

FINAL PROJECT
MATA KULIAH MATEMATIKA DISKRIT



OLEH :
KELOMPOK 3

**"PENCARIAN LOKASI RUMAH SAKIT BERDASARKAN RUTE
TERPENDEK DARI PINTU EXIT TOL TAMBAK SUMUR
MENGGUNAKAN ALGORITMA DIJKSTRA"**

| | |
|-------------|----------------------------|
| 23083010046 | STEFFANY MARCELLIA WITANTO |
| 23083010051 | FRIZA NUR FATMALA |
| 23083010061 | DIFTA ALZENA SAKHI |
| 23083010064 | AZIZAH ZALFA ASSYADIDA |
| 23083010068 | NI LUH AYU NARISWARI DEWI |

**PROGRAM STUDI SAINS DATA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS PEMBANGUNAN NASIONAL "VETERAN" JATIM
2024**

Tabel Pembagian Tugas

| NPM | Nama Mahasiswa | Kontribusi |
|-------------|----------------------------|--|
| 23083010046 | Steffany Marcellia Witanto | <ul style="list-style-type: none">- Pengumpulan data rumah sakit dan jarak dari aplikasi Google Maps.- Penyusunan bab latar belakang, penjelasan dataset, dan dokumentasi hasil pengujian aplikasi. |
| 23083010051 | Friza Nur Fatmala | <ul style="list-style-type: none">- Pengembangan antarmuka GUI menggunakan Tkinter.- Penyusunan bab hasil dan pembahasan. |
| 23083010061 | Difta Alzena Sakhi | <ul style="list-style-type: none">- Pengembangan visualisasi peta menggunakan pustaka <i>Folium</i>.- Penyusunan bab penjelasan dataset dan kesimpulan. |

| | | |
|-------------|---------------------------|---|
| 23083010064 | Azizah Zalfa Assyadida | <ul style="list-style-type: none"> - Pengujian kesesuaian antarmuka dan pengolahan dataset. - Penyusunan bab metodologi penelitian. |
| 23083010068 | Ni Luh Ayu Nariswari Dewi | <ul style="list-style-type: none"> - Pengembangan antarmuka GUI menggunakan Tkinter. ak ubah - Penyusunan bab hasil dan pembahasan. |

"PENCARIAN LOKASI RUMAH SAKIT BERDASARKAN RUTE TERPENDEK DARI PINTU EXIT TOL TAMBAK SUMUR MENGGUNAKAN ALGORITMA DIJKSTRA"

"Algoritma Pencarian Rute Terpendek dengan Metode Dijkstra"

LATAR BELAKANG

Jalan tol merupakan salah satu inovasi prasarana di bidang transportasi yang mampu menghubungkan antar wilayah dengan cepat dan efisien. Dikenal sebagai jalan bebas hambatan, jalan tol memberikan fasilitas bagi pengemudi untuk melakukan perjalanan dengan waktu yang lebih singkat dibandingkan melakukan perjalanan di jalan raya konvensional atau jalan umum. Selain mempercepat waktu perjalanan, jalan tol juga dapat mengurangi kemacetan di beberapa titik pada jalan raya konvensional, serta memberikan rasa nyaman bagi pengemudi karena terhindar dari rasa stress akibat kemacetan yang terjadi. Jalan tol juga menyediakan fasilitas seperti *rest area* yang menyediakan tempat makan, pom bensin, ataupun *mini market*. Hal ini memberikan kemudahan aksesibilitas bagi pengemudi jika membutuhkan sesuatu.

Dari beberapa fasilitas yang tersedia di sepanjang jalan tol, fasilitas kesehatan seperti rumah sakit akan menjadi salah satu fasilitas yang sangat penting dan dibutuhkan. Hal ini bertujuan untuk memberikan pertolongan pertama ketika terjadi masalah kesehatan yang darurat, seperti asma, serangan jantung, bersalin, bahkan kecelakaan yang bisa terjadi di jalan tol. Jika di sepanjang jalan tol tidak tersedia rumah sakit, maka rumah sakit dengan rute terpendek dari pintu exit tol akan menjadi tujuan utama yang dikunjungi. Hal ini dilakukan agar masalah kesehatan darurat yang terjadi segera mendapat penanganan dan tidak menimbulkan akibat yang fatal.

Pemilihan rute terpendek menuju rumah sakit dari pintu exit tol untuk kondisi kesehatan darurat dapat dilakukan dengan sebuah metode matematika, yaitu algoritma Dijkstra. Algoritma ini pertama kali dikemukakan oleh Edsger

W.Dijkstra pada tahun 1959 dan telah digunakan secara luas untuk menentukan rute terpendek berdasarkan kriteria tertentu yang digunakan sebagai batasan. Selain itu, algoritma Dijkstra juga dapat didefinisikan sebagai sebuah algoritma rakus (*greedy algorithm*) yang digunakan untuk memecahkan permasalahan jarak terpendek (*shortest path problem*) untuk sebuah graf berarah (*directed graph*) dengan bobot-bobot sisi (*edge weights*) yang bernilai tidak negatif. Pada kasus pencarian rute terpendek dari Pintu Exit Tol Tambak Sumur menuju ke sebuah rumah sakit, simpul (*nodes*) menggambarkan Pintu Exit Tol Tambak Sumur sebagai titik awal dan beberapa rumah sakit sebagai tujuan. Sedangkan, bobot-bobot sisi (*edge weights*) menggambarkan jarak antara Pintu Exit Tol Tambak Sumur dengan beberapa rumah sakit. Sehingga, algoritma Dijkstra pada kasus ini digunakan untuk menemukan rute terpendek antara Pintu Exit Tol Tambak Sumur dengan beberapa rumah sakit.

Penggunaan algoritma Dijkstra bagi orang awam akan cukup sulit untuk dipahami, sehingga perlu adanya pengembangan berupa aplikasi yang mempermudah pengguna. Aplikasi GUI (*Graphical User Interface*) menjadi sebuah pilihan aplikasi yang dapat mempermudah pengguna untuk mencari rumah sakit dengan rute terpendek ketika terjadi masalah kesehatan darurat. GUI menampilkan pilihan titik awal dan titik akhir pengemudi untuk mencari rute terpendek menuju rumah sakit di sekitar, serta menampilkan jarak antara titik awal dan titik akhir. Selain itu, tersedia tampilan *maps* dari titik awal menuju titik akhir yang mempermudah pengemudi untuk mengetahui rute jalan mana saja yang akan dilalui.

