

Pencarian Rute Terdekat Menghindari Jalan Rusak Menggunakan Algoritma A-Star

Rifki Setiawan

Universitas Pembangunan Nasional Veteran Jawa Timur

Alamat: Jalan Raya Rungkut Madya No. 1, Gunung Anyar, Surabaya

Korespondensi penulis: rifkisetiawan1038@gmail.com

Abstract. *In our daily lives, we are often faced with the challenge of broken roads that can affect our travel. To solve this problem, a web-based map system has been developed. The system uses road data from OpenStreetMap and the A-star algorithm to find the shortest route that avoids bad roads. The website is built using JavaScript, PHP, HTML, and CSS, with the support of a MySQL database that stores information on damaged roads and user accounts. The process of using the system starts with registration and account login. Users can view the map on the main page, report the damaged road with photo evidence equipped with a time watermark and coordinate data, and view their initial route. The admin then validates the damaged road data, and the system automatically constructs an alternative route that avoids the verified damaged roads. The accuracy of the A-star algorithm in finding routes depends on the amount and correctness of the damaged road data entered by the admin. In further development, special attention is paid to admin validation, data security, integration with location service provider APIs, and performance optimization. Richer reporting features, multi-platform support, and notifications to users could also be valuable additions. With good documentation and thorough testing, the system is expected to provide an effective solution and be well received by users.*

Keywords: *Algoritma A-Star, Locations, Nodes, Broken Roads, Maps*

Abstrak. Dalam kehidupan sehari-hari, kita sering dihadapkan pada tantangan jalan rusak yang dapat mempengaruhi perjalanan kita. Untuk mengatasi masalah ini, sebuah sistem peta berbasis web telah dikembangkan. Sistem ini menggunakan data jalan dari OpenStreetMap dan algoritma A-star untuk mencari rute terpendek yang menghindari jalan rusak. Website ini dibangun dengan menggunakan JavaScript, PHP, HTML, dan CSS, dengan dukungan database MySQL yang menyimpan informasi jalan rusak dan akun pengguna. Proses penggunaan sistem dimulai dengan registrasi dan login akun. Pengguna dapat melihat peta pada halaman utama, melaporkan jalan rusak dengan foto bukti yang dilengkapi watermark waktu dan data koordinat, serta melihat rute awal mereka. Admin kemudian memvalidasi data jalan rusak, dan sistem secara otomatis menyusun rute alternatif yang menghindari jalan rusak yang telah diverifikasi. Tingkat keakuratan algoritma A-star dalam mencari rute tergantung pada jumlah dan kebenaran data jalan rusak yang dimasukkan oleh admin. Dalam pengembangan lebih lanjut, perhatian khusus diberikan pada validasi admin, keamanan data, integrasi dengan API penyedia layanan lokasi, dan optimalisasi kinerja. Fitur pelaporan yang lebih kaya, dukungan multi-platform, dan pemberitahuan kepada pengguna juga dapat menjadi tambahan yang berharga. Dengan adanya dokumentasi yang baik dan uji coba menyeluruh, diharapkan sistem ini dapat memberikan solusi yang efektif dan diterima dengan baik oleh pengguna.

Kata kunci: Algoritma A-Star, Lokasi, Node, Jalan Rusak, Maps

PENDAHULUAN

Pada kehidupan sehari-hari, kita sering menggunakan jalan sebagai prasarana transportasi untuk sampai ke tujuan. Namun, tidak semua jalan memiliki kondisi yang sama. Salah satu tantangan yang sering kita hadapi adalah adanya jalanan yang rusak sehingga kita harus berpikir dua kali sebelum memilih rute. Melewati jalan rusak dapat menimbulkan berbagai kerugian yang dapat mempengaruhi kenyamanan, keamanan dan efisiensi perjalanan.

Kerusakan jalan raya dapat menimbulkan guncangan dan getaran pada kendaraan yang dapat mengakibatkan kerusakan pada suspensi, roda, dan komponen lainnya. Hal ini dapat menyebabkan biaya perbaikan yang tidak terduga dan mengganggu penggunaan normal kendaraan. Selain itu, berkendara di jalan yang rusak juga dapat meningkatkan risiko kecelakaan lalu lintas, terutama jika pengemudi tidak mengantisipasi dengan baik rintangan atau lubang di jalan. Telah banyak korban jatuh akibat jalan yang rusak tersebut, sehingga menimbulkan kerugian tidak hanya terhadap rusaknya kendaraan namun juga telah menghilangkan nyawa seseorang (Siregar M. T., 2020). Oleh karena itu kita menghadapi jalanan yang rusak, terkadang kita harus mencari alternatif lain yang lebih cepat dan efisien. dengan kemajuan teknologi telah memberikan solusi untuk masalah ini dalam bentuk sistem peta yang dilengkapi dengan fitur pengalihan arah jalan.

