

# SIG Penentuan Rute Terdekat Menuju Faskes di Sidoarjo Menggunakan Dynamic Dijkstra

I Kadek Dwi Nuryana<sup>1</sup>, Mohammad Aris Saputra<sup>2</sup>

<sup>1,2</sup>jurusan Teknik Informatika Fakultas Teknik Universitas Negeri Surabaya

<sup>1</sup>[dwinuryana@unesa.ac.id](mailto:dwinuryana@unesa.ac.id)

<sup>2</sup>[mohammadaris.18041@mhs.unesa.ac.id](mailto:mohammadaris.18041@mhs.unesa.ac.id)

**Abstrak**— Pencarian rute terdekat sangat dibutuhkan untuk menuju Fasilitas Kesehatan. Hal ini karena dibutuhkannya penindakan secepat mungkin untuk menangani pasien agar tidak terjadi hal yang tidak diinginkan. Algoritma Dijkstra merupakan salah satu algoritma yang handal dalam menentukan rute terdekat antara dua titik. Dijkstra dapat digunakan secara langsung dalam penentuan rute dengan menghitung bobot di setiap titik yang terhubung dari titik sumber sampai dengan titik tujuan. Bobot yang digunakan oleh Dijkstra statis bisa berupa jarak antar titik yang terhubung. Namun dalam kondisi nyata terdapat data lalu lintas yang bisa digunakan untuk memberi bobot antar titik agar bisa menentukan rute yang benar-benar sesuai. Oleh karena itu dibutuhkan pengembangan dari algoritma Dijkstra statis agar bisa mendeteksi perubahan bobot yang terjadi. Pada penelitian sebelumnya diusulkan algoritma Dynamic Dijkstra yang bertujuan untuk mendinamisasi algoritma Dijkstra statis. Pada metode ini digunakan struktur Data Retroaktif yaitu *Retroactive Priority Queue* (Antrian Prioritas Retroaktif), dengan struktur data ini algoritma Dijkstra dapat mendeteksi perubahan bobot saat dieksekusi sehingga dapat mengakomodasikan perubahan-perubahan yang terjadi akibat dari perubahan bobot tersebut. Pengujian dilakukan pada aplikasi penunjuk rute terdekat menuju Faskes di Sidoarjo, hasilnya aplikasi dapat berjalan dengan baik untuk merekomendasikan Faskes terdekat dan menunjukkan rute dari titik sumber (pengguna) menuju Faskes terdekat. Perbandingan akurasi dengan aplikasi lain yang serupa juga dilakukan dengan hasil 85,72 % untuk *Google Maps* dan 75,9 % untuk *Bing Maps*.

**Kata Kunci**— *Dijkstra, Retroactive Priority Queue, Dynamic Dijkstra, Rute Terdekat Dinamis, Faskes.*

## I. PENDAHULUAN

Kebutuhan medis merupakan salah satu kebutuhan masyarakat yang sangat penting. Terlebih lagi pada saat ini muncul berbagai macam penyakit baru yang bahkan sampai menyebabkan pandemi. Dengan kondisi semacam itu kebutuhan medis sering kali dianggap sebagai kebutuhan darurat sehingga dibutuhkan penanganan yang cepat terhadap pasien agar tidak menyebabkan penyakit yang lebih serius. Sylvana B. dalam [1] menjelaskan bahwa angka kematian di Indonesia tujuh puluh persen terjadi saat pasien belum sampai di Rumah Sakit, dan tiga puluh persennya meninggal di Rumah Sakit. Hal ini disebabkan oleh penanganan menuju rumah sakit yang kurang maksimal melalui lajur darat.

Perkembangan teknologi yang semakin dinamis membuat segala sesuatu menjadi lebih mudah. Salah satunya mengenai perkembangan peta yang semula hanyalah sebuah gambar saat ini sudah menjadi peta digital dan hampir dimiliki oleh setiap orang pada perangkat telepon genggamnya. Tidak cukup

sampai di situ, peta digital juga mengalami perkembangan lebih pesat dengan dapat ditampilkannya informasi-informasi yang dapat membantu kehidupan manusia, peta ini bisa disebut sebagai Sistem Informasi Geografis (SIG). Sistem Informasi Geografis (SIG) dapat diidentifikasi sebagai teknik dalam menganalisis dan mengolah sebuah data geografis atau spasial dan ditampilkan dalam bentuk peta digital [2]. Didukung oleh kemajuan teknologi saat ini dan metode yang sudah ada SIG mampu menjadi sebuah media pendukung berbagai keputusan berdasarkan data informasi geografis atau spasial yang diberikan. Sistem Pendukung Keputusan Spasial difungsikan sebagai media pembantu pengambilan keputusan pengguna dengan memecahkan keputusan spasial semi-terstruktur [3].

