	V17	-6.926179735672178, 106.9278187270905
18	V18	-6.925633237351347, 106.92803261468225
19	V19	-6.9253558428236515, 106.92715152443503
20	V20	-6.925583658016427, 106.9270680736722
21	V21	-6.925374670536413, 106.9255906158987
22	V22	-6.9252860210354505, 106.92399417976658
23	V23	-6.926808647784577, 106.9236873269482
24	V24	-6.929859968848001, 106.92235300799994
25	V25	-6.92964502644458, 106.92036820318195
26	V26	-6.926940330112837, 106.92104834326773
27	V27	-6.926477816211182, 106.92115041373442
28	V28	-6.921723352454664, 106.91606980348455
29	V29	-6.9226794698825636, 106.91922139116187
30	V30	-6.923034101848092, 106.92191733065305
31	V31	-6.923508053829049, 106.9239987248534
32	V32	-6.924045908871239, 106.92606939024101
33	V33	-6.924280221771152, 106.92744804567452
34	V34	-6.9247592530845, 106.92934947960559
35	V35	-6.924920461569541, 106.93026181315395
36	V36	-6.925869799315797, 106.93354576287206
37	V37	-6.926089221441636, 106.93351418643176
38	V38	-6.924911505569859, 106.92724851849496
39	V39	-6.924753183857769, 106.92729317441992
40	V40	-6.924320190964712, 106.92574504466621
41	V41	-6.923692866205044, 106.9236560429344
42	V42	-6.924631272593173, 106.92304841269865
43	V43	-6.9251822319812195, 106.92229245331367
44	V44	-6.925014411086661, 106.92147649309234
45	V45	-6.924128522521333, 106.92166637816032
46	V46	-6.920161406118822, 106.91565113090496
47	V47	-6.9214099473765565, 106.92226885964728
48	V48	-6.921649516329615, 106.9237090754255
49	V49	-6.922710749523759, 106.92437428484918
50	V50	-6.921798432270713, 106.92463762232775
51	V51	-6.921865839243865, 106.92513953898099
52	V52	-6.922814051752837, 106.92484161733056
53	V53	-6.923645436001669, 106.92614849275606
54	V54	-6.923300532214603, 106.92620692448726
55	V55	-6.921983427434203, 106.92591378180852
56	V56	-6.922038631791279, 106.92659622142101
57	V57	-6.92209246147541, 106.92703161574464
58	V58	-6.923343654401506, 106.92668266122193
59	V59	-6.92346799197401, 106.92756250183024
60	V60	-6.9222156621532624, 106.92785276470626
61	V61	-6.922325533149241, 106.92872475101794
62	V62	-6.922927828774956, 106.92886458953923
63	V63	-6.922354640390472, 106.92911269014147
64	V64	-6.922371286403925, 106.92950050410921
65	V65	-6.922437852808622, 106.93012814101333
66	V66	-6.922575544099671, 106.93091894642036
67	V67	-6.923586582172465, 106.93077431245072
68	V68	-6.924221631115711, 106.93065229236218
69	V69	-6.9237054153466975, 106.92855221864951