Menurut Penelitian dari (Yudha, Supian, & Napitupulu, 2022) yang berjudul *Optimalization Route to Tourism Places in West Java Using A-STAR Algorithm* mendapatkan hasil penelitian bahwa “Algoritma A-STAR dapat digunakan ketika informasi tentang mengenai jarak dan kepadatan lalu lintas telah diperoleh. Dalam hal ini, kepadatan lalu lintas menjadi nilai-nilai heuristic yang dapat digunakan sebagai dasar pertimbangan. Dapat disimpulkan bahwa rute optimal yang dipengaruhi oleh jarak dan kepadatan lalu lintas dapat dicari dengan menggunakan Algoritma A-STAR. Penelitian ini juga dapat membantu wisatawan untuk melewati rute yang paling optimal yang paling optimal menuju tempat-tempat pariwisata di Provinsi Jawa Barat.” Dengan hasil penelitian tersebut maka penulis mengangkat penelitian dengan Algoritma yang sama namun berbeda bidang penelitian. Penulis mengharapkan dengan penggunaan Algoritma A-star dapat memperoleh rute terpendek dalam pencarian jalan menghindari jalan rusak.

KAJIAN TEORITIS

Location Based Services

Location based Services adalah sebuah Location-based Service atau secara umum sebagai teknologi yang digunakan untuk menemukan lokasi dari berbagai perangkat teknologi contohnya GPS. Location Based Service merupakan Location-based Service atau istilah umum yang sering digunakan untuk menggambarkan teknologi yang digunakan untuk menemukan lokasi perangkat yang pengguna gunakan. Layanan ini menggunakan teknologi Global Positioning Service (GPS) dan Cell Based Location dari Google (Siregar, Koryanto, & Faizah, 2023). LBS memiliki beberapa komponen yaitu mobile devices, communication network, position component, dan service and content provider.

MySQL

MySQL adalah sebuah implementasi dari sistem manajemen basis data relasional (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (General Public License). Setiap pengguna dapat secara bebas menggunakan MySQL, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial (Hidayat, Yani, Rusidi, & Saadulloh, 2019).

Algoritma A* (A Star)

Algoritma A* (atau A-star) adalah algoritma pencarian jalur terpendek yang digunakan dalam pemrograman komputer untuk menemukan jalur terpendek antara dua titik dalam sebuah graf berbobot. Algoritma ini menggunakan pendekatan heuristic untuk mencari jalur terpendek dengan mempertimbangkan biaya yang diperlukan untuk mencapai setiap simpul dalam graf, serta perkiraan jarak yang tersisa dari setiap simpul ke tujuan. Algoritma A* dapat menentukan rute (jalur) terbaik dari titik awal (start) menuju titik akhir (finish) dengan hambatan-hambatan yang diberikan di setiap rute. Dari hasil pengujian, rute yang ditemukan merupakan rute yang terbaik dengan nilai $f(n)$ terkecil dibandingkan dengan rute-rute (jalur-jalur) lainnya (Antara Dalem, 2018). Fungsi heuristic yang terdapat pada Algoritma A-STAR untuk menghitung nilai estimasi dari sebuah node yang telah dilalui adalah.

$$F(n) = G(n) + H(n)$$

Dimana:

$F(n)$: total estimasi biaya

$G(n)$: biaya dari simpul awal ke simpul ke- n

$H(n)$: estimasi biaya untuk tiba di tujuan dari node ke- n

Algoritma A-STAR menggunakan dua buah list, yaitu OPEN dan CLOSED. OPEN adalah sebuah list yang digunakan untuk menyimpan node/simpul yang telah dihitung dan nilai heuristic-nya telah juga telah dihitung tetapi belum terpilih sebagai simpul terbaik ($F(n)$). Dengan kata lain, daftar ini berisi node-node yang masih memiliki kesempatan untuk dipilih sebagai node terbaik. CLOSED adalah list untuk menyimpan node-node yang telah dihitung dan telah terpilih sebagai node terbaik node ($F(n)$) (Yudha, Supian, & Napitupulu, 2022).

Open Street Map (OSM)

OpenStreetMap (OSM) adalah proyek kolaboratif pemetaan global yang dimulai pada tahun 2004. OpenStreetMap (OSM) adalah sebuah proyek berbasis web untuk membuat peta seluruh dunia yang gratis dan terbuka, dibangun sepenuhnya oleh sukarelawan dengan melakukan survei menggunakan GPS, digitalisasi citra satelit, dan mengumpulkan serta membebaskan data geografis yang tersedia di publik (Syaifudin, Puspitasari, & Nugroho,

2019). Beberapa Data yang dikumpulkan adalah informasi tentang jalan, bangunan, sungai, taman, dan banyak elemen geografis lainnya. OSM hingga saat ini terus berkembang dengan kontribusi dari ribuan orang di seluruh dunia, hal ini menjadikannya OSM sebagai salah satu proyek pemetaan terbesar dan paling sukses yang ada.