Dengan meningkatnya keselarasan Sistem Informasi Geografis (SIG) yang dapat digunakan sebagai dasar Sistem Pengambilan Keputusan menyebabkan integrasi di banyak bidang [3]. Dari berbagai bidang Sistem Informasi Geografis dapat menjanjikan kemampuan baru yang cukup besar untuk pengambilan keputusan berdasarkan informasi spasial. Sistem ini dapat digunakan untuk mengambil keputusan penentuan jalur dengan cepat dari titik awal sampai dengan titik tujuan.

Terdapat banyak metode yang bisa diterapkan dalam Sistem Informasi Geografis untuk pengambilan keputusan. Salah satu metode yang bisa digunakan adalah metode Dijkstra Algorithm. Dijkstra Algorithm mengalami beberapa peningkatan melalui berbagai cara pada beberapa penelitian sebelumnya. Salah satu cara dalam meningkatkan Dijkstra Algorithm yaitu dengan menggunakan Struktur Data Retroaktif Antrian Prioritas (*Retroactive Data Structure Priority Queue*) yang diperkenalkan oleh [4]. Permasalahan yang mendasari untuk diusulkannya algoritma ini yaitu masalah jalur terpendek dari simpul sumber ke simpul lain dari graf dengan skenario yang dinamis. Dalam Struktur Data Retroaktif, perubahan data yang terjadi dalam struktur data karena adanya modifikasi di masa lalu secara otomatis diakomodasi di masa sekarang tetapi dalam retroaktif yang tidak terlupakan, sehingga pengguna dapat diberitahu mengenai operasi pertama yang dilakukan karena modifikasi yang dilakukan pada masa lalu [4]. Karena struktur data ini dapat membantu mempertahankan urutan historis mengenai peristiwa yang terjadi dalam struktur data, jadi ini dapat digunakan untuk mendinamisasi algoritma Dijkstra statis menjadi dinamis atau bisa disebut sebagai Dynamic Dijkstra. Algoritma Dynamic Dijkstra menggunakan Antrian Prioritas Retroaktif (*Priority Queue*) yang merupakan modifikasi dari algoritma standar Dijkstra dengan modifikasi menjadikan graf bersifat dinamis dan setiap perubahan bobot tepi graf juga dimasukkan ke dalam antrian prioritas selama algoritma di

eksekusi, antrian prioritas ini dapat mengakomodasi perubahan dan mengembalikan operasi urutan pertama yang telah terpengaruh karena perubahan bobot [4].