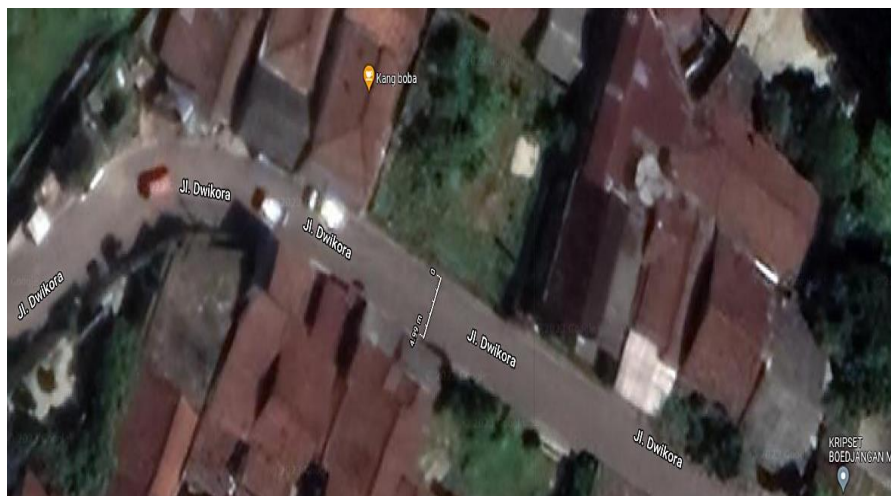


No	Vertex	Koordinat
70	V70	-6.924241482468165, 106.931252927918
71	V71	-6.923761438844845, 106.9315082987519
72	V72	-6.923775206664867, 106.93215960834021
73	V73	-6.92269146341737, 106.93199808847254
74	V74	-6.9227923382659196, 106.93245752911399
75	V75	-6.922954628937173, 106.93401706093607
76	V76	-6.923055809752844, 106.93548154690194
77	V77	-6.92303139240348, 106.93595359396788
78	V78	-6.9208405797352, 106.9356640226421
79	V79	-6.92074079193303, 106.9342889817107
80	V80	-6.920544981808239, 106.9325972072394
81	V81	-6.920701253353433, 106.9322671973221
82	V82	-6.920652300831312, 106.93150096769794
83	V83	-6.920802923960071, 106.93129234082008
84	V84	-6.9215092187017735, 106.93109331055713
85	V85	-6.921326826353249, 106.93029803561208
86	V86	-6.9212762357597954, 106.93007943556495
87	V87	-6.9204108695429705, 106.93015185520764
88	V88	-6.92033311748606, 106.92940732139255
89	V89	-6.921199812543529, 106.92967607367727
90	V90	-6.921060993030182, 106.92920693804469
91	V91	-6.92091321737796, 106.92800703328294
92	V92	-6.921375208152998, 106.92710563741251
93	V93	-6.920949552771206, 106.92719913577008
94	V94	-6.9205353329362715, 106.92665556996006
95	V95	-6.921041352793534, 106.92618868978299
96	V96	-6.920940596670586, 106.92570151039477
97	V97	-6.918347313907382, 106.92625897451293
98	V98	-6.918343451489344, 106.92585510928888
99	V99	-6.919586892897961, 106.92370710324572
100	V100	-6.921136840666126, 106.92280081031947
101	V101	-6.920487683425931, 106.92170114072856
102	V102	-6.919950710731745, 106.92004750587749
103	V103	-6.919742772314514, 106.91947726771454
104	V104	-6.9194272456339085, 106.91754339502019
105	V105	-6.9191636409072, 106.91601587698639
106	V106	-6.916133479159345, 106.91916732926485
107	V107	-6.91780013079515, 106.91824986381148
108	V108	-6.910227401205698, 106.92196204108978
109	V109	-6.908621350855137, 106.92132838385115
110	V110	-6.920588785701861, 106.93117250640765
111	V111	-6.903109977638332, 106.92483205165722
112	V112	-6.901774142880969, 106.9266663639001
113	V113	-6.905683093210194, 106.92643840998862
114	V114	-6.910719449381788, 106.92343478638598
115	V115	-6.911046964112144, 106.92462836938809
116	V116	-6.911387792219535, 106.92587023215658
117	V117	-6.912182087070565, 106.92916955056492
118	V118	-6.91291256497632, 106.93039623344607
119	V119	-6.915315276719187, 106.9325413717298
120	V120	-6.914434208975944, 106.93307027708639
121	V121	-6.910796824426536, 106.9354039785027
122	V122	-6.914858175681361, 106.93566760839867
123	V123	-6.917319011490071, 106.93622446396301
124	V124	-6.918500392919313, 106.93609711191553
125	V125	-6.918407864830235, 106.93461854419624
126	V126	-6.918300934010578, 106.93245753723065
127	V127	-6.9176158314499165, 106.9325203359604
128	V128	-6.917216428764584, 106.9326431071312
129	V129	-6.915925824458903, 106.93210535008751
130	V130	-6.916230459022595, 106.93181901515206

No	Vertex	Koordinat
131	V131	-6.9152091467561005, 106.93000741695596
132	V132	-6.916754682788774, 106.93085980107654
133	V133	-6.917161789111415, 106.93138893799514
134	V134	-6.917962199705104, 106.93104238631169
135	V135	-6.918143257049058, 106.93153332123961
136	V136	-6.918506936798004, 106.93144624693431
137	V137	-6.9194257344180325, 106.9312352553062
138	V138	-6.919284474021571, 106.93068523895772
139	V139	-6.919017327373027, 106.92984373550583
140	V140	-6.917649974196362, 106.93002793641519
141	V141	-6.917387699816798, 106.92954647991216
142	V142	-6.917110953513679, 106.93001908103274
143	V143	-6.919683710757825, 106.92754861635132
144	V144	-6.9197466938993655, 106.92766132081452
145	V145	-6.919879022086478, 106.92764078527564
146	V146	-6.919788780701424, 106.92742779340023
147	V147	-6.917710984455599, 106.92566060602083
148	V148	-6.915807925209871, 106.92309608557876
149	V149	-6.914547658801387, 106.92338794330315
150	V150	-6.912759926017589, 106.92196872476616
151	V151	-6.914866661190431, 106.92087146732172
152	V152	-6.915566473146669, 106.92213305916188
153	V153	-6.916977360931974, 106.92190651796045
154	V154	-6.916591671473243, 106.92078210563297
155	V155	-6.916591719635936, 106.92604379649357
156	V156	-6.914785925713091, 106.92739256150274
157	V157	-6.923882975061092, 106.92949101216405
158	V158	-6.91943673808607, 106.92265280648824