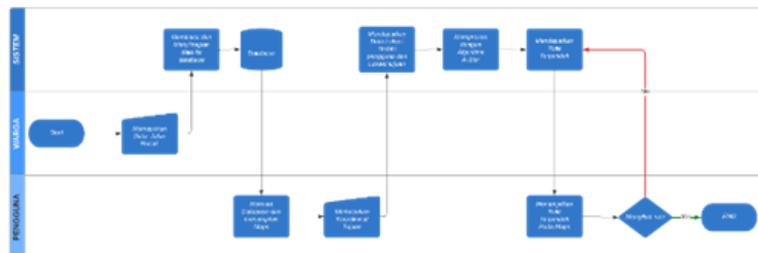
Sistem Informasi Geografis

Sistem Informasi Geografis (SIG) memiliki peran penting dalam mengelola, menganalisis, dan menampilkan informasi yang berhubungan dengan lokasi atau ruang geografis. Sistem informasi geografis digunakan untuk menangani data spasial atau data tentang keruangan, sistem seperti ini banyak digunakan antara lain untuk pemetaan tanah dan agrikultur, arkeologi dan jaringan listrik (Ikhsan, Najib, & Ulum, Sistem Informasi Geografis Toko Distro Berdasarkan Rating Kota Bandar Lampung Berbasis Web, 2020).

METODE PENELITIAN

Perencanaan Alur Sistem

Hal pertama yang wajib dilakukan sebelum penciptaan aplikasi yaitu pembuatan alur jalannya sistem aplikasi dengan tujuan sebagai pedoman logika programmer untuk membuat aplikasi sesuai dengan tujuan utama aplikasi dirancang. Berikut sistem yang telah dibuat oleh penulis dapat dilihat gambar.

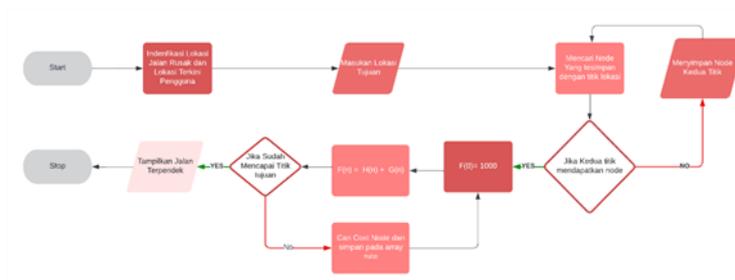


Gambar 1. Alur Sistem

Dari gambar diatas dapat jelaskan alur dari sistem yang akan dibuat sebagai besar dijelaskan bahwa awal para warga sekitar dapat memasukan data jalan rusak di sekitar mereka yang akan disimpan database oleh sistem. Data tersebut akan di akses jika ada pengguna aplikasi menggunakan untuk mencari jalan yang akan di proses oleh algoritma A-Star untuk mendapatkan rute terdekat pada maps yang tampil pada Sistem.

Alur Algoritma A-Star

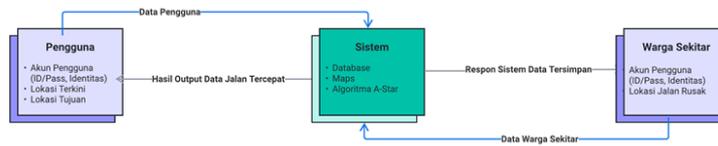
Selanjutnya adalah pemaparan alur jalannya A-Star dalam pencarian rute terdekat untuk mencapai lokasi tujuan tanpa melalui jalan rusak. Alur tersebut dapat dilihat pada gambar.



Gambar 2. Alur Algoritma

Context Diagram

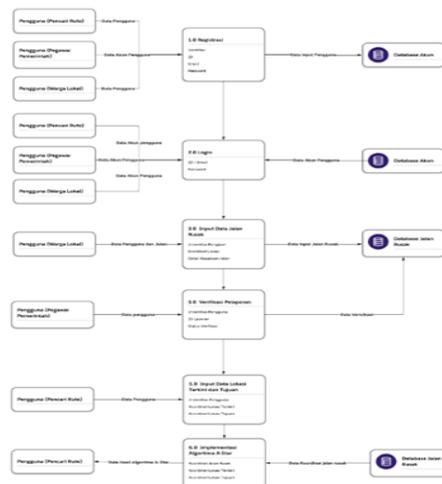
Context Diagram adalah diagram yang terdiri dari suatu proses dan menggambarkan ruang lingkup yang berada dalam sistem. Context Diagram sering disebut juga dengan Level-0 dan menjadi penentu utama pada sebuah sistem yang dirancang dalam bentuk Data Flow Diagram. Penulis telah Membuat Context Diagram yang dapat dilihat pada gambar.