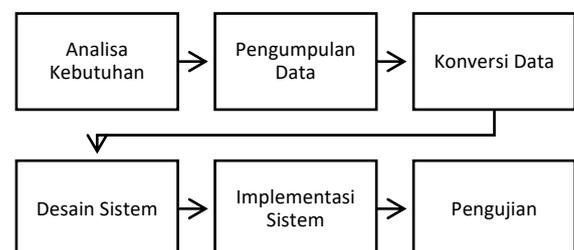
Berikut hasil beberapa penelitian sebelumnya yang membahas mengenai metode Dijkstra Algorithm untuk penentuan jarak terpendek antara lain : [3] melakukan studi mengenai metode Improved Dijkstra Algorithm dengan membatasi cakupan wilayah penghitungan sebagai penentu jarak terpendek menuju Rumah Sakit dan hasilnya Algoritma Dijkstra dengan sedikit perbaikan dapat mengalami peningkatan kecepatan eksekusi dengan lebih baik namun agar akurasinya lebih tepat dan cepat dibutuhkan perbaikan lebih lanjut mengenai data yang digunakan, [5] memperkenalkan metode Improved Dijkstra Algorithm dengan optimalisasi performa beberapa kali lipat pada peta grid-8 2D dengan cara memastikan bahwa setiap posisi hanya perlu menghitung satu kali sehingga dapat menghemat banyak waktu eksekusi dan pengujian yang dilakukan pada peta grid-8 2D menunjukkan bahwa Improved Dijkstra Algorithm dapat mempercepat algoritma Dijkstra menjadi beberapa kali lipat, [6] membuktikan dengan Dijkstra Algorithm mampu mengoptimalkan waktu, biaya serta keamanan dalam penanganan ataupun evakuasi banjir namun diperlukan pertimbangan mengenai kondisi jaringan jalan secara nyata mengenai kerusakan jalan dan kemacetan, [4] melakukan studi mengenai Dinamizing Dijkstra Algorithm menggunakan struktur data retroaktif untuk memecahkan masalah penentuan jalur terpendek dengan lebih dinamis dan hasilnya adalah metode Dynamic Dijkstra dapat memberikan solusi yang efektif untuk permasalahan jalur yang dinamis, namun biaya semua operasi sangat bergantung pada ketinggian struktur data pohon yang mendasarinya, oleh karena itu perlu selektif dalam memilih titik yang benar-benar terpengaruh oleh perubahan dan menerapkan pembaruan hanya pada bagian itu, [7] menggunakan Dijkstra Algorithm untuk penentuan rute pengiriman makanan terbaik baik ketika adanya informasi mengenai jaringan jalan maupun saat tidak adanya informasi mengenai jaringan jalan serta dapat memberikan navigasi yang akurat ketika informasi jalan tidak diketahui dan juga pada penelitian ini dapat dihasilkannya F1\_Score sebesar 8 % ketika informasi jalan tidak diketahui namun pengiriman makanan dapat terus menerima pesanan baru dalam pengiriman mereka, sehingga diperlukan penyesuaian rekomendasi rute secara dinamis untuk mempertimbangkan lebih dari satu tujuan.

Penelitian ini akan menggunakan metode Dynamic Dijkstra sebagai Sistem Pendukung Keputusan untuk menentukan rute terdekat menuju fasilitas kesehatan yang dituju. Dynamic Dijkstra dipilih karena dapat memperhitungkan jarak terpendek dari lokasi sumber ke lokasi tujuan dengan lebih dinamis. Pertimbangan mengenai dinamisasi dan pembaruan bobot berasal dari kondisi kemacetan dan lalu lintas jalan. Dengan menggunakan struktur data Antrian Prioritas Retroaktif (Retroactive Priority Queue) diharapkan algoritma ini dapat mendeteksi perubahan kondisi lalu lintas jalan sehingga memberikan rekomendasi rute yang terbaik. Data yang digunakan pada penelitian ini adalah data geografis dari Fasilitas Kesehatan maupun rute jalan yang diambil dari [8] dan

[9] dengan berbentuk shapefile. Data geografis yang diperoleh akan diolah menjadi data spasial dan kemudian direpresentasikan menjadi model vektor yang mengacu dan dibandingkan dengan data yang ada pada Google Maps menggunakan aplikasi QGIS. Model vektor terdiri dari poin, garis, dan polygon yang efektif untuk menyimpan data dengan batas-batas terpisah [2]. Data akan diseleksi dengan hanya menampilkan Fasilitas Kesehatan yang valid dan aktif. Data jalan diambil dari Bakosurtanal (Badan Koordinasi Survei dan Pemetaan Nasional) dan kondisi lalu lintas yang diatur dengan memilih titik koordinat secara random untuk keperluan simulasi penelitian dengan pertimbangan biaya eksekusi program, dan diharapkan kedepannya terdapat data lalu lintas *real-time* yang dapat digunakan dengan baik. Data spasial yang sudah diolah akan ditampilkan melalui Sistem Informasi Geografis (SIG) berbasis Web yang dapat digunakan langsung oleh pengguna baik melalui desktop maupun perangkat mobile. Pengujian aplikasi akan menggunakan Teknik Alpha-testing dan Black box-testing dan untuk pengujian metode akan dilakukan dengan uji akurasi serta jarak yang ditempuh untuk dibandingkan dengan Aplikasi penyedia layanan rute lain yang sudah ada yaitu Google Maps dan Bing Maps. Penelitian ini dibuat dengan dikhususkan untuk pencarian rute menuju fasilitas kesehatan terdekat dari pengguna dengan mudah, karena cukup banyak orang yang tidak mengetahui nama dan lokasi dari fasilitas kesehatan yang ada di dekatnya. Algoritma Dynamic Dijkstra yang digunakan pada penelitian ini juga diharapkan dapat memberikan satu rute yang tercepat dengan mempertimbangkan kondisi jalan saat algoritma dieksekusi. Dengan dibuatnya penelitian ini diharapkan dapat membantu pengguna ataupun pasien untuk menentukan perjalanan terdekat menuju fasilitas kesehatan agar pasien mendapatkan penanganan medis yang lebih cepat untuk menghindari hal yang tidak diinginkan.