Pemilihan rute terpendek untuk masalah kesehatan darurat di jalan tol menjadi sangat penting karena harus memanfaat waktu sesingkat mungkin untuk segera mendapatkan pertolongan. Pengembangan aplikasi GUI untuk algoritma Dijkstra dapat membantu pengemudi mencari rute terpendek menuju rumah sakit terdekat dengan mempertimbangkan jarak antar titik. Melalui pengembangan aplikasi ini, diharapkan dapat mengurangi risiko fatal akibat keterlambatan pertolongan medis di jalan tol.

"PENCARIAN LOKASI RUMAH SAKIT BERDASARKAN RUTE TERPENDEK DARI PINTU EXIT TOL TAMBAK SUMUR MENGGUNAKAN ALGORITMA DIJKSTRA"

"Algoritma Pencarian Rute Terpendek dengan Metode Dijkstra"

PENJELASAN DATASET

Dataset yang digunakan dalam penelitian ini merupakan daftar rumah sakit yang berada di sekitar lokasi awal, yaitu Pintu Exit Tol Tambak Sumur. Dataset ini dirancang untuk mendukung proses pencarian rute terpendek menggunakan algoritma Dijkstra. Penggunaan algoritma Dijkstra akan memberikan rekomendasi rute terpendek menuju rumah sakit terdekat, sehingga mempermudah pengguna dalam mendapatkan pertolongan saat berada dalam masalah kesehatan darurat.

A. Sumber Data

Data dalam dataset ini dikumpulkan secara manual melalui platform Google Maps. Metode pengumpulan data melibatkan pencatatan jarak langsung berdasarkan rute kendaraan dari Pintu Exit Tol Tambak Sumur ke rumah sakit tujuan. Proses ini memberikan estimasi jarak yang realistik sesuai dengan kondisi jalan aktual, termasuk belokan, jalur khusus, dan lalu lintas. Dengan menggunakan data ini, aplikasi dapat memberikan hasil yang lebih akurat dibandingkan estimasi jarak berbasis koordinat geografis (*latitude* dan *longitude*).

B. Tujuan Dataset

Dataset yang digunakan dalam penelitian ini memiliki beberapa tujuan utama yang dirancang untuk mendukung pengembangan aplikasi berbasis algoritma Dijkstra. Tujuan tersebut antara lain:

1. Memberikan Rekomendasi Rute Terpendek

Dataset ini memungkinkan aplikasi untuk menghitung dan

memberikan rekomendasi rute terpendek dari Pintu Exit Tol Tambak Sumur ke rumah sakit terdekat. Dengan menggunakan data jarak langsung yang diambil dari Google Maps, aplikasi dapat memberikan hasil yang cepat dan akurat.

2. Mendukung Implementasi Algoritma Dijkstra

Dataset menyediakan bobot graf berupa jarak antar lokasi untuk mendukung perhitungan rute terpendek menggunakan algoritma Dijkstra. Struktur dataset yang sederhana mempermudah algoritma dalam menentukan rute optimal.

3. Mempermudah Navigasi Pengguna

Dataset ini dirancang untuk membantu pengguna memahami lokasi rumah sakit secara geografis melalui integrasi dengan visualisasi peta. Dengan peta interaktif, pengguna dapat melihat rute yang direkomendasikan secara visual.

C. Kriteria Rumah Sakit

Rumah sakit yang termasuk dalam dataset memiliki beberapa kriteria sebagai berikut:

1. Jarak Maksimal 10 Kilometer

Rumah sakit yang termasuk dalam dataset harus berada dalam radius maksimal, yaitu 10 kilometer dari Pintu Exit Tol Tambak Sumur untuk memastikan relevansi lokasi dalam keadaan darurat.

2. Akses Jalan yang Memadai

Lokasi rumah sakit harus dapat diakses dengan kendaraan tanpa hambatan besar, seperti jalan yang terlalu sempit atau kondisi jalan yang rusak.

3. Kelayakan Layanan Medis

Rumah sakit harus memiliki fasilitas medis yang memadai untuk menangani kondisi darurat, seperti unit gawat darurat atau layanan trauma.

D. Struktur Dataset

Dataset ini memiliki beberapa kolom utama yang menjelaskan elemen-elemen penting dalam pencarian rute terpendek. Berikut adalah penjelasan rinci mengenai kolom-kolom dalam dataset:

| Nama Kolom | Deskripsi | Tipe Data |
|------------------|--|----------------|
| No | Nomor baris dari dataset | <i>Integer</i> |
| Nama Rumah Sakit | Nama rumah sakit yang memenuhi kriteria dalam dataset | <i>String</i> |
| Jarak | Jarak antara Pintu Exit Tol Tambak Sumur ke rumah sakit tujuan | <i>Float</i> |
| Titik Awal | Titik awal pencarian rute, yaitu Pintu Exit Tol Tambak Sumur | <i>String</i> |

E. Pengolahan Data

1. Nilai bobot (*weight*) pada graf untuk setiap sisi (*edge*) diambil berdasarkan jarak yang diperoleh melalui aplikasi Google Maps.
2. Data diorganisasi dalam bentuk dictionary Python untuk memudahkan pengolahan dalam aplikasi berbasis algoritma Dijkstra.
3. Dataset ini tidak menggunakan koordinat geografis (*latitude* dan *longitude*), melainkan jarak langsung dari titik awal ke titik tujuan.

F. Manfaat Dataset

Adapun manfaat dari dataset yang digunakan dalam implementasi algoritma Dijkstra sebagai berikut:

1. Memastikan akurasi perhitungan rute dengan jarak realistik berdasarkan kondisi jalan.
2. Mendukung visualisasi peta sehingga pengguna dapat melihat rute dan lokasi rumah sakit dengan lebih mudah.
3. Meningkatkan kecepatan proses pencarian rute terpendek dalam sistem aplikasi.