3.1.2 Jarak Tempuh Pada Setiap Titik dan Lebar Jalan yang Dilalui

Pada tahap ini penulis melakukan penentuan nilai pembobotan dari 3 jenis kriteria pada setiap titik yang saling terhubung menggunakan Metode SAW diantaranya yaitu :
 Pengambilan data di lakukan dengan aplikasi Citra Satelit dari V1- V2 = 4,99 m.



Gambar 3. Lebar Jalan

Untuk menghitung jarak tempuh menggunakan perhitungan antara garis lintang dan garis bujur dari setiap titik koordinat dari persimpangan jalan antar vertex. pertama - pertama yang dilakukan yaitu mengkonversi Titik koordinat kedalam format radian lalu menghitung setiap variabel yang dibutuhkan.

1. Titik Kordinat (V1)

$$\begin{aligned}
 \text{Garis lintang (latitude) 1} &= -6.931283724322995 \\
 \phi_x &= (\text{latitude) 1} * \pi / 180 \\
 \phi_x &= -6.931283724322995 * 3.14159265359 / 180 \\
 \phi_x &= -0,1209737223793
 \end{aligned}$$



- Garis bujur (longitude) 1 = 106.92084939344362
- λ_x = (longitude) 1 * r / 180
- λ_x = 106.92084939344362 * 3.14159265359 / 180
- λ_x = 1,8661208609447
- 2. Titik Kordinat (V2)
- Garis lintang (latitude) 2 = -6.929958288262984
- Φ_y = (latitude) 2 * r / 180
- ϕ_y = -6.929958288262984 * 3.14159265359 / 180
- ϕ_y = -0,1209505891561
- Garis bujur (longitude) 2 = 106.92588757998976
- λ_y = (longitude) 2 * r / 180
- λ_y = 106.92588757998976 * 3.14159265359 / 180
- λ_y = 1,8662087938882

Setelah nilai dari setiap variabel nya di temukan, maka Langkah selanjutnya melakukan perhitungan dengan rumus Haversine

$$d = r \times \left(2 \operatorname{Arcsin} \left(\sqrt{\sin^2 \left(\frac{\text{lat}2 - \text{lat}1}{2} \right)^2 + \cos \phi_x (\text{lat}1) \times \cos \phi_y (\text{lat}2) \times \sin^2 \left(\frac{\text{lon}2 - \text{lon}1}{2} \right)^2} \right) \right)$$

$$d = 6.371,1 \times \left(2 \operatorname{Arcsin} \left(\sqrt{\sin^2 \left(\frac{-6.929958288262984 - (-6.931283724322995)}{2} \right)^2 + \cos(-0,1209737223793) \times \cos(-0,1209505891561) \times \sin^2 \left(\frac{106.92588757998976 - 106.92084939344362}{2} \right)^2} \right) \right)$$

$d = 0.5754875 \text{ km}$

$d = 575.4875 \text{ m}$

Jadi nilai akhir dari jarak tempuh antara V1 dan V2 adalah 575.4875 m

3.2 Pengolahan Data dan Implementasi

Pada tahap ini melakukan pengolahan data menggunakan metode SAW, Haversine dan implementasi algoritma dijkstra

3.2.2 Analisis Dari Jenis Kriteria

Penentuan 3 jenis kriteria dari lebar jalan, jarak tempuh dan estimasi waktu berdasarkan parameter masing-masing yang di dapatkan dari hasil observasi secara tidak langsung menggunakan bantuan Google Maps.