## II. METODE PENELITIAN

Proses penelitian ini diatur dalam sebuah diagram alur penelitian. Berikut rangkaian proses yang ada pada penelitian ini, diawali dengan analisa kebutuhan, pengumpulan data-data yang dibutuhkan, kemudian data dikonversi ke dalam format data spasial (point, linestring), desain sistem, implementasi sistem, dan terakhir pengujian. Alur penelitian ini dapat dilihat pada gambar 1 berikut.



Gbr 1. Alur Penelitian

### A. Analisa Kebutuhan

Analisa Kebutuhan merupakan tahap awal yang dilakukan pada penilitan ini. Pada tahap ini dilakukan analisa mengenai kebutuhan apa saja yang dibutuhkan dalam sistem termasuk data-data yang akan diolah. Pada tahap ini juga akan dipertimbangkan fitur apa saja yang akan dibuat di dalam sistem.

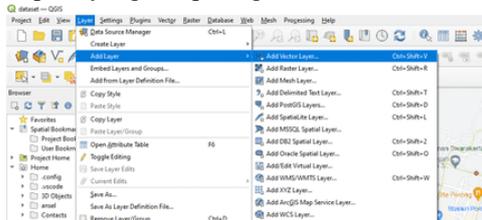
### B. Pengumpulan Data

Setelah mengetahui kebutuhan data apa saja yang diperlukan maka dilanjutkan ke tahap pengumpulan data. Data yang digunakan dan diolah dalam penelitian ini merupakan data spasial tersebut dan jalan yang ada di wilayah Sidoarjo. Data spasial tersebut dikumpulkan dari laman [8] dan laman [9]. Data yang didapatkan merupakan data yang berbentuk shapefile (SHP). Shape file merupakan format khusus dalam menyimpan data SIG yang dirancang oleh Environmental Systems Research Institute (ESRI) [10].

### C. Konversi Data

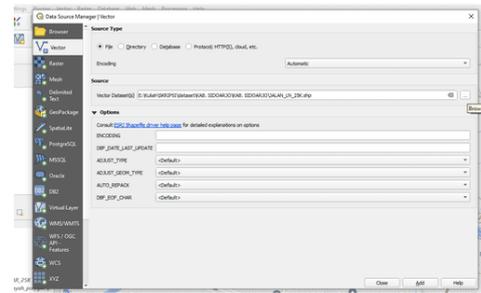
Data yang dikumpulkan terdiri dari data Fasilitas Kesehatan dan data geografis, Data geografis yang ada pada shapefile perlu dikonversi terlebih dahulu menjadi format data spasial. Data spasial merupakan format data untuk menyimpan sebuah koordinat (Latitude & Longitude) yang berupa sebuah point, linestring, maupun polygon[2]. Karena data yang diolah cukup banyak maka dibutuhkan perangkat lunak tambahan untuk membantu proses konversi dan input data ke dalam basis data sistem yaitu QGIS. QGIS merupakan perangkat lunak yang berfungsi untuk menampilkan, memanajemen, mengedit, Menyusun dan menganalisa data geografis. Berikut langkah untuk konversi data dari shapefile.

1. Tekan menu *layer* dan kemudian tekan *add vector layer* seperti yang ada pada gambar 2.



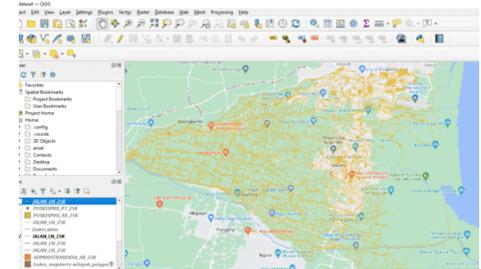
Gbr 2. Add New Layer QGIS

2. Pilih shape file dengan menekan tombol *browse* yang berupa tombol dengan simbol ..., tidak perlu mengubah pengaturan lain dan kemudian tekan tombol Add yang ada di bawah seperti yang tampil pada gambar 3.



