

**PERBANDINGAN ALGORITMA FLOYD-WARSHALL DAN DIJKSTRA
UNTUK MENENTUKAN RUTE RUMAH SAKIT TERDEKAT JALUR
EVAKUASI KECELAKAAN LALU LINTAS BERBASIS WEB**

***COMPARISON OF FLOYD-WARSHALL AND DIJKSTRA ALGORITHM FOR
DETERMINING THE NEAREST HOSPITAL ROUTE WEB-BASED TRAFFIC
ACCIDENT EVACUATION ROUTES***

Agung Dwika Rudiyanto, M. Iwan Wahyuddin, Andrianingsih

Informatika, Fakultas Teknologi Komunikasi dan Informatika, Universitas Nasional
agungdwika99@gmail.com

ABSTRACT

Traffic accidents frequently happens in streets are dealing with of a crisis circumstance to be done as before long as conceivable. Taking care of is done amid the mishap activity is to clear the casualties specifically to the hospital. But, to did the departure, officers having numerous deterrents in deciding course to urge to the hospital. As the development of the innovation, the assurance of in these obstacles route to be completed employing a parcel of pathfinding algorithm or look the shortest line, a few of whom by utilizing the strategy that's utilizing look course Dijkstra algorithm and Floyd-Warshall algorithm. Inquire about points to get it the calculations floyd-warshall with dijkstra in deciding most brief course chosen choice for officers to bolster clearing mischance casualties.

Keywords: *Dijkstra, Floyd-Warshall, Route, Accident, Hospital.*

ABSTRAK

Kecelakaan lalu lintas yang sering terjadi di jalan merupakan situasi darurat yang harus dilakukan penanganan secepatnya. Penanganan yang wajib dilakukan saat terjadi kecelakaan lalu lintas adalah melakukan evakuasi korban kecelakaan langsung ke rumah sakit. Namun, untuk melakukan evakuasi, petugas memiliki banyak kendala dalam menentukan rute jalan untuk sampai ke rumah sakit. Seiring berkembangnya teknologi, kendala penentuan rute tersebut saat ini sudah dapat diselesaikan menggunakan berbagai algoritma pathfinding atau pencarian jalur terpendek, beberapa diantaranya dengan menggunakan metode pencarian rute yaitu menggunakan algoritma Floyd-Warshall dan Dijkstra. Penelitian ini bertujuan untuk mengetahui perbandingan antara algoritma Floyd-Warshall dengan Dijkstra dalam menentukan rute terpendek yang akan dipilih sebagai pendukung keputusan bagi petugas untuk melakukan evakuasi korban kecelakaan.

Kata Kunci: Dijkstra, Floyd-Warshall, Jalur, Kecelakaan, Rumah Sakit.

PENDAHULUAN

Salah satu kejadian yang banyak ditemui dalam kehidupan sehari-hari di jalan raya ialah insiden kecelakaan, kecelakaan pada lalu lintas merupakan peristiwa yang terjadi pada pengguna jalan secara tidak terduga atau tiba tiba serta tidak diinginkan sehingga tidak mungkin untuk diprediksi mengenai kapan terjadinya dan dimana tepat kejadian tersebut berlangsung, kecelakaan lalu lintas setidaknya terjadi dengan kendaraan tunggal atau melibatkan pengguna jalan lain yang

dapat mengakibatkan terjadinya kerugian pada korban seperti cedera, kecacatan, trauma bahkan kematian. Saat terjadinya kecelakaan penanganan yang tepat adalah penanganan yang mengutamakan keselamatan dari korban kecelakaan. Penanganan yang tepat dan yang paling utama adalah langsung membawa korban kecelakaan ke rumah sakit, namun untuk membawa korban langsung ke rumah sakit diperlukan rute tercepat untuk sampai ke rumah sakit karena saat melakukan evakuasi korban kecelakaan waktu menjadi faktor

penting dalam penanganan. Maka dari itu penerapan algoritma serta sistem informasi geografis sangat diperlukan untuk menyelesaikan permasalahan ini.