Gambar 3. Context Diagram

Data Flow Diagram Sistem

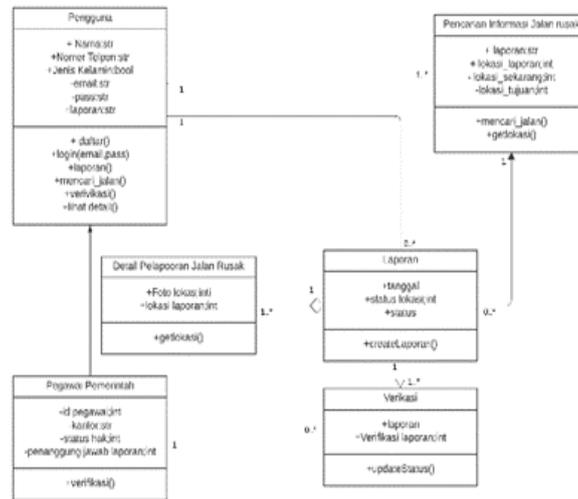
DFD adalah sebuah diagram yang menjelaskan alur data atau proses keseluruhan yang berjalan dalam sistem aplikasi. Proses ini digambarkan dengan beberapa simbol yang telah ditentukan. DFD Sistem pencarian Rute terdekat dapat dilihat pada gambar.



Gambar 4. Data Flow Diagram

Class Diagram

Dalam rekayasa perangkat lunak, diagram kelas dalam Unified Modeling Language (UML) adalah jenis diagram struktur statis yang menggambarkan struktur sistem dengan menunjukkan kelas-kelas sistem, atribut-atributnya, operasi-operasinya (atau metode-metode nya), dan hubungan-hubungan di antara objek-objek. Untuk Class Diagram Aplikasi Penghinder Jalan Rusak dapat dilihat pada gambar.



Gambar 5. Class Diagram

Pengujian Simulasi Algoritma A-Star

Pada pengujian Algoritma A-Star menggunakan metode simulasi yang memiliki Beberapa Skenario yang masing-masing memiliki Validasi hasil tersendiri dengan kondisi yang telah ditentukan sebagai Berikut.

Tabel 1. Pengujian Simulasi Algoritma A-Star

| No | Lokasi Awal | Lokasi Tujuan | Hasil yang diinginkan |
|----|-------------------|--------------------------------------|--|
| 1. | Urangagung Jedong | Alun- Alun Sidoarjo (Sidoarjo) | Rute sebelumnya sudah dimunculkan akan berubah sesuai algoritma mendapatkan jalan baru |
| 2. | Urangagung Jedong | Lumpur Lapindo (Porong,Sidoarjo) | Rute sebelumnya sudah dimunculkan akan berubah sesuai algoritma mendapatkan jalan baru |
| 3. | Urangagung Jedong | Terminal Bungurasih (Waru, Sidoarjo) | Rute sebelumnya sudah dimunculkan akan berubah sesuai algoritma mendapatkan jalan baru |

HASIL DAN PEMBAHASAN

Pengolahan Data OSM

Pada penelitian ini membutuhkan data peta atau jalan, maka tahapan awalan adalah mempersiapkan data peta yang nantinya digunakan untuk menentukan rute. Dalam pencarian data peta, penulis mengunduh peta Openstreet dari situs <https://download.geofabrik.de/>. Peta yang diunduh adalah peta pulau jawa. File peta yang diunduh oleh penulis berformat .osm.pbf yang diharuskan diekstrak menjadi data xml berformat .OSM untuk dapat diolah pada tahap selanjutnya.



Gambar 6. Laman Geofabric

Untuk mengekstrak data peta penulis menggunakan salah satu *command line* dari OpenStreetMap yaitu Osmosis. Penggunaan Osmosis terbilang mudah karena hanya mendownload file, kemudian file diletakkan pada Program file pada perangkat windows pengguna. Untuk menjalankan Osmosis perlu penggunaan terminal atau CMD. Setelah Osmosis dapat dijalankan maka selanjutnya kita masukan *Command line* yang digunakan untuk mengekstrak data peta untuk menjadi data xml dengan kode “**osmosis --read-pbf file='input.pbf' --bounding-box left=112.5079 bottom=-7.5732 right=112.9151 top=-7.3512 --write-xml file='output.osm'**” dan jalankan untuk menghasilkan data peta seperti pada gambar.

```
C:\Program Files (x86)\osmosis>osmosis --read-pbf file="C:\Users\user\Downloads\java-1125079.pbf" --bounding-box left=112.5079 bottom=-7.5732 right=112.9151 top=-7.3512 --write-xml file="C:\Users\user\Downloads\varjo.osm"
Nov 20, 2023 12:01:52 PM org.openstreetmap.osmosis.core.Osmosis run
INFO: Osmosis Version: 0.48.2
Nov 20, 2023 12:01:53 PM org.openstreetmap.osmosis.core.Osmosis run
INFO: Preparing pipeline
Nov 20, 2023 12:01:53 PM org.openstreetmap.osmosis.core.Osmosis run
INFO: Pipeline executing, waiting for completion
Nov 20, 2023 12:01:54 PM org.openstreetmap.osmosis.core.Osmosis run
INFO: Launching pipeline execution
Nov 20, 2023 12:01:54 PM org.openstreetmap.osmosis.core.Osmosis run
INFO: Pipeline completed
Nov 20, 2023 12:02:04 PM org.openstreetmap.osmosis.core.Osmosis run
INFO: Total execution time: 61668 milliseconds
C:\Program Files (x86)\osmosis>
```