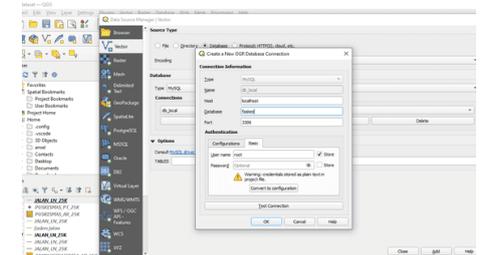
Gbr 3. Add Shapefile QGIS

3. *Shapefile* yang sudah ditambahkan akan tampil di dalam QGIS seperti pada gambar 4 berikut.

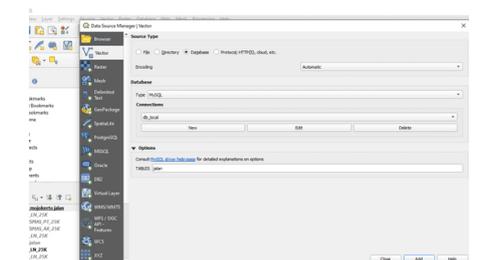


Gbr 4. Tampilan shapefile pada QGIS

4. Lakukan lagi yang ada pada langkah 1, namun dengan pilihan input layer dari database. Tekan *New* pada *Connections* dan inputkan database yang dipakai seperti pada gambar 5. Kemudian masukan nama *table* yang akan dipakai pada *Options TABLES* lalu tekan add seperti pada gambar 6.



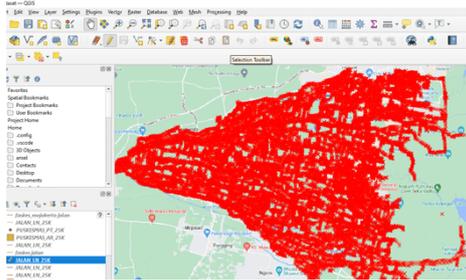
Gbr 5. Add Database Connection QGIS



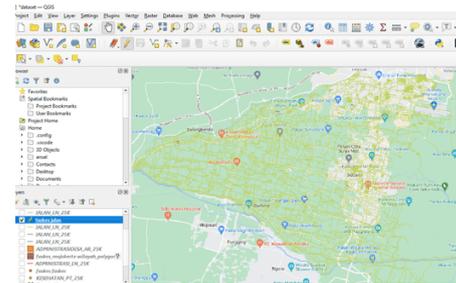
Gbr 6. Add Table QGIS

5. Pilih layer *shapefile* yang sudah dimasukan sebelumnya, tekan tombol *toggle editing* (ikon pensil), lalu tekan *ctrl + a* dan *ctrl + c* kemudian tampilan datanya akan menjadi seperti pada gambar 7. Kemudian buka layer table dari database yang sudah dimasukan, tekan tombol *toggle editing* dan paste data

yang sudah dicopy. Data akan menyesuaikan secara otomatis sesuai dengan tipe field yang ada pada shapefile. Dan hasilnya seperti yang ada pada gambar 8.



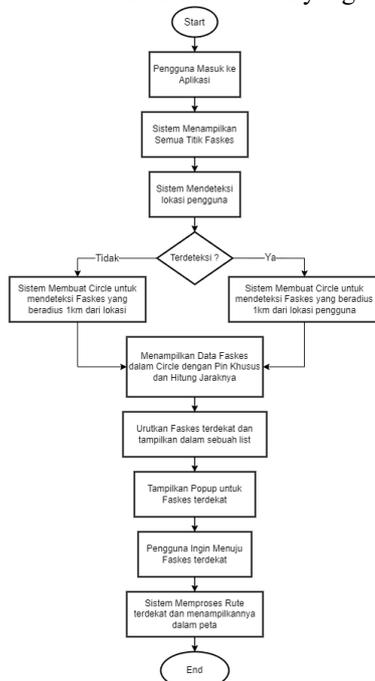
Gbr 7. Select dan copy semua data shapefile



Gbr 8. Hasil konversi data

**D. Desain Sistem**

Setelah kebutuhan dan data yang dibutuhkan telah diperoleh selanjutnya dilakukan proses desain sistem yang sesuai. Tujuannya adalah agar pengguna dapat menggunakan sistem dengan maksimal dan mudah dipahami. Dalam gambar 9 merupakan alur secara umum dari sistem yang akan dibuat.