Pada penelitian sebelumnya penelitian yang dilakukan oleh Nawagusti dkk., (2018) mengenai implementasi Algoritma Floyd-Warshall pada aplikasi penentuan rute terpendek untuk mencari lokasi BTS (Base Tower Station), dalam penelitian tersebut didapatkan hasil rute terpendek menuju BTS 18, BTS 21 dan BTS 46 adalah 1-4-3-2 dengan jarak tempuh sebesar 10.9 km sehingga dalam penentuan rute terpendek yang diterapkan dalam aplikasi sudah berhasil, karena didapatkan acuan yang sama antara hasil dari penentuan rute terpendek dalam aplikasi dengan hasil perhitungan manual yang dihasilkan berdasarkan teori Floyd Warshall. Lalu dalam penelitian Ardana & Saputra membahas tentang Penerapan Algoritma Dijkstra pada Aplikasi Pencarian Rute Bus Trans Semarang. Pada penelitian ini dihasilkan sebuah aplikasi untuk mencari rute bus trans Semarang menggunakan algoritma Dijkstra. Algoritma Dijkstra digunakan dengan tujuan menentukan titik berhenti terdekat serta titik berhenti untuk perpindahan koridor bus. Dalam penelitian ini aplikasi mampu menghasilkan informasi berkaitan dengan rute perjalanan bus serta titik berhenti dari awal sampai akhir pemberhentian sehingga algoritma Dijkstra berhasil diimplementasikan pada aplikasi pencarian rute bus trans Semarang (Ardana & Saputra, 2016). Selain itu dalam menentukan rute terpendek pada pencarian apotek menerapkan algoritma Floyd-Warshall yang dilakukan oleh Ari Muzakir didapatkan hasil terpendek pada setiap jalur alternatif oleh algoritma Floyd-Warshall dalam pencarian apotek menjadi hasil utama dalam penelitian

tersebut (Muzakir, 2020). Selain itu dalam penelitian penentuan rute wisata terdekat di pulau Lombok Nusa Tenggara Barat dengan menerapkan algoritma Dijkstra yang dilakukan L. Masyudi & Wahyu didapatkan hasil jarak tempuh terdekat dari 38 tujuan wisata yang ada di pulau Lombok dan rute destinasi sampai pada tujuan akhir (Masyudi & Khalik, 2018).

Algoritma Floyd-Warshall termasuk dalam jenis pemrograman dinamis. Algoritma Floyd Warshall adalah sebuah metode dalam pemecahan masalah dengan memandang solusi yang diperoleh sebagai pengambilan keputusan yang saling terkait. Dalam arti tiap solusi yang dibentuk dari solusi tahap sebelumnya ada kemungkinan solusi tersebut lebih dari satu (Kriswanto, 2014).

```

1. function fw(int[1..n,1..n] graph)
2. {
3. // Inisialisasi
4. var int[1..n,1..n] jarak := graph
5. var int[1..n,1..n] sebelum
6. for i from 1 to n
7. for j from 1 to n
8. if jarak[i,j] < Tak-hingga
9. sebelum[i,j] := i
10.
11. // Perulangan utama pada algoritma
12. for k from 1 to n
13. for i from 1 to n
14. for j from 1 to n
15. if jarak[i,j] > jarak[i,k] + jarak[k,j]
16. jarak[i,j] = jarak[i,k] + jarak[k,j] sebelum[i,j] = sebelum[k,j]
17. return jarak
18. }

```

Gambar 1. Pseudocode Algoritma Floyd-Warshall

Algoritma Dijkstra ditemukan Edsger Dijkstra yang kemudian di berikan nama sesuai dengan penemunya algoritma ini menggunakan prinsip greedy, karena pada setiap jarak dipilih berdasarkan nilai jarak yang terkecil kemudian menghubungkan titik yang sudah terpilih terhadap titik lain yang belum terpilih dengan nilai jarak tak-negatif (Muzzakir, 2020).

```

1. function Dijkstra(Graph, source):
2.   for each vertex v in Graph:
3.     dist[v] := infinity ;
4.     previous[v] := undefined ;
5.   end for
6.   dist[source] := 0 ;
7.   Q := the set of all nodes in Graph
8.   while Q is not empty:
9.     u := vertex in Q with smallest distance in dist[] ;
10.    remove u from Q ;
11.    if dist[u] = infinity:
12.      break ;
13.    end if
14.    for each neighbor v of u:
15.      alt := dist[u] + dist_between(u, v) ;
16.      if alt < dist[v]:
17.        dist[v] := alt ;
18.        previous[v] := u ;
19.        decrease-key v in Q;
20.      end if
21.    end for
22.  end while
23.  return dist;
    
```

Gambar 2. Pesudocode Algoritma Dijkstra

Sistem Informasi Geografis (SIG) adalah sebuah sistem informasi berguna untuk mengolah data serta menyimpan data dalam bentuk informasi geografis (Chang, 2016). SIG ialah alat yang berguna untuk mengumpulkan, melakukan penimbunan, pengambilan kembali data yang ingin ditampilkan serta menampilkan data keruangan berasal dari dunia nyata (Fan, 2013).