Gambar 7. Kode Osmosis

Setelah penulis mendapatkan data peta yang berformat osm , selanjutnya penulis mengolah data tersebut untuk dijadikan data array berformat Json. Penulis menggunakan library *xml2js* untuk mengolah data berformat xml dan diproses menjadi array dengan *source code* berikut.

```
1 const xml2js = require('xml2js');
2 const fs = require('fs');
3 const osmParser = require('./osm-parser');
4 const parser = new xml2js.Parser();
5
6 fs.readFile('./junda.osm', (err, data) => {
7   parser.parseString(data, (err, result) => {
8     osmParser.parseNodes(result.osm);
9     osmParser.createGraph(result.osm);
10    const nodes = osmParser.getNodes();
11    const data = {
12      nodes: nodes
13    }
14  });
15   fs.writeFileSync('./osmway.json', JSON.stringify(data));
16 });
17 });
```

Gambar 8. Kode Membuat Json

Semua fungsi terdapat pada file *Osm-parse.js* yang berisi *parseNodes* dan *createNodes* yang memiliki fungsi tersendiri. Untuk *parseNodes* digunakan untuk mengubah data xml menjadi array bernama *nodes* yang disusun oleh *source code* berikut.

```

1 let nodes = {};
2
3 function parseNodes(osm) {
4   nodes = {};
5
6   if (osm.node) {
7     for (let n of osm.node) {
8       let attr = n[""];
9       nodes[attr["id"]] = {
10        lat: parseFloat(attr["lat"]),
11        lon: parseFloat(attr["lon"]),
12        con: new Set()
13      };
14    }
15  }
16
17  return nodes;
18 };

```

Gambar 9. Membuat Node

Selanjutnya adalah mencari koneksi node yang akan disimpan variabel *con* yang akan diisi oleh hasil *source code* berikut.

```

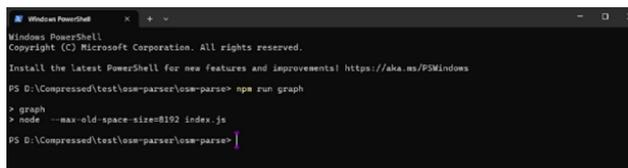
1 function createGraphCon() {
2   if (!osm.way) return;
3   for (let way of osm.way) {
4     let highway = false;
5     let junction = false;
6     let oneway = false;
7
8     for (let node of way.nodes) {
9       let attr = node[""];
10      if (attr["id"] == "highway") {
11        highway = true;
12      } else if (attr["id"] == "junction") {
13        junction = true;
14      } else if (attr["id"] == "oneway" || attr["id"] == "one") {
15        oneway = true;
16      }
17    }
18
19    if (junction) {
20      for (let i = 0; i < way.nodes.length; i++) {
21        let curr = way.nodes[i];
22        if (nodes[curr] || nodes[curr].con) {
23          for (let j = 0; j < way.nodes.length; j++) {
24            let next = way.nodes[j];
25            if (nodes[next] || nodes[next].con) {
26              if (curr.con.add(nodes[next].con) || nodes[next].con.add(curr)) {
27                nodes[curr].con.add(next);
28                nodes[next].con.add(curr);
29              }
30            }
31          }
32        }
33      }
34    } else if (highway) {
35      if (oneway) {
36        for (let i = 0; i < way.nodes.length; i++) {
37          let curr = way.nodes[i];
38          if (nodes[curr] || nodes[curr].con) {
39            for (let j = 1; j < way.nodes.length; j++) {
40              let next = way.nodes[j];
41              if (nodes[next] || nodes[next].con) {
42                nodes[curr].con.add(next);
43                nodes[next].con.add(curr);
44              }
45            }
46          }
47        }
48      } else {
49        for (let i = 1; i < way.nodes.length; i++) {
50          let prev = way.nodes[i-1];
51          let curr = way.nodes[i];
52          if (nodes[prev] || nodes[prev].con || nodes[curr] || nodes[curr].con) {
53            nodes[prev].con.add(curr);
54            nodes[curr].con.add(prev);
55          }
56        }
57      }
58    }
59  }
60 }