"PENCARIAN LOKASI RUMAH SAKIT BERDASARKAN RUTE TERPENDEK DARI PINTU EXIT TOL TAMBAK SUMUR MENGGUNAKAN ALGORITMA DIJKSTRA"

"Algoritma Pencarian Rute Terpendek dengan Metode Dijkstra"

METODOLOGI PENELITIAN

A. Pengertian Algoritma Dijkstra

Algoritma Dijkstra merupakan metode iteratif yang digunakan untuk menentukan jalur terpendek dari sebuah simpul sumber (*source node*) ke simpul tujuan (*destination node*) dalam graf berbobot positif. Dalam konteks *graph*, simpul (*node*) mewakili lokasi, sedangkan sisi (*edge*) mewakili koneksi antara dua lokasi dengan bobot (*weight*) yang menunjukkan nilai tertentu seperti jarak, waktu, atau biaya. Algoritma Dijkstra ditemukan oleh Edsger W. Dijkstra pada tahun 1956 dan dipublikasikan pada tahun 1959. Algoritma Dijkstra dikenal juga sebagai algoritma *greedy* di mana cara kerjanya adalah dengan melakukan pemilihan pada simpul yang memiliki jarak terpendek kemudian memperbarui jarak simpul yang lain. Deskripsi matematis untuk grafik pada algoritma ini dapat diwakili dengan $G = \{V, E\}$, yang berarti sebuah grafik (G) didefinisikan oleh satu set simpul (*Vertex* = V) dan koleksi *Edge* (E).

Rumus Matematis:

$$d(v) = \min(d(v), d(u) + w(u,v))$$

Di mana:

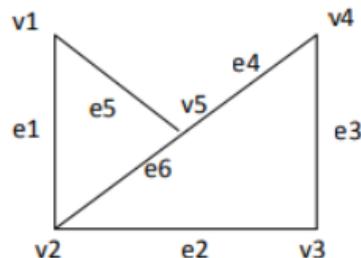
- $d(u)$: Jarak terpendek dari simpul sumber ke simpul u
- $d(v)$: Jarak terpendek dari simpul sumber ke simpul v
- $w(u,v)$: Bobot dari sisi antara simpul u dan v

B. Pengertian Graf

Graf merupakan himpunan dari benda-benda yang disebut simpul (*node*) yang terhubung oleh sisi (*edge*) atau busur (*arc*). Biasanya graf digambarkan sebagai kumpulan titik yang melambangkan simpul dan dihubungkan oleh garis yang melambangkan sisi atau garis berpanah yang melambangkan busur. Suatu graf G didefinisikan sebagai pasangan himpunan (V,E) , dimana V merupakan himpunan yang berisikan simpul pada graf tersebut $\{ , \}$ dan E adalah himpunan sisi yang menghubungkan simpul-simpul $\{ , \}$ atau dapat ditulis dengan notasi $G = (V,E)$. Graf dapat dibedakan menjadi 2 jenis berdasarkan arah pada sisinya.

1. Graf Tak Berarah

Graf tak berarah (*Undirected Graph*) merupakan graf yang setiap sisinya tidak memiliki arah tertentu. Hubungan antara 2 simpul graf ini bersifat 2 arah.

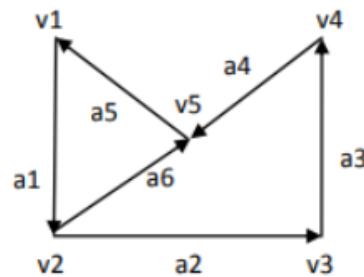


Gambar 3.1 Graf Tak Berarah

Gambar 3.1 menunjukkan graf tidak berarah dengan himpunan simpul, $V(G) = \{ , \}$ dan himpunan sisi $E(G) \{e1, e2, e3, e4, e5, e6\}$ yaitu pasangan terurut dari $\{(v1,v2), (v2,v3), (v3,v4), (v4,v5), (v5,v1), (v5,v2)\}$.

2. Graf Berarah

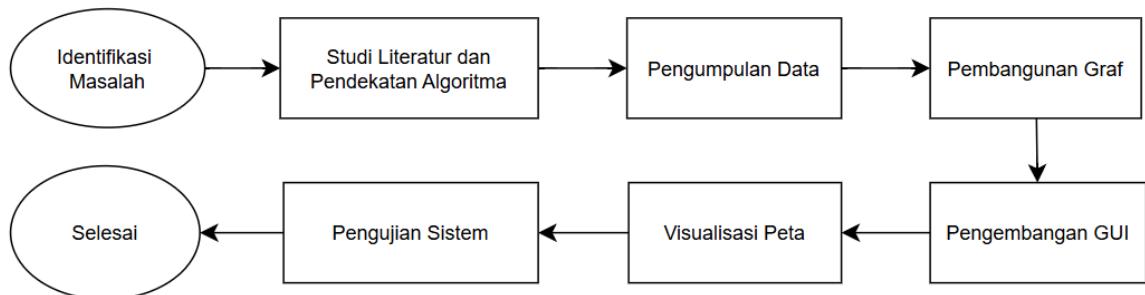
Graf Berarah (*Directed Graph*) merupakan graf yang setiap sisinya memiliki arah tertentu dengan hanya 1 hubungan arah yang terjadi di antara 2 simpul.



Gambar 3.2 Graf Berarah

Gambar 3.2 menunjukkan graf berarah dengan himpunan simpul $V(G) = \{ , \}$ dan himpunan busur $A(G) = \{a_1, a_2, a_3, a_4, a_5, a_6\}$ yaitu pasangan terurut dari $\{(v_1, v_2), (v_2, v_3), (v_3, v_4), (v_4, v_5), (v_5, v_1), (v_2, v_5)\}$.

C. Alur Penelitian



Gambar 3.3 Diagram Alir Penelitian

Penelitian untuk mengembangkan aplikasi yang mampu mencari rumah sakit terdekat dari Pintu Exit Tol Tambak Sumur dilakukan melalui berbagai tahapan yang sistematis. Langkah-langkah yang dilakukan tidak hanya membahas penggunaan algoritma graf untuk menemukan

jarak terpendek, tetapi juga bagaimana data dapat diolah secara efektif, dimaksudkan untuk meningkatkan efisiensi dan akurasi sistem, dan bagaimana hasilnya dapat disajikan dalam visualisasi yang mudah dipahami oleh pengguna. Berikut adalah tahapan metodologi dalam penelitian ini:

1. Identifikasi Masalah

Tahap pertama penelitian ini adalah menemukan masalah yang dialami pengguna, terutama dalam situasi darurat. Dalam situasi seperti ini, pengguna membutuhkan solusi yang cepat dan tepat untuk menemukan rumah sakit terdekat yang mudah diakses dari Pintu *Exit* Tol Tambak Sumur. Oleh karena itu, fokus penelitian ini adalah membuat aplikasi yang dapat secara efektif menggunakan konsep graf untuk memetakan lokasi rumah sakit dan menghitung jarak antar titik.