Gbr 9. Alur Umum Aplikasi

Alur secara umum dari proses yang dilakukan oleh aplikasi ini adalah sebagai berikut :

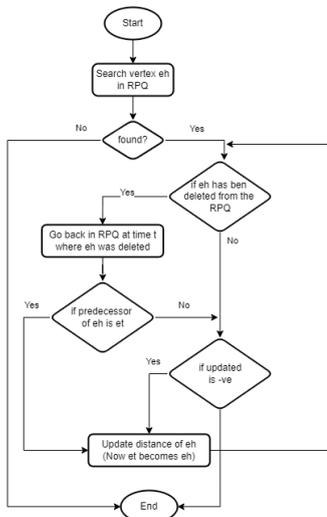
1. Pengguna terlebih dahulu masuk ke halaman app aplikasi melalui halaman beranda.
2. Sistem secara otomatis akan menampilkan data semua Faskes yang ada di dalam basis data dengan pin penanda berupa bulatan hijau.
3. Sistem melakukan proses deteksi lokasi pengguna, apabila lokasi pengguna dapat dideteksi maka sistem akan menampilkan data Faskes dengan radius jarak 1 km dari titik lokasi pengguna terdeteksi dengan pin khusus. Namun jika lokasi pengguna tidak terdeteksi, sistem akan menampilkan data Faskes dengan radius jarak 1 km dari titik lokasi default yang ada pada sistem. Pengguna dapat memindah lokasi secara manual dengan cara menggeser pin yang ada di peta maupun menekan salah satu lokasi yang ada pada peta agar sesuai dengan lokasi pengguna yang sebenarnya. Dan sistem akan melakukan deteksi Faskes dengan radius 1 km setiap ada pemindahan titik lokasi pengguna.
4. Data Faskes yang dideteksi diproses dengan menggunakan algoritma Dynamic Dijkstra untuk menentukan jarak antara Faskes dan lokasi pengguna. Data Faskes yang terdeteksi diurutkan berdasarkan jarak terdekat sampai terjauh dari pengguna dan di tampilkan pada rekomendasi terdekat.
5. Untuk Faskes yang berjarak paling dekat dengan pengguna akan menampilkan popup nama Faskes tersebut.
6. Jika pengguna ingin menuju ke Faskes tedekat, pengguna cukup menekan nama yang muncul pada popup. Sistem akan secara otomatis menghitung jaraknya sekali lagi menggunakan algoritma Dynamic Dijkstra dan menampilkan rutennya. Pengguna juga bisa memilih Faskes lain yang ada pada peta maupun rekomendasi dengan cara menekan salah satu pin yang ada pada peta maupun menekan salah satu Faskes pada list rekomendasi untuk menampilkan rute menuju Faskes tersebut.

**E. Implementasi Sistem**

Pada tahap ini akan dilakukan implementasi algoritma yang digabungkan dengan desain sistem yang sudah dibuat. Dalam proses penghitungan jarak terdekat dari pengguna pada proses aplikasi yang ditunjukkan pada gambar 3 digunakan algoritma Dynamic Dijkstra. Algoritma ini berfungsi untuk menghitung jarak terdekat dari 2 titik yang diberikan dengan mendeteksi dan mengakomodasi perubahan yang ada pada saat proses algoritma dijalankan [4]. Tujuannya adalah pengguna akan mendapatkan rute yang terbaik untuk menuju ke Faskes yang diinginkan. Proses yang ada pada Dynamic Dijkstra hampir sama seperti algoritma Dijkstra standar, yang membedakan adalah di dalam Dynamic Dijkstra terdapat pengecekan perubahan weight antar vertex dan mengakomodasi akibat dari perubahan tersebut. Untuk mendukung perubahan dinamis tersebut digunakan Retroactive Priority Queue (RPQ) dalam

algoritma ini. RPQ adalah struktur data yang secara efisien dapat mendukung modifikasi urutan operasi yang dilakukan [11].