```

Gambar 10. Kode Membuat Data Con

Pertama kita ambil data *osm.way* yang berisi data jalan yang terdapat pada file *.osm*. setelah mendapatkan data *osm.way* dikelompokkan menjadi 2 data yang bernilai *junction*, *oneway* dan *highway* untuk membenarkan atau bernilai *true* fungsi penambahan node jalan antara jalan tikungan, jalan biasa dan jalan satu arah. Setelah fungsi dibenarkan atau bernilai *true* maka data diolah secara berbeda, untuk data *junction* diolah dengan jika ada data selanjutnya dan tidak sama dengan data sekarang maka nilai data selanjutnya akan di simpan pada *con* pada array data jalan sekarang. Untuk data *highway* bercabang menjadi 2 kelompok yaitu *oneway* dan jalan biasa, jika *oneway* bernilai *true* maka data yang diambil adalah data sekarang dan data node selanjutnya kemudian data node selanjutnya akan disimpan pada *con* data node sekarang. Berbeda dengan jalan biasa, untuk jalan biasa akan mengambil data sekarang dan data sebelumnya. Kemudian data sekarang akan di simpan pada *con* data sebelumnya dan untuk data node sebelumnya di simpan pada *con* data sekarang Lalu hasil return disimpan pada variabel *nodes* dan dijalankan fungsi “*fs.writeFileSync("./map.json", JSON.stringify(data));*” untuk dicetak sebagai file json. Untuk menjalankan program javascript

dibutuhkan bantuan node.js maka perlu menginstall node.js dan menjalankan pada cmd seperti gambar berikut.



```

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS D:\Compressed\test\osm-parser\osm-parser> npm run graph
> graph
> node --max-old-space-size=8192 index.js
PS D:\Compressed\test\osm-parser\osm-parser>

```

Gambar 11. Hasil Output Index.js

Halaman Pendaftaran dan Masuk Akun

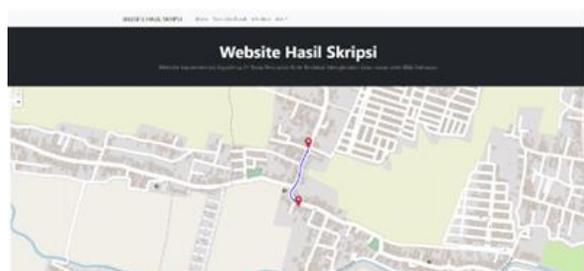
Setelah mengelola data peta maka selanjutnya adalah pembuatan halaman Pendaftaran dan masuk akun untuk pengguna sistem. Untuk data pendaftaran membutuhkan data email, password, nomor, nama, dan jenis pengguna (warga atau pegawai). Dan untuk data login hanya membutuhkan email dan password. Untuk tampilan dapat dilihat pada gambar berikut.



Gambar 12. Tampilan Halaman Login dan Pendaftaran

Halaman Rute Jalan

Setelah halaman masuk dan pendaftaran akun telah dibuat dan berjalan sesuai rancangan penulis maka tahap selanjutnya adalah pembuatan halaman utama dari website skripsi ini yaitu pengimplementasian algoritma A* untuk mencari rute terpendek. Untuk tampilan dari halaman ini dapat dilihat dari gambar berikut.



Gambar 13. Tampilan Halaman Rute Jalan

Halaman Memasukan Data Jalan Rusak

Setelah halaman untuk mencari rute terpendek sudah dibuat maka tahap selanjutnya adalah pembuatan fitur pendukung palung utama pada penelitian ini adalah memasukan data jalan rusak ke data map yang telah dibuat. Fitur halaman ini sangat mendukung tujuan utama dari penelitian ini untuk menghindari jalan rusak. Untuk tampilan dari halaman dapat dilihat pada gambar.



Gambar 14. Halaman Input Data Jalan Rusak

Halaman Validasi Data Jalan Rusak

Setelah data jalan rusak telah disimpan pada database maka perlu validasi dari pegawai untuk menyimpan data jalan rusak tersebut ke Json data map. Penyimpanan data ini adalah mengubah nilai status lokasi id yang terdaftar pada data jalan rusak dari status bernilai 1 akan berubah menjadi bernilai 0 supaya tidak terdeteksi dalam pencarian children node. Untuk tampilan validasi data jalan rusak dapat dilihat pada gambar.



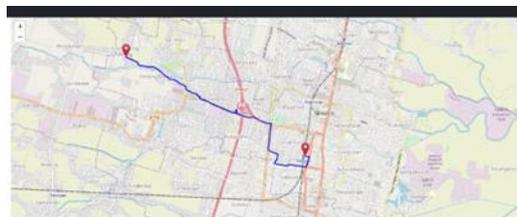
Gambar 15. Tampilan Validasi Data Jalan

Pengujian Algoritma

Setelah pengujian sistem website algoritma a-star mencari rute pendek menghindari jalan rusak berdasarkan skenario yang telah dibuat oleh penulis sebelumnya. Untuk percobaan per point dapat dilihat dari beberapa sub bab berikut.

A. Percobaan 1

Percobaan pertama penulis akan menelusuri rute pendek dari rumah penulis menuju alun-alun sidoarjo dengan mendapatkan rute jalan seperti gambar berikut.