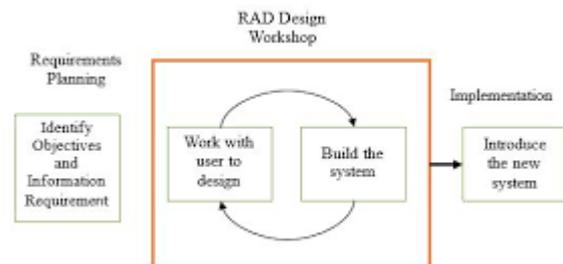
Berdasarkan penelitian sebelumnya serta penjelasan di atas dapat dipahami bahwa algoritma Floyd-Warshall dan Dijkstra dapat digunakan secara maksimal untuk mengetahui jalur tercepat menggunakan SIG dalam menentukan pencarian lokasi BTS, lokasi bus trans Semarang, penentuan rute eke wisata di Lombok dan pencarian rute apotek terdekat. Oleh karena itu, penelitian ini akan menerapkan perbandingan algoritma Floyd-Warshall dan Dijkstra untuk menentukan ruta atau jalur evakuasi terbaik yang terjadi pada korban kecelakaan untuk sampai kerumah sakit. Perbandingan kedua algoritma ini dilakukan dengan tujuan mencari rute terpendek dan mendapatkan pilihan hasil algoritma yang dapat memberikan saran berupa jalur untuk mencapai tujuan (Attamimi, 2017).

METODE

Penelitian ini menerapkan Penelitian ini dilakukan dengan menerapkan *participant observation* yaitu peneliti langsung melakukan pengumpulan data dengan terlibat langsung pada objek penelitian, dalam hal ini peneliti melakukan pengumpulan data langsung dengan mengambil titik koordinat tiap-tiap objek atau titik sebagai data utama.

Tahapan pengembangan sistem dalam penelitian ini adalah menerapkan konsep pengembangan perangkat lunak yaitu menggunakan metode *Software Development Life Cycle* (SDLC). SDLC ialah metode bertahap dalam melakukan analisa dan membangun rancangan sistem dengan menerapkan proses berulang yang spesifik pada kebutuhan pengguna (Kendall, 2006).

Salah satu model dalam *Software Development Life Cycle* (SDLC) adalah *Rapid Application Development* (RAD). Menurut Kendall, RAD merupakan pengembangan perangkat lunak dengan pendekatan berorientasi objek (*object oriented approach*) pada pengembangan sistem. RAD memiliki tujuan dalam mempersingkat waktu perencanaan, waktu perancangan dan penerapan sistem bila dikomparasi berdasarkan metode tradisional. Gambar 1 menunjukkan proses RAD dengan tiga proses utama yaitu perencanaan, perancangan dan penerapan sistem yang mencakup analisis dan pengguna dengan tahapan sebagai berikut (Hamzah, dkk., 2019):



Gambar 3. Siklus RAD

1. Perancangan Kebutuhan Sistem

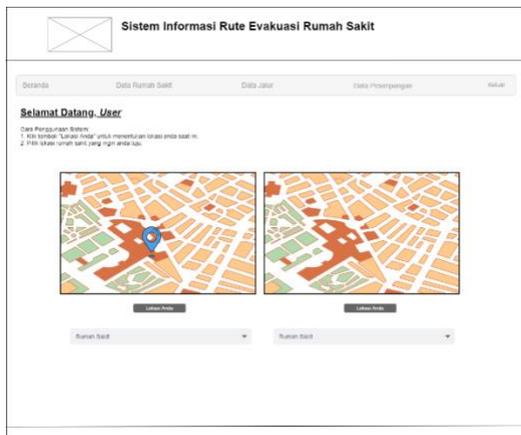
Berdasarkan hasil pengamatan dan studi literatur serta kebutuhan pada implementasi langsung mengenai penerapan algoritma Floyd dan Dijkstra dalam sistem rute evakuasi berbasis web maka didapatkan kebutuhan sistem dalam table 1 berikut:

Tabel 1. Kebutuhan Sistem

No.	Kebutuhan Sistem
1	Sistem menampilkan informasi titik koordinat lokasi pengguna web.
2	Sistem menampilkan lokasi rumah sakit.
3	Sistem dapat menampilkan rute hasil algoritma Floyd dan Dijkstra.
4	Sistem dapat menampilkan rute hasil algoritma Floyd dan Dijkstra.
5	Sistem menyediakan fitur menambahkan data jalur.
6	Sistem menyediakan fitur menambahkan data titik.