2. Studi Literatur dan Pendekatan Algoritma

Langkah selanjutnya adalah mempelajari konsep dasar dari graf untuk memastikan bahwa metode yang digunakan dalam penelitian ini memenuhi persyaratan. Pendekatan graf menggambarkan setiap pintu keluar tol dan lokasi rumah sakit sebagai simpul (*node*). Sementara itu, jarak antar lokasi digambarkan sebagai sisi (*edge*) yang memiliki bobot tertentu. Selain itu, penelitian melihat algoritma Dijkstra, yang telah terbukti efektif dalam menemukan rute terpendek untuk graf berbobot. Selain itu, pustaka *Python* seperti *NetworkX* dipelajari untuk membantu operasi graf seperti membangun grafik dan menjalankan algoritma Dijkstra. Bagian penting dari penelitian ini adalah visualisasi peta. Untuk itu, *Folium Library* dapat digunakan karena pustaka tersebut mampu memvisualisasikan peta interaktif untuk menunjukkan lokasi dan hubungan antar titik.

3. Pengumpulan Data

Penelitian ini menggunakan data berupa daftar rumah sakit di sekitar Pintu Exit Tol Tambak Sumur, serta jarak antar lokasi. Data ini diperoleh melalui sumber terpercaya seperti peta digital dan survei lapangan. Setelah dikumpulkan, data diproses dan disusun dalam format *dictionary Python* yang akan memudahkan peneliti dalam mengakses informasi selama proses pembangunan graf dan implementasi algoritma.

4. Pembangunan Graf

Pada tahap ini, graf tidak berarah akan dibuat menggunakan pustaka *NetworkX*. Sebagai contoh, setiap simpul akan menunjukkan lokasi tertentu, seperti pintu keluar tol atau rumah sakit, dan jarak antar lokasi diberi bobot di sisi-sisi graf. Algoritma Dijkstra menggunakan grafik ini untuk menemukan jalan terpendek antara dua simpul yang dipilih pengguna. Metode ini juga memungkinkan fleksibilitas untuk menambahkan simpul atau bagian baru ke dalam sistem jika ada rumah sakit tambahan yang ingin dimasukkan.

5. Pengembangan Antarmuka Pengguna (GUI)

Pustaka *Tkinter* digunakan untuk mengembangkan antarmuka aplikasi, yang menawarkan komponen GUI yang sederhana namun berfungsi dengan baik. Melalui menu *drop-down* interaktif aplikasi ini, pengguna dapat memilih titik awal dan titik yang akan dituju. Selain itu, terdapat fitur penambahan data baru, yang memungkinkan pengguna memasukkan rumah sakit tambahan dan jarak dari rumah sakit tertentu. Selanjutnya, aplikasi ini akan menampilkan hasil pencarian rute terpendek secara langsung di layar dalam bentuk teks yang mudah dipahami oleh pengguna.

6. Visualisasi Peta

Dalam penelitian ini, visualisasi geografis menjadi bagian penting karena membantu pengguna memahami lokasi rumah sakit dalam peta yang sebenarnya. Pustaka *Folium* dapat digunakan untuk membuat peta interaktif yang dibutuhkan. Pada peta, *marker* akan menunjukkan lokasi rumah sakit dan pintu keluar tol, serta jalur rute terpendek yang ditentukan oleh algoritma Dijkstra. Dengan mengekspor peta dalam format HTML, pengguna dapat membuka dan mengaksesnya langsung melalui *browser web* mereka.

7. Pengujian Sistem

Tahap akhir penelitian adalah pengujian aplikasi untuk memastikan bahwa semua fitur berfungsi dengan baik. Pengujian dapat dilakukan dengan melibatkan beberapa skenario, seperti memilih kombinasi rumah sakit awal dan tujuan yang berbeda, menambahkan data baru, dan memverifikasi jarak terpendek yang dihitung oleh algoritma. Selain menguji kejelasan antarmuka dan peta melalui umpan balik pengguna, keakuratan hasil juga diuji dengan melakukan perbandingan antara data awal dan hasil yang didapatkan.

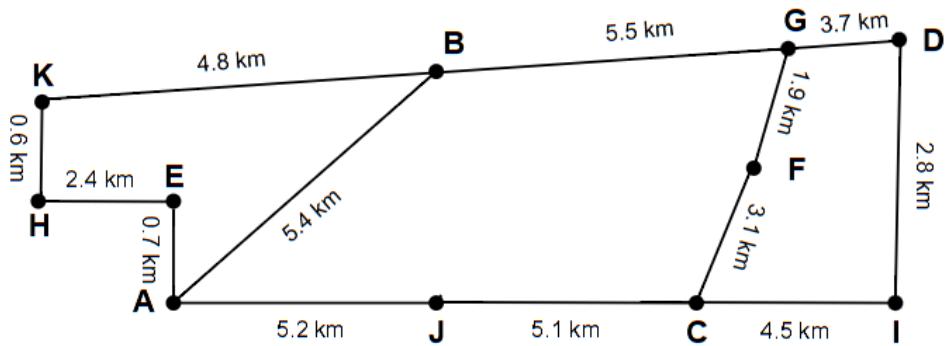
Tujuan penelitian ini adalah untuk membuat aplikasi pencarian rumah sakit terdekat yang menggunakan algoritma Djikstra. Algoritma ini akan memungkinkan pengguna menemukan rumah sakit terdekat dari Pintu Exit Tol Tambak Sumur dengan mudah. Setiap langkah dalam proses pengembangan, mulai dari membuat grafik, menggunakan algoritma Djikstra, dan membuat antarmuka grafis, telah dirancang untuk memastikan hasil pencarian yang akurat dan mudah digunakan.

"PENCARIAN LOKASI RUMAH SAKIT BERDASARKAN RUTE TERPENDEK DARI PINTU EXIT TOL TAMBAK SUMUR MENGGUNAKAN ALGORITMA DIJKSTRA"

"Algoritma Pencarian Rute Terpendek dengan Metode Dijkstra"