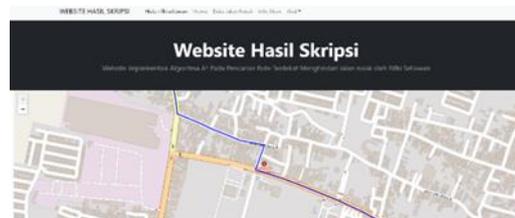
Gambar 16. Rute Awal Pengujian 1

Setelah penulis menelusuri rute tersebut bahwa jalan rute tersebut terdapat jalan rusak dengan koordinat $[-7.441905, 112.686234]$ dengan kriteria jalan gang tanpa beraspal. Maka penulis menambahkan jalan rusak dengan gambar bukti. Maka data jalan rusak akan muncul pada halaman data jalan rusak seperti berikut.



Gambar 17. Lokasi Jalan Rusak Pengujian 1

Dikarenakan pada jalan tersebut terdapat tanda lokasi berwarna biru menandakan bahwa jalan tersebut rusak atau tidak layak dilewati. Maka jika dicoba lagi mencari rute pendek mendapatkan hasil berbeda sebagai berikut.

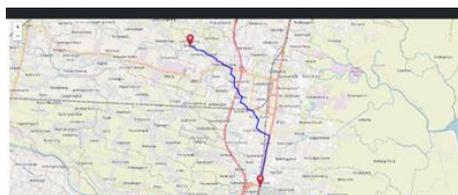


Gambar 18. Hasil Algoritma Pengujian 1

Pada Gambar diatas terdapat dua jalan berwarna biru dan merah yang memiliki arti tersendiri. Untuk jalan berwarna merah adalah tanda jalan untuk hasil algoritma dari rute jalan sebelum ditambahkan data jalan rusak, Sedangkan jalan berwarna biru adalah tanda jalan untuk hasil algoritma dari rute jalan yang telah ditambahkan data jalan rusak. Dari 2 rute tersebut menjelaskan bahwa rute akan berubah jika ditambahkan data jalan rusak.

B. Percobaan 2

Percobaan pertama penulis akan menelusuri rute pendek dari rumah penulis menuju wisata Lumpur Lapindo yang terletak pada kecamatan porong dengan mendapatkan rute jalan seperti gambar berikut.



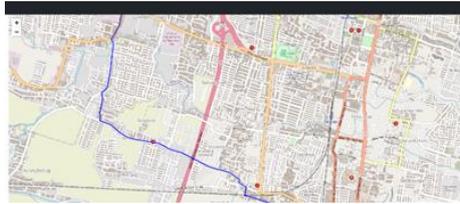
Gambar 19. Rute Awal Pengujian 2

Setelah penulis menelusuri rute tersebut bahwa jalan rute tersebut terdapat jalan rusak dengan beberapa koordinat $[-7.441905, 112.686234]$ dengan kriteria jalan gang tanpa beraspal. Maka penulis menambahkan jalan rusak dengan gambar bukti. Maka data jalan rusak akan muncul pada halaman data jalan rusak seperti berikut.



Gambar 20. Lokasi Jalan Rusak Pengujian 2

Dikarenakan pada jalan tersebut terdapat tanda lokasi berwarna biru menandakan bahwa jalan tersebut rusak atau tidak layak dilewati. Maka jika dicoba lagi mencari rute pendek mendapatkan hasil berbeda sebagai berikut.



Gambar 21. Hasil Algoritma Pengujian 2

Pada Gambar diatas terdapat dua jalan berwarna biru dan merah yang memiliki arti tersendiri. Untuk jalan berwarna merah adalah tanda jalan untuk hasil algoritma dari rute jalan sebelum ditambahkan data jalan rusak, Sedangkan jalan berwarna biru adalah tanda jalan untuk hasil algoritma dari rute jalan yang telah ditambahkan data jalan rusak. Dari 2 rute tersebut menjelaskan bahwa rute akan berubah jika ditambahkan data jalan rusak.

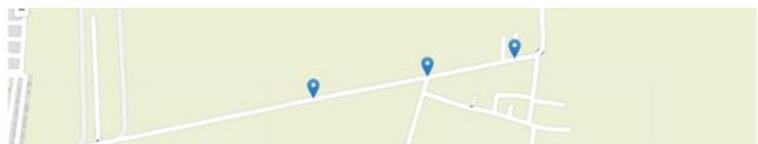
C. Percobaan 3

Pengujian pertama penulis akan menelusuri rute pendek dari rumah penulis menuju Terminal Bungurasih yang terletak daerah kecamatan waru dengan mendapatkan rute jalan seperti Gambar berikut.