Retroactive Priority Queue pertama kali diperkenalkan oleh Erik Demaine pada tahun 2007 yang merupakan bagian dari Retroactive Data Structure melalui [12] dengan tujuan untuk mendinamisasi algoritma statis. Kanat Tangwongsan melalui [13] memperkenalkan varian dari struktur data ini yaitu Non-oblivious Retroactive (Retroaktif yang tidak disadari). Varian inilah yang akan digunakan pada Dynamic Dijkstra, dalam struktur data retroaktif tersebut perubahan yang timbul dalam struktur data akan diakomodasi secara otomatis. Untuk menerapkan Retroactive Priority Queue, diperlukan sebuah cara untuk mempertahankan masa pakai dari semua elemen. Oleh karena itu dibuatlah 2 *binary tree* yaitu  $T_{in}$  untuk menampung data yang disisipkan (RPQ *active*) dan  $T_{dm}$  untuk menampung data yang sudah dihapus dari RPQ *active* (del-min).  $T_{in}$  mempertahankan pasangan kunci  $(x, t)$  dengan  $x$  sebagai item dan  $t$  menunjukkan waktu saat item disisipkan ke dalam RPQ. Sedangkan  $T_{dm}$  mempertahankan waktu dimana operasi del min dilakukan.  $T_{in}$  diurutkan berdasarkan *weght* dan waktu, dan untuk  $T_{dm}$  diurutkan berdasarkan waktu. Dalam Dynamic Dijkstra, pertama-tama dilakukan pemeriksaan terlebih dahulu apakah ada bobot atau jarak yang diperbarui dan mempengaruhi operasi yang dilakukan sampai saat ini, jika iya maka identifikasi operasi yang dilakukan pada titik yang mengalami perubahan bobot dan akomodasikan perubahan [4]. Berikut pada gambar 10 merupakan langkah-langkah algoritma Dynamic Dijkstra.



Gbr 10. Alur Retroactive Priority Queue Dynamic Dijkstra

Alur proses Retroactive Priority Queue di dalam Dynamic Dijkstra dijelaskan sebagai berikut :

1. eh merupakan vertex tujuan dari edge yang mengalami perubahan jarak, cari eh di dalam RPQ yang sedang aktif. Jika iya lanjutkan ke langkah 2, jika tidak lanjutkan ke langkah 5.
2. Jika entri yang sesuai dengan eh terdapat di dalam RPQ aktif, lanjut ke langkah 3, jika tidak lanjut ke langkah 4.

3. Jika pembaruan negative(bobot jarak berkurang),maka perbarui simpul et ke eh, et menjadi eh dan lanjutkan ke langkah 1, jika tidak lanjut ke langkah 5.
4. Pindah ke RPQ sebelum eh dihapus dan mengakomodasikan perubahan sesuai kebutuhan, kembali ke langkah 1.
5. Keluar.

Dikarenakan keterbatasan sumber daya dan biaya aplikasi ini belum bisa mendapatkan data lalu lintas secara real-time. Aplikasi ini memanfaatkan sistem yang memperbarui data bobot jalan secara acak dengan memilih antara 10 – 100 bobot yang diperbarui dengan menambah bobot sebanyak 5 untuk setiap jarak antar koordinat dari total koordinat yang diperoleh – 50. Hal ini bisa diubah melalui *controller*, dengan harapan kedepannya terdapat API lalu lintas yang bisa digunakan sehingga proses dinamisasi bobot dan rute menjadi lebih baik dan akurat yang sesuai dengan keadaan nyata. Refensi kode program yang diimplementasikan pada aplikasi ini diambil dari [14] dengan beberapa modifikasi dan penyesuaian khusus untuk mendukung proses dinamis menggunakan *Retroactive Priority Queue*.

#### F. Pengujian

Pada tahap ini merupakan tahap akhir dari proses penelitian, yaitu dengan melakukan pengujian sistem aplikasi dengan menggunakan pengujian alpha. Pengujian *alpha* adalah salah satu bentuk pengujian aplikasi yang dilakukan untuk memastikan apakah sebuah sistem aplikasi dapat berjalan dengan baik pada sistem operasi pengguna[15]. Teknik yang digunakan pada pengujian ini adalah teknik black box testing, merupakan sebuah teknik pengujian untuk menemukan adanya kesalahan yang berkaitan dengan fungsionalitas dan antarmuka sistem [16]. Pada pengujian ini juga dilakukan uji akurasi dengan melakukan perbandingan akurasi rute serta jarak tempuh yang dibandingkan oleh *Google Maps* dan *Bing Maps*.