HASIL DAN PEMBAHASAN

Kondisi darurat yang terjadi di jalan tol seringkali menjadi masalah serius dikarenakan jalur ini umumnya didesain untuk kecepatan tinggi dan bebas hambatan, sehingga cenderung jauh untuk mencari tempat untuk melakukan penanganan medis. Dalam situasi darurat seperti kecelakaan dan darurat bersalin, kecepatan respons medis sangat penting untuk dilakukan. Berdasarkan problematika ini, pencarian solusi dalam mencari rute tercepat dari titik awal pada Pintu Exit Tol Tambak Sumur dirancang untuk memudahkan para pengguna jalan tol. Model simulasi dan antarmuka telah berhasil diimplementasikan dan diuji menggunakan algoritma Dijkstra untuk mencari rute terpendek antara Pintu Exit Tol Tambak Sumur dan beberapa lokasi rumah sakit di Surabaya. Algoritma tersebut bekerja dengan mempertimbangkan berbagai jalur alternatif dan memastikan bahwa pengguna tidak perlu melewati rute yang lebih lama atau terhambat. Dalam melakukan pengujian ini, pengguna harus menentukan titik awal (Gerbang Tol Tambak Sumur) dan titik tujuan (rumah sakit yang diinginkan) sebelum menjalankan antarmuka. Sistem kemudian akan menunjukkan rute terpendek serta rekomendasi rumah sakit yang akan dilalui. Berikut merupakan graf yang dibentuk dari peta lokasi Rumah sakit yang memiliki letak terdekat dari Pintu Exit Tol Tambak Sumur :



Gambar 4.1 Graf yang Terbentuk Dari Rute Rumah Sakit

Proses ini memanfaatkan data koordinat geografis dari aplikasi Google Maps untuk menentukan lokasi rumah sakit yang mengutamakan efisiensi waktu dan jarak. Algoritma Dijkstra berperan dalam mencari rute terpendek dengan rumah sakit tujuan digambarkan sebagai titik (*node*) dan jarak antara rumah sakit tujuan dan pintu exit tol digambarkan sebagai sisi yang memiliki bobot. Data awal mencakup sepuluh rumah sakit, namun pengguna juga dapat menambahkan data rumah sakit baru sesuai kebutuhan.

Tabel 4.1 Kode Rumah Sakit Terdekat dari Exit Tol

| Kode | Keterangan |
|------|----------------------------------|
| A | Pintu Exit Tol Tambak Sumur |
| B | Royal Hospital Surabaya |
| C | RS Gotong Royong |
| D | RS Manyar Medical Surabaya |
| E | RS Mitra Keluarga Pondok Tjandra |
| F | RS Premiere Surabaya |

| | |
|---|-------------------------------|
| G | RS Ubaya |
| H | RSU Bunda Waru |
| I | RSUD Haji Provinsi Jawa Timur |
| J | RSUD Surabaya Timur |
| K | RSU Prima Husada |

Berikut ini merupakan *code script* dengan implementasi algoritma Dijkstra dalam mencari rute terpendek lokasi rumah sakit terdekat dari Pintu Exit Tol Tambak Sumur :

```
import heapq
import tkinter as tk
from tkinter import ttk, messagebox

class Node:
    def __init__(self, code):
        self.code = code
        self.neighbors = {}

    def add_neighbor(self, neighbor, weight):
        self.neighbors[neighbor] = weight

def dijkstra(start_node, end_node):
    priority_queue = [(0, start_node, [start_node.code])]
    visited = set()
    shortest_path = None

    while priority_queue:
        current_distance, current_node, path =

```



```

    "Royal Hospital Surabaya (B)": Node("Royal Hospital
Surabaya (B)") ,
    "RS Gotong Royong (C)": Node("RS Gotong Royong (C)") ,
    "RS Manyar Medical Surabaya (D)": Node("RS Manyar Medical
Surabaya (D)") ,
    "RS Mitra Keluarga Pondok Tjandra (E)": Node("RS Mitra
Keluarga Pondok Tjandra (E)") ,
    "RS Premiere Surabaya (F)": Node("RS Premiere Surabaya
(F)") ,
    "RS Ubaya (G)": Node("RS Ubaya (G)") ,
    "RSU Bunda Waru (H)": Node("RSU Bunda Waru (H)") ,
    "RSUD Haji Provinsi Jawa Timur (I)": Node("RSUD Haji
Provinsi Jawa Timur (I)") ,
    "RSUD Surabaya Timur (J)": Node("RSUD Surabaya Timur
(J)") ,
    "RSU Prima Husada (K)": Node("RSU Prima Husada (K)") ,
}

# Menambahkan tetangga (jarak antar titik)
nodes["Exit Tol Tambak Sumur (A)"].add_neighbor(nodes["Royal
Hospital Surabaya (B)"], 5.4)
nodes["Exit Tol Tambak Sumur (A)"].add_neighbor(nodes["RSUD
Surabaya Timur (J)"], 5.2)
nodes["Exit Tol Tambak Sumur (A)"].add_neighbor(nodes["RS
Mitra Keluarga Pondok Tjandra (E)"], 0.7)

nodes["Royal Hospital Surabaya (B)"].add_neighbor(nodes["RS
Gotong Royong (C)"], 5.1)

nodes["RS Gotong Royong (C)"].add_neighbor(nodes["RSUD Haji
Provinsi Jawa Timur (I)"], 4.5)
nodes["RS Gotong Royong (C)"].add_neighbor(nodes["RS Manyar

```

```
Medical Surabaya (D)", 2.8)
nodes["RS Gotong Royong (C)"].add_neighbor(nodes["RS Premiere
Surabaya (F)", 3.1)

nodes["RS Mitra Keluarga Pondok Tjandra
(E)"].add_neighbor(nodes["RSU Bunda Waru (H)", 2.4)

nodes["RS Premiere Surabaya (F)"].add_neighbor(nodes["RS
Ubaya (G)", 1.9]

nodes["RSU Bunda Waru (H)"].add_neighbor(nodes["RSU Prima
Husada (K)", 0.6]

# Fungsi untuk menjalankan Dijkstra dari GUI
def find_shortest_path():
    start_name = start_combobox.get()
    end_name = end_combobox.get()

    if start_name and end_name:
        if start_name == end_name:
            messagebox.showinfo("Rute", "Titik awal dan akhir
tidak boleh sama.")
            return

        start_node = nodes.get(start_name)
        end_node = nodes.get(end_name)

        if start_node and end_node:
            result = dijkstra(start_node, end_node)
            messagebox.showinfo("Rute Terpendek", result)
        else:
            messagebox.showerror("Error", "Node tidak
```

```
valid.")

else:
    messagebox.showerror("Error", "Pilih titik awal dan
akhir terlebih dahulu.")

# Membuat GUI
app = tk.Tk()
app.title("Dijkstra Shortest Path")
app.geometry("500x300")

# Label dan dropdown untuk memilih titik awal
start_label = tk.Label(app, text="Titik Awal:",
font=("Arial", 12))
start_label.pack(pady=5)
start_combobox = ttk.Combobox(app, state="readonly",
width=40)
start_combobox["values"] = list(nodes.keys())
start_combobox.pack(pady=5)

# Label dan dropdown untuk memilih titik akhir
end_label = tk.Label(app, text="Titik Akhir:", font=("Arial",
12))
end_label.pack(pady=5)
end_combobox = ttk.Combobox(app, state="readonly", width=40)
end_combobox["values"] = list(nodes.keys())
end_combobox.pack(pady=5)

# Tombol untuk mencari rute terpendek
find_button = tk.Button(app, text="Cari Rute Terpendek",
command=find_shortest_path, bg="green", fg="white",
font=("Arial", 12))
find_button.pack(pady=20)
```

```
app.mainloop()
```