Tabel 3. Tabel Kriteria

Kode	Kriteria	Bobot	Atribut
C1	Lebar Jalan	40	Benefit
C2	Jarak Tempuh	30	Cost
C3	Estmasi waktu	30	Cost

Pada **Tabel 3** terdapat 2 jenis atribut diantaranya yaitu benefit yaitu jika nilai bobot lebih besar maka akan menjadi sebuah keuntungan, sebaliknya nilai cost lebih besar maka akan menjadi sebuah kerugian. Selanjutnya Diketahui **Tabel 3** Melakukan normalisasi pada jenis kriteria menggunakan persamaan (3)

Perhitungan tersebut merupakan hasil dari perhitungan 3 jenis kriteria benefit dan cost. Adapun sample hasil perhitungan sebagai berikut

Tabel 4. Sample Perhitungan 3 Jenis Kriteria

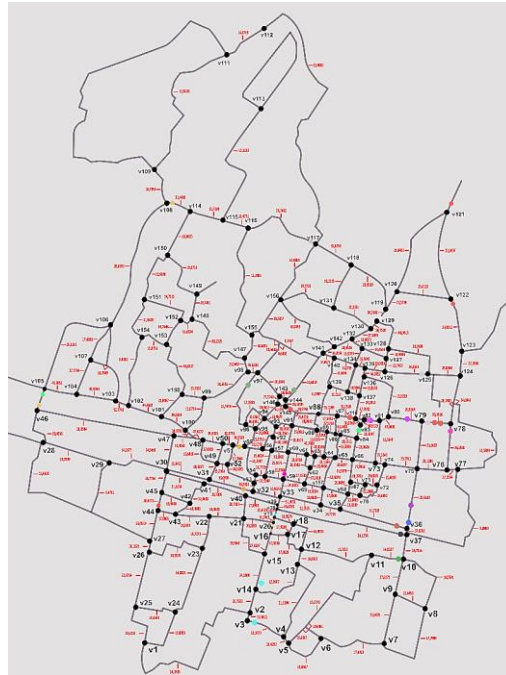
NO	Vertex	Kode Kriteria		
		C1	C2	C3
1	V1-V2	0,3136	0,0248	0,0248
2	V1-V24	0,2018	0,0622	0,0622
3	V1-V25	0,4978	0,0752	0,0752

Selanjutnya yaitu melakukan perangkingan dari 3 jenis kriteria hingga menjadi bobot. Berikut sebagai contoh sample di atas V1-V2, V1-V24 dan V1-V25.

V1 (Preferensi) = {(C1 (0,3136) * Bobot C1 (40)) + (C2 0,0248) * Bobot C2(30)) + (C3 (0,0248) * Bobot C3(30))}

V1 (Preferensi) = 14,0345

Jadi bobot akhir dari V1-V2 adalah 14,0345. Bobot ini akan digunakan pada proses selanjutnya untuk mencari rute terdekat menggunakan algoritma Dijkstra. Diantaranya persebaran bobot sebagai berikut.



Gambar 4. Persebaran Bobot Akhir

3.2.3 Penentuan Titik Awal dan Lokasi Tujuan

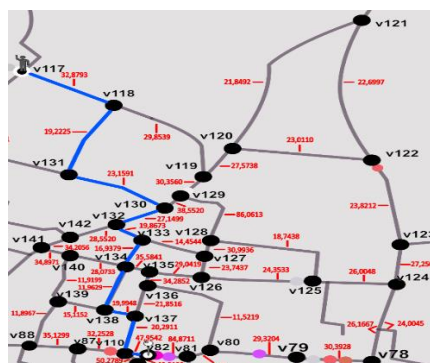
Sebelum menjalankan pencarian dengan algoritma dijkstra selanjutnya menentukan titik awal lokasi V117 dan titik tujuan tempat pemberhentian angkot dengan menggunakan metode Haversine yang berada di 3 lokasi yaitu V59, 78 dan V82. Karena lokasi V117 menuju V82 paling dekat, di pilih sebagai lokasi tujuan untuk trayek angkot Sukaraja V82.

3.2.4 Implementasi Algoritma Dijkstra

Titik awal mulai dari V117 yang berlokasi di jalan bayangkara menuju V82 di jalan R.E Martadinata pada trayek angkot Sukaraja. Berikut merupakan Langkah-langkah algoritma dijkstra.

- Inisialisasi titik awal hingga ke titik tujuan yang telah di beri bobot hasil dari perhitungan dengan metode SAW
- Pemberian label sementara jarak dari setiap titik antara lain vertex (V1-V2)
- Menentukan nilai terkecil dari setiap titik label sementara
- Memberikan tanda label permanen dari titik nilai terkecil
- Menghapus label sementara dari daftar
- Melakukan perbandingan jarak terdekat melalui label permanen
- Jika titik terdekat ditemukan lebih dekat/nilai lebih kecil dari label yang dibandingkan maka di beri label permanen, jika tidak ada maka abaikan
- Lintasan terdekat di temukan

Sehingga di fokuskan dengan cara mmperecil jangkauan untuk mendapatkan rute terdekat dengan 30 proses iterasi yaitu V117-V118-V131-V130-V132-V133-V134-V138-V137-V110-V82, bila di gambarkan dalam bentuk graf sebagai berikut.