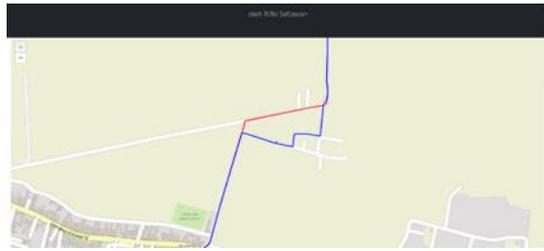
Gambar 22. Rute Awal Pengujian 3

Setelah penulis menelusuri rute tersebut bahwa jalan rute tersebut terdapat jalan rusak dengan beberapa koordinat $[-7.403697, 112.682256]$ dan koordinat $[-7.4027967, 112.6865405]$.Maka penulis menambahkan jalan rusak dengan gambar bukti. Maka data jalan rusak akan muncul pada halaman data jalan rusak seperti Gambar berikut.



Gambar 23. Lokasi Jalan Rusak Pengujian 3

Dikarenakan pada jalan tersebut terdapat tanda lokasi berwarna biru menandakan bahwa jalan tersebut rusak atau tidak layak dilewati. Maka jika dicoba lagi mencari rute pendek mendapatkan hasil berbeda sebagai Gambar Berikut.



Gambar 24. Hasil Algoritma Pengujian 3

Pada Gambar diatas terdapat dua jalan berwarna biru dan merah yang memiliki arti tersendiri. Untuk jalan berwarna merah adalah tanda jalan untuk hasil algoritma dari rute jalan sebelum ditambahkan data jalan rusak, Sedangkan jalan berwarna biru adalah tanda jalan untuk hasil algoritma dari rute jalan yang telah ditambahkan data jalan rusak. Dari 2 rute tersebut menjelaskan bahwa rute akan berubah jika ditambahkan data jalan rusak.

KESIMPULAN DAN SARAN

Kesimpulan dari penelitian skripsi yang berjudul “Implementasi Algoritma A-Star Pada Pencarian Rute Terdekat Menghindari Jalan rusak” adalah Algoritma A-star pada website Implementasi Algoritma A-Star Pada Pencarian Rute Terdekat Menghindari Jalan rusak berjalan lumayan lancar dengan dikembangkan oleh bahasa pemrograman PHP, JavaScript, MySQL dan didukung oleh beberapa library JavaScript seperti Leaflet dan Xml2js. Dengan beberapa dukungan tool tersebut Algoritma A-star dapat memunculkan rute menghindari jalan rusak dengan satu line berwarna biru. Berdasarkan hasil dari lima percobaan yang dilakukan, diperoleh informasi bahwa algoritma A-Star memiliki tingkat keakuratan yang memuaskan. dari total tiga percobaan, tiga di antaranya berhasil memberikan solusi yang dapat menghindari jalan rusak dengan sukses. hasil ini menunjukkan bahwa algoritma A-Star cenderung efektif dalam perhitungan untuk menghindari jalan rusak dan memberikan solusi rute terdekat yang aman.

DAFTAR REFERENSI

- Antara Dalem, I. W. (2018). PENERAPAN ALGORITMA A* (STAR) MENGGUNAKAN GRAPH UNTUK. *JURNAL RESISTOR*, 41-47.
- Enterprise, J. (2018). *HTML, PHP, dan MySQL*. Jakarta: PT Elex Media Komputindo.
- Hidayat, A., Yani, A., Rusidi, & Saadulloh. (2019). MEMBANGUN WEBSITE SMA PGRI GUNUNG RAYA RANAU MENGGUNAKAN PHP DAN MYSQL. *JTIM: Jurnal Teknik Informatika Mahakarya*, 41-52.

- Ikhsan, A., Najib, M., & Ulum, F. (2020). SISTEM INFORMASI GEOGRAFIS TOKO DISTRO BERDASARKAN RATING KOTA BANDAR LAMPUNG BERBASIS WEB. *Jurnal Teknologi dan Sistem Informasi (JTISI)*, 71-79.
- Paunsyah, H., Mubarak, H., & Shofa, R. N. (2019). Penentuan Jalur Terpendek Menggunakan Google Maps API pada Sistem Informasi Geografis Panti Sosial di kota Tasikmalaya. *Innovation in Research of Informatics (INNOVATICS)*, 1-6.
- Siregar, D. R., Koryanto, L., & Faizah, N. (2023). Aplikasi Pencarian Hotel di Kota Jakarta Berbasis Android dengan Metode Location Based Service (LBS) Menggunakan Android Studio. *Computer Journal*, 64-72.
- Siregar, M. T. (2020). Upaya yang dapat Dilakukan oleh Korban/Pengguna Jalan Meminta Pertanggungjawaban Pidana Penyelenggara Jalan atas Terjadinya Kecelakaan Akibat Jalan Rusak. *Jurnal Edutech*, 36-44.
- Syaifudin, Y. W., Puspitasari, D., & Nugroho, M. A. (2019). Penentuan Jarak Terpendek Menggunakan Metode Dijkstra Pada. *SMATIKA*, 45 - 50.
- Yudha, M. H., Supian, S., & Napitupulu, H. (2022). Optimalization Route to Tourism Places in West Java Using A-STAR . *CAUCHY –Jurnal Matematika Murni dan Aplikasi*, 464-473.