Gambar 4.2 *Code Script* untuk Menampilkan *GUI* dan Pencarian Rute Terpendek dengan Algoritma Dijkstra

Code ini akan menampilkan *interface* yang memungkinkan pengguna untuk memilih titik awal berupa Pintu Exit Tol Tambak Sumur serta rumah sakit yang berada di wilayah Surabaya. Kemudian, sistem akan menampilkan *output* berupa rute tercepat antara kedua titik dengan mempertimbangkan berbagai rute yang tersedia. Untuk mengetahui secara visual letak rumah sakit terhadap titik awal, kami menambahkan visualisasi berupa peta yang akan mengarahkan pengguna menuju rumah sakit tujuan. Berikut merupakan *code script* untuk menampilkan visualisasi tersebut.

```
!pip install folium

import tkinter as tk
from tkinter import messagebox, ttk
import folium
import webbrowser

# Data lokasi dan jarak dalam kilometer
tol_tambak_sumur_coords = (-7.344223, 112.776719) # Koordinat Pintu Exit Tol Tambak Sumur
hospitals = {
    "RSUD Surabaya Timur": {"coords": (-7.326444, 112.784041)},
    "RS Gotong Royong": {"coords": (-7.306387, 112.787441)},
    "RS Ubaya": {"coords": (-7.314095, 112.765730)},
    "RS Mitra Keluarga Pondok Tjandra": {"coords": (-7.314095, 112.765730)}
```

```
(-7.344303, 112.775406)},  
    "RS Premiere Surabaya": {"coords": (-7.298850,  
112.758370)},  
    "RSU Bunda Waru": {"coords": (-7.363487, 112.768248)},  
    "Royal Hospital Surabaya": {"coords": (-7.336043,  
112.723533)},  
    "RSU Prima Husada": {"coords": (-7.341423, 112.707838)},  
    "RSUD Haji Jawa Timur": {"coords": (-7.353929,  
112.772970)},  
    "RS Manyar Medical Surabaya": {"coords": (-7.285398,  
112.776765)},  
}  
  
# Fungsi untuk memperbarui dropdown  
def update_dropdown():  
    combobox["values"] = [name for name in hospitals.keys()]  
  
# Fungsi untuk menampilkan peta  
def show_map():  
    selected_item = combobox.get()  
    if selected_item:  
        hospital_name = selected_item  
        if hospital_name in hospitals:  
            hospital_coords =  
hospitals[hospital_name]["coords"]  
  
            # Membuat peta  
            map_tambak_sumur =  
folium.Map(location=tol_tambak_sumur_coords, zoom_start=15)  
            folium.Marker(  
                location=tol_tambak_sumur_coords,  
                popup="Pintu Exit Tol Tambak Sumur",
```

```

        icon=folium.Icon(color="green", icon="road")
    ) .add_to(map_tambak_sumur)
    folium.Marker(
        location=hospital_coords,
        popup=hospital_name,
        icon=folium.Icon(color="red",
icon="hospital")
    ) .add_to(map_tambak_sumur)

    # Simpan peta dan buka di browser
    map_file = "hospital_map.html"
    map_tambak_sumur.save(map_file)
    webbrowser.open(map_file)
else:
    messagebox.showerror("Error", "Rumah sakit tidak
ditemukan!")
else:
    messagebox.showerror("Error", "Pilih rumah sakit
terlebih dahulu!")

# Fungsi untuk menambahkan rumah sakit baru
def add_hospital_form():
    # Sembunyikan tombol tambah rumah sakit dan tampilkan
form
    btn_add.pack_forget()
    form_frame.pack(pady=10)

# Fungsi untuk menambahkan rumah sakit baru
def add_hospital():
    name = entry_name.get()
    try:
        lat = float(entry_lat.get())

```

```
    lon = float(entry_lon.get())
    if name in hospitals:
        messagebox.showerror("Error", "Rumah sakit sudah
terdaftar.")
    else:
        hospitals[name] = {"coords": (lat, lon),
"distance": 0}
        update_dropdown() # Perbarui dropdown dengan
data baru
        messagebox.showinfo("Sukses", f"Rumah sakit
{name} berhasil ditambahkan!")
        # Reset form dan sembunyikan
        form_frame.pack_forget()
        btn_add.pack(pady=10)
    except (ValueError, TypeError):
        messagebox.showerror("Error", "Input tidak valid.")

# GUI menggunakan Tkinter
app = tk.Tk()
app.title("Pencarian Rumah Sakit dari Pintu Exit Tol Tambak
Sumur")
app.geometry("500x500")

frame_title = tk.Frame(app)
frame_title.pack(pady=10)
frame_buttons = tk.Frame(app)
frame_buttons.pack(pady=10)

# Label judul
label_title = tk.Label(frame_title, text="Pencarian Rumah
Sakit Terdekat", font=("Arial", 16))
label_title.pack()
```

```
# Dropdown untuk memilih rumah sakit
combobox_label = tk.Label(frame_buttons, text="Pilih Rumah
Sakit:", font=("Arial", 12))
combobox_label.pack(pady=5)
combobox = ttk.Combobox(frame_buttons, state="readonly",
width=40)
combobox.pack()
update_dropdown() # Isi dropdown dengan data awal

# Tombol untuk menampilkan peta
btn_map = tk.Button(frame_buttons, text="Tampilkan Peta",
command=show_map, bg="green", fg="white", width=25)
btn_map.pack(pady=10)

# Tombol untuk menambah rumah sakit baru
btn_add = tk.Button(frame_buttons, text="Tambah Rumah Sakit",
command=add_hospital_form, bg="orange", fg="white", width=25)
btn_add.pack(pady=10)

# Form untuk menambah rumah sakit baru
form_frame = tk.Frame(app)
form_frame.pack_forget() # Awalnya disembunyikan

label_name = tk.Label(form_frame, text="Nama Rumah Sakit:",
font=("Arial", 12))
label_name.grid(row=0, column=0, sticky="e", padx=5, pady=5)
entry_name = tk.Entry(form_frame, width=30)
entry_name.grid(row=0, column=1, padx=5, pady=5)

label_lat = tk.Label(form_frame, text="Latitude:",
font=("Arial", 12))
```

```
label_lat.grid(row=1, column=0, sticky="e", padx=5, pady=5)
entry_lat = tk.Entry(form_frame, width=30)
entry_lat.grid(row=1, column=1, padx=5, pady=5)

label_lon = tk.Label(form_frame, text="Longitude:",
font=("Arial", 12))
label_lon.grid(row=2, column=0, sticky="e", padx=5, pady=5)
entry_lon = tk.Entry(form_frame, width=30)
entry_lon.grid(row=2, column=1, padx=5, pady=5)

app.mainloop()
```