2. Proses Perancangan Sistem

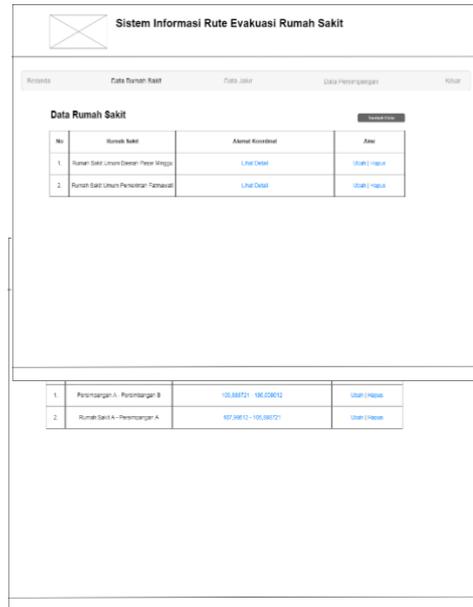
Tahap selanjutnya adalah proses perancangan sistem bersama user, pada proses ini disusun prototype aplikasi yang akan dirancang. Prototype yang akan digunakan adalah model *wireframe*.



Gambar 4. Wireframe Menu Utama

Pada gambar 4 *wireframe* menu utama berisi tata cara penggunaan sistem, menyediakan informasi titik koordinat lokasi pengguna langsung dan pilihan rumah sakit yang ingin dituju.

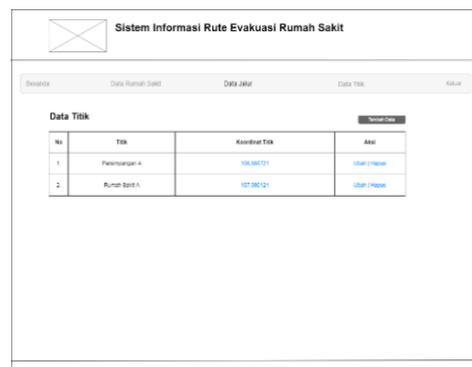
Gambar 5. Wireframe Data Rumah Sakit



Gambar 5 wireframe data rumah sakit menampilkan data rumah sakit yang sudah ada pada sistem dan menyediakan fungsi untuk melakukan tambah data, ubah data dan hapus data rumah sakit.

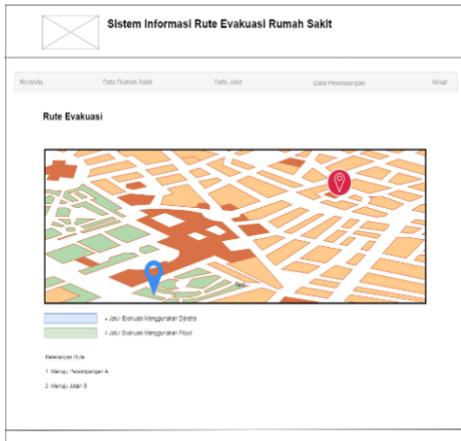
Gambar 6. Wireframe Data Jalur

Gambar 6 *wireframe* data jalur menampilkan data jalur antara titik diambil dari data persimpangan dan menyediakan fungsi untuk melakukan tambah data, ubah data dan hapus data jalur.



Gambar 7. Wireframe Data Persimpangan

Gambar 7 *wireframe* data persimpangan menampilkan data titik yang akan menjadi simpul atau *node* untuk pemrosesan algoritma yang menghubungkan tiap *node* yang ada dan menyediakan fungsi untuk melakukan tambah data, ubah data dan hapus data titik.



Gambar 8. Wireframe Rute Evakuasi

Gambar 8 *wireframe* rute evakuasi menampilkan rute evakuasi hasil pemrosesan menggunakan algoritma Floyd dan Dijkstra dan disediakan informasi jalur yang hasil oleh algoritma Floyd dan Dijkstra.