Gambar 5. Mencari Rute Terpendek

3.2.5 Evaluasi Model

Menganalisis jalur-jalur lain yang di lewati menuju lokasi tujuan dengan mengambil beberapa rute lalu di bandingkan.

Tabel 5. Evaluasi Model

No	Rute	Hasil
1.	V117-V118-V119-V120-V121-V122-V123-V124-V125-V126-V135-V136-V137-V110-V82	389,6998
2.	V117-V118-V119-V120-V122-V122-V123-V124-V78-V79-V80-V81-V82	402,6568
3.	V117-V118-V119-V129-V128-V127-V126-V135-V136-V80-V81-V82	439,5960
4.	V117-V118-V131-V130-V132-V133-V134-V138-V137-V110-V82	239,4190
5.	V117-V118-V131-V130-V132-V142-V141-V140-V139-V138-V137-V110-V82	315,3407
6.	V117-V118-V131-V130-V132-V142-V141-V140-V139-V88-V87-V110-V82	339,2190
7.	V117-V118-V131-V130-V132-V142-V141-V134-V135-V136-V137-V110-V82	535,2079
8.	V117-V118-V131-V130-V129-V128-V127-V126-V135-V136-V80-V81-V82	460,3197
9.	V117-V118-V119-V129-V130-V132-V133-V134-V138-V110-V82	295,7993

Dari beberapa rute yang di lewati terdapat rute yang memiliki bobot paling kecil pada nomer 4 yaitu V117-V118-V131-V130-V132-V133-V134-V138-V137-V110-V82 dengan bobot nilai 239,4190.

3.3 Penerapan Model Kedalam Aplikasi Berbasis Web

Langkah terakhir yaitu menerapkan dari setiap tahapan-tahapan penelitian pada aplikasi berbasis Web untuk mencari rute terdekat dalam menemukan tempat pemberhentian angkot di Kota Sukabumi. Pada Tampilan ini yaitu merupakan basis data yang terdiri 9 tabel yang telah dibuat.

Tabel	Tindakan	Baris	Jenis	Penyortiran	Ukuran	Beban
failed_jobs	Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	0	InnoDB	utf8mb4_unicode_ci	32.0 KB	-
graphs	Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	500	InnoDB	utf8mb4_unicode_ci	176.0 KB	-
migrations	Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	8	InnoDB	utf8mb4_unicode_ci	16.0 KB	-
password_reset_tokens	Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	0	InnoDB	utf8mb4_unicode_ci	16.0 KB	-
personal_access_tokens	Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	5	InnoDB	utf8mb4_unicode_ci	48.0 KB	-
transportations	Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	17	InnoDB	utf8mb4_unicode_ci	32.0 KB	-
transportation_destinations	Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	34	InnoDB	utf8mb4_unicode_ci	48.0 KB	-
users	Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	1	InnoDB	utf8mb4_unicode_ci	48.0 KB	-
vertices	Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	158	InnoDB	utf8mb4_unicode_ci	32.0 KB	-
9 tabel	Jumlah	723	InnoDB	utf8mb4_general_ci	448.0 KB	0 B

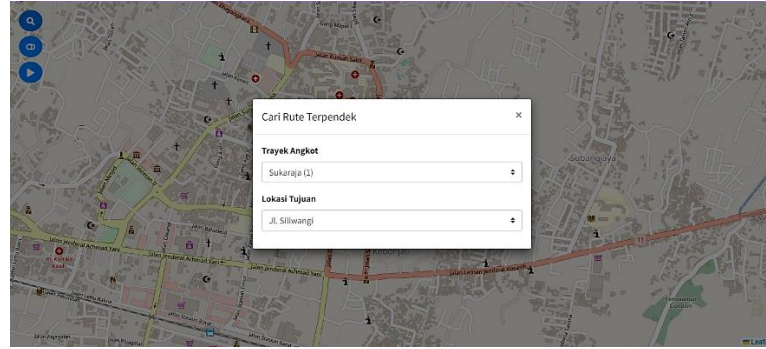
Gambar 6. Tampilan Basis Data

Pada tampilan ini merupakan tampilan awal yang muncul pada saat user mengakses aplikasi pencarian rute terdekat untuk menemukan titik pemberhentian angkot.