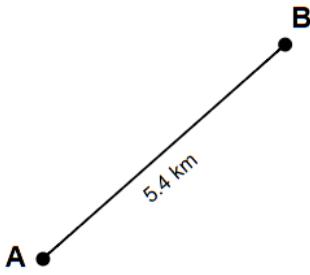
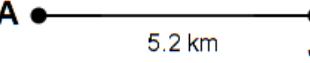
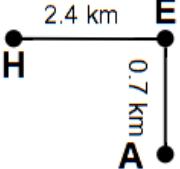
Gambar 4.3 *Code Script* untuk Menampilkan Peta

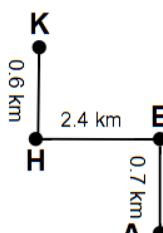
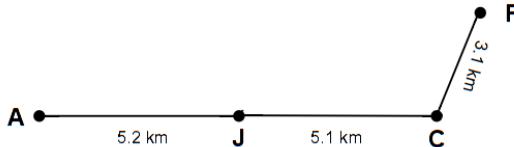
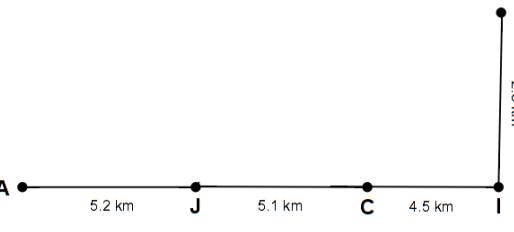
Selain visualisasi peta, *code script* diatas juga menyajikan pilihan bagi pengguna untuk menambahkan daftar rumah sakit ke dalam sistem. Pengguna hanya perlu memasukkan nama rumah sakit beserta koordinat geografisnya, dan rumah sakit tersebut akan terdaftar pada sistem. Dengan fitur ini, jumlah rumah sakit yang dapat dipilih oleh pengguna akan semakin banyak sehingga memberikan lebih banyak alternatif dalam situasi darurat.

Hasil penelitian menunjukkan bahwa terdapat beberapa rute terpendek dari lokasi awal Pintu Exit Tol Tambak Sumur ke setiap rumah sakit di Surabaya. Berdasarkan hasil analisis yang telah dilakukan menggunakan algoritma Dijkstra, terlihat bahwa tidak semua rute terpendek memiliki perbedaan yang signifikan sehingga diindikasikan adanya kemungkinan untuk melakukan perjalanan ke satu rumah sakit dengan melibatkan kunjungan ke rumah sakit lainnya. Hal ini sangat menguntungkan apabila terjadi kondisi darurat dalam skala besar, misalnya kecelakaan beruntun, tabrakan antar kendaraan bermuatan besar, serta bencana alam. Di samping itu, ketersediaan peralatan medis seringkali juga menjadi hambatan dalam pemilihan rumah sakit. Dengan adanya rute tercepat yang dapat menghubungkan satu rumah

sakit dengan rumah sakit lainnya, pengguna dapat menentukan rumah sakit yang sesuai dengan kebutuhan medisnya sehingga dapat segera memperoleh penanganan yang sesuai. Berikut ini merupakan hasil analisis pencarian jarak terpendek antara Rumah Sakit tujuan dan Pintu Exit Tol Tambak Sumur :

Tabel 4.2 Rute Terpendek Rumah Sakit dengan Node dan Graph

| No | Graf | Tujuan | Rute | Jarak |
|----|---|--------|-------|--------|
| 1 | A (root) | A | - | - |
| 2 |  | B | A-B | 5.4 km |
| 3 |  | J | A-J | 5.2 km |
| 4 |  | E | A-E | 0.7 km |
| 5 |  | H | A-E-H | 3.1 km |

| | | | | |
|----|---|---|-----------|---------|
| 6 |  | C | A-J-C | 10.3 km |
| 7 |  | K | A-E-H-K | 3.7 km |
| 8 |  | F | A-J-C-F | 13.4 km |
| 9 |  | I | A-J-C-I | 14.8 km |
| 10 |  | G | A-J-C-F-G | 15.3 km |
| 11 |  | D | A-J-C-I-D | 17.6 km |

Berdasarkan hasil rute terpendek dengan representasi titik dan garis di atas, informasi lengkap mengenai rute terpendek menuju rumah sakit di Surabaya dapat dilihat pada tabel berikut.

Tabel 4.3 Hasil Analisis Rute Terpendek Rumah Sakit dari Exit Tol

| No | Lokasi Awal | Tujuan | Rute | Jarak |
|----|-----------------------|-------------------------------------|---|--------|
| 1 | Exit Tol Tambak Sumur | RS Mitra Keluarga Pondok Tjandra | Exit Tol Tambak Sumur - RS Mitra Keluarga Pondok Tjandra | 700 m |
| 2 | Exit Tol Tambak Sumur | RSU Bunda Waru | Exit Tol Tambak Sumur - RS Mitra Keluarga Pondok Tjandra - RSU Bunda Waru | 3.1 km |
| 3 | Exit Tol Tambak Sumur | RSU Prima Husada | Exit Tol Tambak Sumur - RS Mitra Keluarga Pondok Tjandra - RSU Bunda Waru - RSU Prima Husada | 3.7 km |
| 4 | Exit Tol Tambak Sumur | RSUD Surabaya Timur | Exit Tol Tambak Sumur - RSUD Surabaya Timur | 5.2 km |
| 5 | Exit Tol Tambak Sumur | Royal Hospital Surabaya | Exit Tol Tambak Sumur - Royal | 5.4 km |