3. Proses Pengujian

Proses pengujian yang dilakukan yaitu melalui 3 (tiga) tahap pengujian, pertama melakukan *blackbox testing*, selanjutnya adalah *user acceptance testing*, dan melakukan perhitungan akurasi dan *execution time*. *Blackbox testing* adalah pengujian langsung yang dilakukan oleh peneliti untuk menguji apakah sistem yang dibuat sudah sesuai dengan rencana dan sudah dapat berjalan dengan baik. Skenario pengujian fungsional ditunjukkan dalam tabel 2 berikut

Tabel 2. Skenario Pengujian Fungsional

Fungsi	Skenario Proses	Hasil yang diharapkan	Hasil
<i>Redirecting</i>	Saat pengguna mengakses <i>domain</i> .	Diarahkan ke halaman utama web.	

Masuk	Saat pengguna klik tombol masuk.	Diarahkan ke halaman masuk.
Autentikasi	Saat pengguna memasukkan <i>username</i> dan <i>password</i> , klik tombol masuk.	Diarahkan ke <i>controller login</i> dan melakukan autentikasi jika <i>username</i> dan <i>password</i> sesuai makan akan diarahkan ke halaman beranda.
Mendapatkan posisi <i>realtime</i> pada peta	Saat pengguna berada di halaman beranda.	Ditampilkan lokasi langsung pada peta.
Melihat data rumah sakit terdekat	Saat pengguna berada di beranda.	Ditampilkan list rumah sakit terdekat.
Mengarahkan ke rumah sakit terdekat	Saat pengguna klik tombol arahkan	Diarahkan ke halaman rute evakuasi untuk menunjukkan rute terdekat menuju rumah sakit.
Penambahan data rumah sakit	Saat pengguna menambahkan data rumah sakit dan klik tombol simpan.	Diarahkan ke halaman list data rumah sakit dan ditampilkan pesan saat berhasil atau tidak berhasil menambahkan data.
Mengubah data rumah sakit	Saat pengguna mengubah data rumah sakit dan klik tombol simpan	Diarahkan ke halaman list data rumah sakit dan ditampilkan pesan saat berhasil atau tidak berhasil mengubah data.
Menghapus data rumah sakit	Saat pengguna klik tombol hapus.	Diarahkan ke halaman <i>list</i> data rumah sakit dan ditampilkan pesan saat berhasil atau tidak berhasil menghapus data.

Penambahan data jalur	Saat pengguna menambahkan data jalur dan klik tombol simpan.	Diarahkan ke halaman list data jalur dan ditampilkan pesan saat berhasil atau tidak berhasil menambahkan data.
Mengubah data jalur	Saat pengguna mengubah data jalur dan klik tombol simpan	Diarahkan ke halaman list data jalur dan ditampilkan pesan saat berhasil atau tidak berhasil mengubah data.
Penambahan data titik	Saat pengguna menambahkan data titik dan klik tombol simpan.	Diarahkan ke halaman list data titik dan ditampilkan pesan saat berhasil atau tidak berhasil menambahkan data.
Mengubah data titik	Saat pengguna mengubah data titik dan klik tombol simpan	Diarahkan ke halaman list data titik dan ditampilkan pesan saat berhasil atau tidak berhasil mengubah data.
Menghapus data titik	Saat pengguna klik tombol hapus.	Diarahkan ke halaman <i>list</i> data titik dan ditampilkan pesan saat berhasil atau tidak berhasil menghapus data.
Keluar	Pada saat pengguna klik tombol keluar.	Sistem akan menghapus sesi dan kemudian pengguna akan diarahkan kembali ke halaman utama dari domain.

Selanjutnya adalah *user acceptance testing*, yaitu pengujian yang melibatkan calon pengguna untuk menguji apakah sistem yang dibangun sudah sesuai dengan harapan pengguna, jika sistem

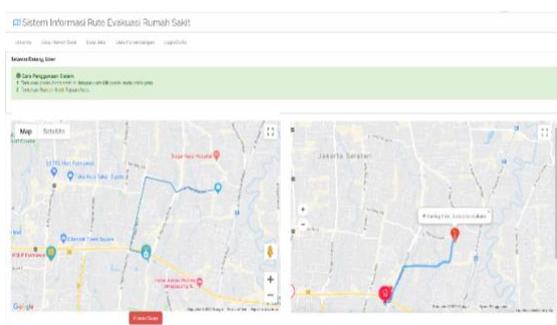
yang dibangun belum berjalan dengan baik, sesuai dengan metode RAD maka proses pengembangan akan dilakukan kembali sampai sistem dapat diterima oleh pengguna dan sesuai dengan rancangan. Proses terakhir yang dilakukan adalah melakukan pengujian dengan menghitung akurasi berdasarkan perbandingan perhitungan algoritma dengan perhitungan pada *maps* dan menghitung *execution time* yang dihasilkan berdasarkan proses *running system* algoritma Floyd dan Dijkstra.