Gambar 7. Tampilan Awal User

Tahap selanjutnya ketika user ingin mencari lokasi angkot, maka user dapat memilih tombol pencarian untuk mengisi trayek angkot yang di pilih beserta lokasi tujuan nya



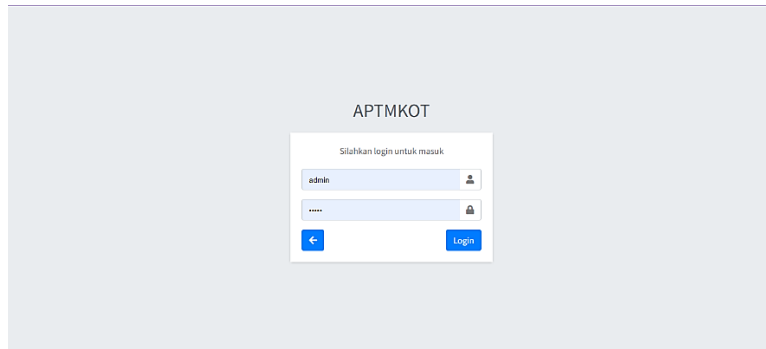
Gambar 8. Pengisian Rute Pencarian

Pada tahap ini setelah user memulai mencari maka system akan menunjukkan graph berwarna biru untuk mengarahkan user hingga ke tempat lokasi tujuan terdekat.



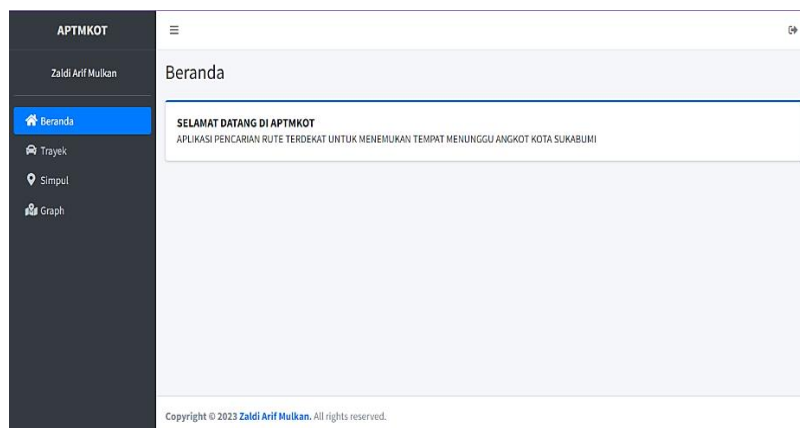
Gambar 9. Pencarian Rute Terdekat

Untuk masuk ke menu admin, maka seorang admin harus terlebih dahulu melakukan login terhadap aplikasi



Gambar 11. Tampilan Login Admin

Setelah admin melakukan login, untuk selanjutnya akan tampil halaman beranda admin yang bertuliskan selamat datang.



Gambar 12. Tampilan Beranda Admin

Pada menu ini seorang admin dapat melakukan perubahan sebuah data ataupun melakukan penambahan data terkait trayek angkot.

No	Jurusan	Kode Trayek	Aksi
1	Sukaraja	1	[Edit] [Hapus]
2	Lembursitu	3	[Edit] [Hapus]
3	Dwikora	03B	[Edit] [Hapus]
4	Goalpara	4	[Edit] [Hapus]
5	Subangjaya	5	[Edit] [Hapus]
6	Cisaat	8	[Edit] [Hapus]

Gambar 13. Tampilan Trayek Admin

Pada menu ini merupakan daftar simpul yang berisikan vertex, titik koordinat dan informasi sebuah jalan tempat lokasi vertex tersebut. Lalu admin dapat melakukan sebuah perubahan data.

No	Label	Latitude	Longitude	Deskripsi	Aksi
1	v1	-6.931283724323	106.92084939344	Jl. Pabuaran	[Edit] [Hapus]
2	v2	-6.92958288263	106.92588757999	Jl. Pelabuhan II	[Edit] [Hapus]
3	v3	-6.9302689901741	106.92586559836	Jl. Pelabuhan II	[Edit] [Hapus]
4	v4	-6.9310402940601	106.9275355618	Jl. Pemuda	[Edit] [Hapus]
5	v5	-6.9314309442765	106.92777897686	Jl. Pemuda II	[Edit] [Hapus]
6	v6	-6.9311103328901	106.92938472865	Jl. Pemuda	[Edit] [Hapus]

Gambar 14. Tampilan Simpul Admin

Selanjutnya yaitu menu Graph yang berfungsi sebagai penghubung antara vertex yang di dalam nya terdapat nilai-nilai jarak tempuh dan estimasi waktu secara otomatis ketika di input data.