| | | | | |
|----|--------------------------|-------------------------------------|---|---------|
| | | | Hospital Surabaya | |
| 6 | Exit Tol Tambak Sumur | RS Gotong Royong | Exit Tol Tambak Sumur - RSUD Surabaya Timur - RS Gotong Royong | 10.3 km |
| 7 | Exit Tol Tambak Sumur | RS Premiere Surabaya | Exit Tol Tambak Sumur - RSUD Surabaya Timur - RS Gotong Royong - RS Premiere Surabaya | 13.4 km |
| 8 | Exit Tol Tambak Sumur | RSUD Haji Provinsi Jawa Timur | Exit Tol Tambak Sumur - RSUD Surabaya Timur - RS Gotong Royong - RSUD Haji Provinsi Jawa Timur | 14.8 km |
| 9 | Exit Tol Tambak Sumur | RS Ubaya | Exit Tol Tambak Sumur - RSUD Surabaya Timur - RS Gotong Royong - RS Premiere Surabaya - RS Ubaya | 15.3 km |
| 10 | Exit Tol Tambak Sumur | RS Manyar Medical Surabaya | Exit Tol Tambak Sumur - RSUD Surabaya Timur - RS Gotong Royong - | 17.6 km |

| | | | | |
|--|--|--|---|--|
| | | | RSUD Haji Provinsi Jawa Timur - RS Manyar Medical Surabaya | |
|--|--|--|---|--|

Berdasarkan hasil analisis dalam tabel diatas, dapat diketahui bahwa RS Mitra Keluarga Pondok Tjandra merupakan fasilitas kesehatan yang memiliki jarak terdekat dengan Exit Tol Tambak Sumur yakni sebesar 700 m. Di samping itu, RS Mitra Keluarga Pondok Tjandra juga menghubungkan dua rumah sakit lain yakni RSU Bunda Waru dan RSU Prima Husada yang dapat menjadi pilihan alternatif apabila salah satu rumah sakit sudah tidak mampu menampung pasien. Di sisi lain, RS Manyar Medical Surabaya dengan jarak 17.6 km merupakan fasilitas kesehatan yang terletak paling jauh dari Exit Tol Tambak Sumur. Rumah sakit ini merupakan alternatif terakhir dari rumah sakit lain yang searah dengannya dan penggunaan mobil ambulans juga dirasa mampu untuk menjangkau jarak ini.

"PENCARIAN LOKASI RUMAH SAKIT BERDASARKAN RUTE TERPENDEK DARI PINTU EXIT TOL TAMBAK SUMUR MENGGUNAKAN ALGORITMA DIJKSTRA"

"Algoritma Pencarian Rute Terpendek dengan Metode Dijkstra"

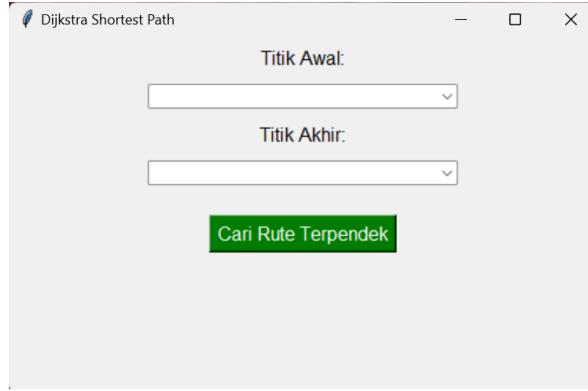
KESIMPULAN

Penelitian ini berhasil mengembangkan aplikasi berbasis algoritma Dijkstra untuk mencari rute terpendek dari Pintu Exit Tol Tambak Sumur menuju rumah sakit terdekat di Surabaya. Aplikasi ini dirancang untuk membantu pengguna, terutama dalam situasi darurat, dengan memberikan rekomendasi rute berdasarkan jarak terpendek. Penerapan algoritma Dijkstra, yang menggunakan graf berbobot untuk merepresentasikan hubungan antara lokasi, telah membuktikan keefektifannya dalam memecahkan masalah *shortest path problem* dengan akurasi tinggi. Data yang digunakan melibatkan koordinat geografis dari rumah sakit yang diambil menggunakan Google Maps, yang kemudian divisualisasikan dalam bentuk graf dan peta interaktif untuk mempermudah pengguna memahami rute.

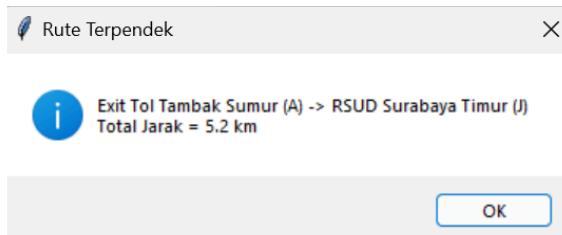
Hasil analisis menunjukkan bahwa rumah sakit RS Mitra Keluarga Pondok Tjandra memiliki rute terpendek sejauh 700 meter dari Pintu Exit Tol Tambak Sumur, diikuti oleh beberapa rumah sakit lain, seperti RSU Bunda Waru dengan jarak 3,1 km dan RSU Prima Husada dengan jarak 3,7 km. Dengan visualisasi peta yang dibuat menggunakan pustaka *Folium*, pengguna dapat melihat rute secara langsung, yang memberikan kemudahan navigasi. Sistem ini juga memungkinkan penambahan rumah sakit baru secara dinamis, memperluas cakupan aplikasinya untuk kebutuhan masa depan. Secara keseluruhan, aplikasi ini tidak hanya meningkatkan efisiensi waktu dalam pencarian fasilitas kesehatan, tetapi juga memberikan solusi praktis untuk masalah transportasi dalam keadaan darurat di jalan tol.

LAMPIRAN

- GUI :



Gambar 1. Tampilan awal aplikasi GUI

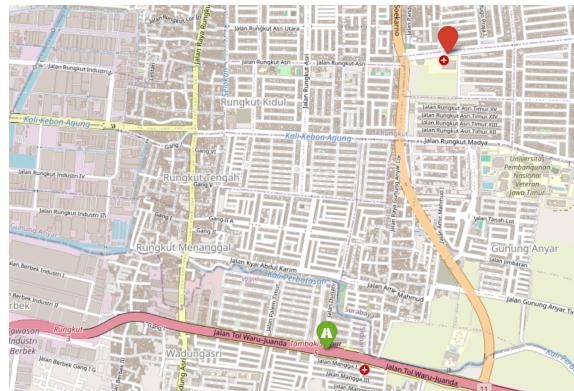


Gambar 2. Tampilan *output* aplikasi GUI

- Maps :



Gambar 3. Tampilan awal peta atau maps



Gambar 4. Tampilan output peta atau maps