HASIL DAN PEMBAHASAN

Sistem yang dibuat untuk melakukan perbandingan algoritma Floyd-Warshall dan Dijkstra adalah sistem informasi berbasis geografis, dimana terdapat 2 *role user* yang dapat mengakses web tersebut yaitu Admin dan Petugas. Petugas dapat mengunjungi halaman utama web untuk langsung mencari tujuan rumah sakit yang terdekat dan Admin dapat mengatur data rumah sakit, jalur dan data titik persimpangan.

Berdasarkan hasil perancangan yang dilakukan pada bagian sebelumnya, maka hasil implementasi dari perancangan yang dibuat adalah sebagai berikut

1. Halaman Beranda

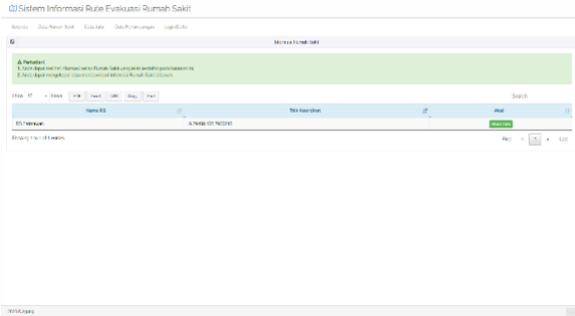


Gambar 9. Tampilan Halaman Beranda

Halaman ini menampilkan data lokasi langsung petugas evakuasi pada peta dan memberikan daftar rumah sakit tujuan, maps yang digunakan berbasis

Google API. Petugas dapat langsung memilih rumah sakit yang dituju.

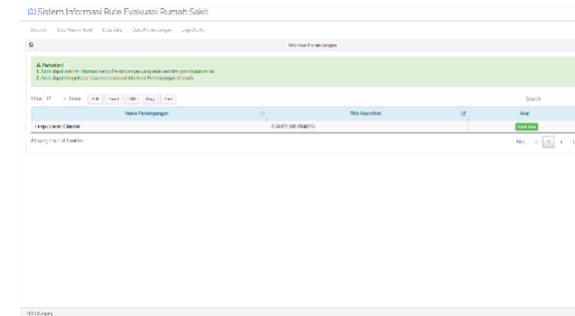
2. Halaman Data Rumah Sakit



Gambar 10. Tampilan Halaman Data Rumah Sakit

Halaman ini menampilkan data rumah sakit yang ada pada sistem dan menyediakan fungsi untuk Admin melakukan tambah data, ubah data dan hapus data rumah sakit.

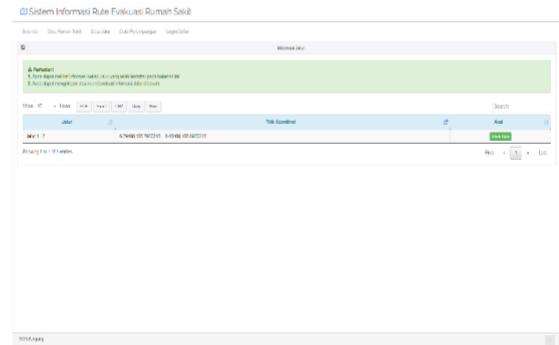
3. Halaman Data Titik Persimpangan



Gambar 11. Tampilan Halaman Data Titik Persimpangan

Halaman ini menampilkan data titik persimpangan yang ada pada sistem dan menyediakan fungsi untuk Admin melakukan tambah data, ubah data dan hapus data rumah Titik Pesimpangan.

4. Halaman Data Jalur



Gambar 12. Tampilan Halaman Data Jalur

Halaman ini menampilkan data jalur yang ada pada sistem dan menyediakan fungsi untuk Admin melakukan tambah data, ubah data dan hapus data Jalur.