No	Graph	Jarak Tempuh (m)	Lebar Jalan (m)	Estimasi Waktu (m/s)	Aksi
1	v33-v38	73.590466819864	7.94	61.325388849886	[Edit] [Hapus]
2	v39-v38	18.2808179193063	7.94	15.23901492392	[Edit] [Hapus]
3	v40-v39	177.59165871196	4.88	147.99304725997	[Edit] [Hapus]
4	v41-v40	240.98912422805	6.19	200.81677018837	[Edit] [Hapus]
5	v21-v40	118.51889632289	5.37	98.765746939707	[Edit] [Hapus]
6	v42-v41	124.07870891159	6.19	103.30892242632	[Edit] [Hapus]

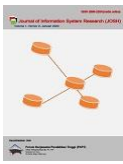
Gambar 15. Tampilan Graph Admin

4. KESIMPULAN

Setelah melakukan penelitian diatas dapat disimpulkan dalam pencarian rute terdekat dengan menggunakan dua metode SAW dan Haversine, setelah menghitung semua jalur - jalur yang dilewati dari titik awal V117 (Jalan Bayangkara) - V82 (Jalan R.E Martadinata) didapatkan hasil rute terpendek yang didapat yaitu V117 - V118 - V131 - V130 - V132 - V133 - V134 - V138 - V137 - V110 - V82. Dengan hasil penerapan algoritma dijkstra pada aplikasi berbasis web dapat membantu masyarakat dalam mencari rute terdekat untuk menemukan titik menunggu pemberhentian angkot di Kota Sukabumi. Adapun saran untuk penelitian selanjutnya dapat memperluas jangkauan peta yang dibuat, mempertimbangkan kepadatan lalu lintas yang berada di Kota Sukabumi dan mengintegrasikan dengan instansi terkait angkutan untuk kelengkapan sebuah data.

UCAPAN TERIMAKASIH

Peneliti mengucapkan terima kasih kepada Bapak Iwan Rizal Setiawan, M.T., M.Kom. dan Ibu Fathia Frazna A, S.ST., M.T. telah membimbing pada penulisan penelitian ini, serta kepada kedua orang tua yang selalu mendukung dan selalu memberi semangat.