Tahap selanjutnya setelah melakukan implementasi adalah melakukan pengujian, berikut hasil pengujian yang dilakukan oleh *role* yang terdapat dalam sistem ditunjukkan dalam tabel 4 dan tabel 5

1. Hasil Pengujian Petugas Evakuasi
Pengujian yang dilakukan adalah melakukan pengujian berdasarkan skenario proses petugas evakuasi pada sistem

Tabel 4. Pengujian Petugas Evakuasi

Role Pengujian	Petugas Evakuasi		
Fungsi	Skenario Proses	Hasil yang diharapkan	Hasil
Redirecting	Saat pengguna mengakses <i>domain</i> .	Diarahkan ke halaman utama web.	Sukses
Masuk	Saat pengguna klik tombol masuk.	Diarahkan ke halaman masuk.	Sukses
Autentikasi	Saat pengguna memasukkan <i>username</i> dan <i>password</i> , klik tombol masuk.	Diarahkan ke <i>controller login</i> dan melakukan autentikasi jika <i>username</i> dan <i>password</i> sesuai maka akan diarahkan ke halaman beranda.	Sukses
Mendapatkan posisi pada peta	Saat pengguna berada di halaman beranda.	Ditampilkan lokasi langsung pada peta.	Sukses

Melihat data rumah sakit terdekat	Saat pengguna berada di beranda.	Ditampilkan list rumah sakit terdekat.	Sukses			untuk menunjukkan rute terdekat menuju rumah sakit.	
Mengarahkan ke rumah sakit terdekat	Saat pengguna klik tombol arahkan	Diarahkan ke halaman rute evakuasi untuk menunjukkan rute terdekat menuju rumah sakit.	Sukses	Penambahan data rumah sakit	Saat pengguna menambahkan data rumah sakit dan klik tombol simpan.	Diarahkan ke halaman list data rumah sakit dan ditampilkan pesan saat berhasil atau tidak berhasil menambahkan data.	Sukses
Keluar	Pada saat pengguna klik tombol keluar.	Sistem akan menghapus sesi dan kemudian pengguna akan diarahkan kembali ke halaman utama dari domain.	Sukses	Mengubah data rumah sakit	Saat pengguna mengubah data rumah sakit dan klik tombol simpan	Diarahkan ke halaman list data rumah sakit dan ditampilkan pesan saat berhasil atau tidak berhasil mengubah data.	Sukses

2. Hasil Pengujian Admin
 Pengujian yang dilakukan adalah melakukan pengujian berdasarkan skenario proses pengujian admin.

Tabel 5. Pengujian Admin

Role Pengujian		Admin					
Fungsi	Skenario Proses	Hasil yang diharapkan	Hasil				
Redirecting	Saat pengguna mengakses domain.	Diarahkan ke halaman utama web.	Sukses	Penambahan data jalur	Saat pengguna menambahkan data jalur dan klik tombol simpan.	Diarahkan ke halaman list data jalur dan ditampilkan pesan saat berhasil atau tidak berhasil menambahkan data.	Sukses
Masuk	Saat pengguna klik tombol masuk.	Diarahkan ke halaman masuk.	Sukses	Mengubah data jalur	Saat pengguna mengubah data jalur dan klik tombol simpan	Diarahkan ke halaman list data jalur dan ditampilkan pesan saat berhasil atau tidak berhasil mengubah data.	Sukses
Autentikasi	Saat pengguna memasukkan username dan password, klik tombol masuk.	Diarahkan ke controller login dan melakukan autentikasi jika username dan password sesuai makan akan diarahkan ke halaman beranda.	Sukses	Penambahan data titik	Saat pengguna menambahkan data titik dan klik tombol simpan.	Diarahkan ke halaman list data titik dan ditampilkan pesan saat berhasil atau tidak berhasil menambahkan data.	Sukses
Mendapatkan posisi realtime pada peta	Saat pengguna berada di halaman beranda.	Ditampilkan lokasi langsung pada peta.	Sukses				
Melihat data rumah sakit terdekat	Saat pengguna berada di beranda.	Ditampilkan list rumah sakit terdekat.	Sukses				
Mengarahkan ke rumah sakit terdekat	Saat pengguna klik tombol arahkan	Diarahkan ke halaman rute evakuasi	Sukses				

Mengubah data titik	Saat pengguna mengubah data titik dan klik tombol simpan	Diarahkan ke halaman list data titik dan ditampilkan pesan saat berhasil atau tidak berhasil mengubah data.	Sukses
Menghapus data titik	Saat pengguna klik tombol hapus.	Diarahkan ke halaman <i>list</i> data titik dan ditampilkan pesan saat berhasil atau tidak berhasil menghapus data.	Sukses
Keluar	Pada saat pengguna klik tombol keluar.	Sistem akan menghapus sesi dan kemudian pengguna akan diarahkan kembali ke halaman utama dari domain.	Sukses

Berdasarkan pengujian didapatkan bahwa hasil pengujian *User Acceptance Testing* pada web menunjukkan tidak adanya kesalahan atau *error* pada setiap fungsi website sehingga dapat diterima oleh pengguna pada setiap peran. Oleh karena itu berdasarkan *Black Box Testing* dan *User Acceptance Testing* dapat dipastikan bahwa web telah berjalan sesuai dengan kebutuhan dan layak digunakan oleh pengguna.