REFERENCES

- [1] P. M. Kemenhub, “PERATURAN MENTERI PERHUBUNGAN REPUBLIK INDONESIA NOMOR PM 15 TAHUN 2019.” 2019.
- [2] M. Ogotan, “EFEKTIVITAS PEMBERIAN IZIN TRAYEK ANGKUTAN UMUM (STUDY DI DINAS PERHUBUNGAN KOTA MANADO) Friedrick Napoleon Lopian Patar Rumapea PENDAHULUAN Transportasi atau pengangkutan merupakan bidang kegiatan yang sangat penting beberapa dalam kehidupan masyarakat”.
- [3] Y. H. Nuryoso, Pradjoko, and Lelah, “Penerapan Algoritma A* pada Pencarian Rute Terpendek pada Rute Angkot Di Kota Sukabumi,” J. Sarj. Tek. Inform. Vol. 8, No. 1, Februari 2020, pp. 21–35 e-ISSN, vol. 8, no. 1, pp. 21–35, 2020.
- [4] Badan Pengembangan Kawasan Perkotaan, “Profil Kota Sukabumi,” <https://portal.sukabumikota.go.id/>, 2017. <https://portal.sukabumikota.go.id/>
- [5] Peraturan Menteri Perhubungan Republik Indonesia, “Peraturan Menteri Perhubungan Republik Indonesia Nomor PM 15 Tahun 2019 Tentang Penyelenggaraan Angkutan Orang Dengan Kendaraan Bermotor Umum Dalam Trayek,” PM 15 Tahun 2019, p. 13, 2019, [Online]. Available: http://jdih.dephub.go.id/assets/uudocs/permen/2015/PM_180_Tahun_2015.pdf
- [6] A. Aji, “Optimasi Jalur Tercepat Degan Menggunakan Modifikasi Algoritma Bellman Fors (Studi Kasus Lintas Antar Kecamatan Kota Malang),” J. EECCIS, vol. 9, no. 2, pp. 168–172, 2015.
- [7] E. S. Arga, G. G. Firmansyah, K. Imam, and M. Fauzi, “Penerapan algoritma djikstra pada pencarian jalur terpendek,” vol. 1, no. 2, pp. 134–142, 2021.
- [8] T. A. Labibah, “Pencarian Rute Trayek Angkot Terpendek Menggunakan Metode Algoritma Dijkstra Dan Haversine Formula,” FIKI(Jurnal Teknol. Inf. dan Komunikasi), vol. VIII, no. 2, pp. 128–134, 2018.
- [9] Umar et al., “Wisata Kabupaten Dompu Comparative Analysis of Dijkstra , a-Star , and Floyd Warshall Algorithm in Searching the Nearest Route Search To Tourism,” vol. 8, no. 2, pp. 227–234, 2021, doi: 10.25126/jtiik.202182866.
- [10] E. Marbun and S. Hansun, “Sistem Pendukung Keputusan Pemilihan Program Studi Dengan Metode Saw Dan Ahp,” Ilk. J. Ilm., vol. 11, no. 3, pp. 175–183, 2019, doi: 10.33096/ilkom.v11i3.432.175-183.
- [11] D. O. Sihombing, W. Nugraha, and F. Andani, “Aplikasi Pengelolaan Data Order Mkios Berbasis Web Pada TDC PT . Telesindo Shop Pontianak,” Simp. Nas. Ilmu Pengetah. dan Teknol. 2016, vol. Vol.1, no. 1, pp. 129–138, 2016.
- [12] D. G. Monica, “Dea Gratyas Monica NIM. 171414070,” 2021.
- [13] M. Arizal Fahribi Cahyo, “Sistem Pencarian Jarak Terdekat menggunakan Metode Haversine Berbasis Android,” Ekon. Akuntansi, vol. 01, no. 08, pp. 1–13, 2017.
- [14] H. Anra, “Sistem Informasi Geografis Perumahan dan Fasilitas Sosial Terdekat dengan Metode Haversine Formula,” J. Sist. dan Teknol. Inf., vol. 1, no. 1, pp. 1–6, 2016.
- [15] T. Rangga, G. Erwin, and M. Yana, “Perancangan Sistem Evakuasi Tsunami Dengan Metode Clustering Dan Algoritma Dijkstra Berbasis Android,” 2020, [Online]. Available: [http://repository.nusaputra.ac.id/id/eprint/60/%0Ahttp://repository.nusaputra.ac.id/id/eprint/60/1/Rangga Turnadi_Erwin_Gunawan_Yana_Mulia_TI20.pdf](http://repository.nusaputra.ac.id/id/eprint/60/%0Ahttp://repository.nusaputra.ac.id/id/eprint/60/1/Rangga_Turnadi_Erwin_Gunawan_Yana_Mulia_TI20.pdf)
- [16] S. Pratama, “Perbandingan Algoritma Dijkstra dan Algoritma Floyd Warshall Pada Pencarian Rute Pariwisata di Kota Palembang,” Sriwij. Univ. Repos., 2018, [Online]. Available: <http://repository.unsri.ac.id/id/eprint/10013>.
- [17] Menteri Perhubungan RI, “Peraturan Menteri Perhubungan Republik Indonesia Nomor PM 15 Tahun 2019 Tentang Penyelenggaraan Angkutan Orang Dengan Kendaraan Bermotor Umum Dalam Trayek,” PM 15 Tahun 2019, p. 13, 2019.
- [18] R. A. D. Novandi, “Perbandingan Algoritma Dijkstra dan Algoritma Floyd-Warshall dalam Penentuan Lintasan Terpendek (Single Pair Shortest Path),” IF2251 Strateg. Algoritm., vol. 1, pp. 1–5, 2013.
- [19] D. Yogaswara and S. Suhartono, “Perbandingan Algoritma A-Star dan Dijkstra pada Pencarian Jalur Evakuasi Tsunami Terpendek Menuju Shelter di Kabupaten Bantul Berbasis Aplikasi Android,” J. Masy. Inform., vol. 12, no. 1, pp. 10–18, 2021, doi: 10.14710/jmasif.12.1.41018.
- [20] R. Paryanti and A. Thobirin, “Penerapan Teori Graf untuk Mencari Lintasan Tercepat Bus Trans-Jogja(Rosyana Paryanti) Penerapan Teori Graf untuk Mencari Lintasan Tercepat Bus Trans-Jogja,” Univ. Ahmad Dahlan, pp. 1–9, 2011.