Tahap akhir adalah melakukan perhitungan *execution time* terhadap sistem yang berjalan dan melakukan perbandingan jarak tempuh, pada perhitungan *execution time* algoritma Dijkstra memiliki waktu eksekusi yang lebih cepat dibandingkan algoritma Floyd-Warshall ditunjukkan dalam tabel 6 berikut

<i>Algoritma</i>	<i>Execution Time</i>
Floyd-Warshall	0.016000032424927 ms
Dijkstra	0.014000177383423 ms

Proses selanjutnya melakukan perhitungan jarak berdasarkan node yang dilewati dengan melakukan pengujian terhadap beberapa rumah sakit terdekat yang ditunjukkan pada tabel 7 berikut

<i>Rumah Sakit</i>	<i>Dijkstra</i>	<i>Floyd-Warshall</i>
RS A	7143.12 Km	6267.36 Km
RS B	4812.22 Km	3302.14 Km

Berdasarkan perhitungan waktu proses pencarian rute dan jarak tempuh web telah berhasil sesuai dengan kebutuhan dan layak digunakan oleh petugas.

SIMPULAN

Berdasarkan hasil penelitian yang dilakukan, maka dapat disimpulkan, diantaranya :

1. Telah dibangun sebuah sistem web untuk melakukan perbandingan algoritma Floyd-Warshall dan Dijkstra dengan baik dan benar sesuai dengan perancangan dan hasil pengujian.
2. Sistem yang dirancang dapat dimanfaatkan petugas evakuasi untuk mencari rute evakuasi terdekat menuju rumah sakit dengan melak.
3. Sistem berhasil dioperasikan dengan menampilkan hasil perbandingan rute terdekat antara algoritma Floyd-Warshall dan Dijkstra untuk menuju rumah sakit dengan hasil algoritma Dijkstra lebih sedikit menggunakan memori dan waktu proses pencarian rute yang lebih cepat dari pada algoritma Floyd Warshall namun algoritma Floyd Warshall menghasilkan rute lebih pendek.

DAFTAR PUSTAKA

Ardana, D., & Saputra, R. (2016,

- October). Penerapan Algoritma Dijkstra pada Aplikasi Pencarian Rute Bus Trans Semarang. In *Seminar Nasional Ilmu Komputer (SNIK 2016)* (pp. 299-306).
- Attamimi, I., Yahya, W., & Hanafi, M. H. (2017). Analisis Perbandingan Algoritma Floyd-Warshall dan Dijkstra untuk Menentukan Jalur Terpendek Pada Jaringan Openflow. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer e-ISSN, 2548, 964X*.
- Chang, K. T. (2016). Geographic information system. *International Encyclopedia of Geography: People, the Earth, Environment and Technology*, 1-10.
- Hamzah, M.L., Purwati, A.A., Rusilawati, E., & Hamzah. (2019). Rapid Application Development In Design Of Library Information System In Higher Education. *International Journal Of Scientific & Technology Research*, 8(11), 153-156.
- Mahalakshmi, M., & Sundararajan, M. (2013). Traditional SDLC Vs Scrum Methodology–A Comparative Study. *International Journal of Emerging Technology and Advanced Engineering*, 3(6), 192-196.
- Masyhudi, L., & Khalik, W. (2018). Penentuan Rute Wisata Minimum Di Pulau Lombok Nusa Tenggara Barat Dengan Pendekatan Algoritma Dijkstra. *Media bina ilmiah*, 12(12), 689.
- Muzakir, A. (2020). Algoritma Floyd Warshall Dan Collaborative Filtering Untuk Penentuan Rekomendasi Dan Rute Terpendek Pencarian Apotek: Studi Eksperimen. *InfoTekJar: Jurnal Nasional Informatika dan Teknologi Jaringan*, 5(1), 9-13.
- Nawagusti, V. A., Nurdin, A., & Aryanti, A. (2018, February). Penentuan Rute Terpendek Pada Optimalisasi Jalur Pendistribusian Barang Di Pt. X Dengan Menerapkan Algoritma Floyd-Warshall. In *Seminar Nasional Inovasi dan Aplikasi Teknologi Di Industri 2018* (Vol. 4, No. 2, pp. 57-64). Fakultas Teknologi Industri-Institut Teknologi Nasional Malang.