### III. HASIL DAN PEMBAHASAN

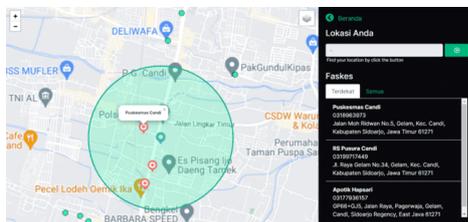
#### A. Implementasi Sistem

Penelitian ini diimplementasikan ke dalam aplikasi *web* untuk menjalankan fungsinya. Aplikasi ini dikembangkan dalam bahasa pemrograman PHP dan Javascript sebagai inti prosesnya, selain itu juga menggunakan HTML dan CSS sebagai struktur *web*. Teknologi yang digunakan pada aplikasi ini yaitu *Laravel Framework* sebagai kerangka kerja penuh aplikasi, serta *plugin Leaflet.js* untuk menampilkan map dan mengelola data geografis. Alasan dipilihnya aplikasi *web* sebagai media implementasi sistem ini karena agar pengguna dapat menggunakan aplikasi ini tanpa perlu memiliki ruang space di perangkatnya serta dapat mendukung lintas platform. Pada gambar 11, merupakan tampilan *landing page* awal pengguna mengakses *web* aplikasi.

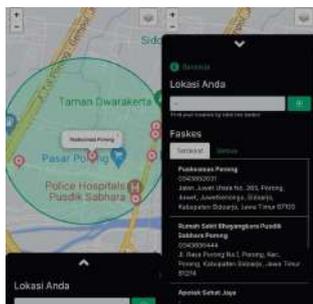


Gbr 11. Landing Page Aplikasi

Selanjutnya pengguna dapat menekan tombol mulai untuk menggunakan aplikasi. Pengguna dapat melihat beberapa informasi yang ditampilkan pada halaman ini yaitu, titik-titik Faskes yang ada, rekomendasi Faskes terdekat dari pengguna, serta semua Faskes yang ada pada aplikasi ini. Terdapat juga bar untuk menampilkan data lokasi pengguna saat ini dan di sampingnya terdapat tombol deteksi lokasi. Berikut pada gambar 12 menampilkan halaman aplikasi pada perangkat desktop dan gambar 13 menampilkan halaman aplikasi pada perangkat mobile.



Gbr 12. Halaman Aplikasi pada Desktop



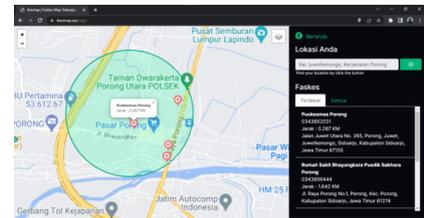
Gbr 13. Halaman Aplikasi pada Mobile

Berikut adalah beberapa hal yang bisa dilakukan dan didapatkan oleh pengguna terhadap aplikasi ini :

1. Mendeteksi lokasi pengguna saat ini,
2. Menggeser ataupun menekan titik di peta untuk memindahkan lokasi pengguna,
3. Mendapatkan informasi titik Fasilitas Kesehatan, baik yang berada di sekitar ataupun tidak selama masih di wilayah Kabupaten Sidoarjo.
4. Mencari rute menuju Fasilitas Kesehatan terdekat ataupun menuju Fasilitas Kesehatan lain yang diinginkan pengguna.

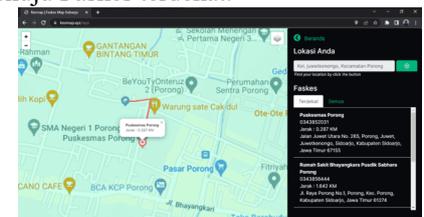
Untuk melakukan proses inti yang ditawarkan oleh aplikasi ini pengguna dapat melakukan beberapa langkah sebagai berikut :

1. Pengguna membuka halaman aplikasi, aplikasi akan mendeteksi lokasi pengguna secara otomatis untuk menentukan titik awal, jika tidak terdeteksi aplikasi akan menentukan titik awal dari titik *default* yang tersedia pada aplikasi.
2. Pengguna akan mendapatkan rekomendasi Faskes dengan radius 1km dari titik lokasinya secara otomatis. Sistem akan menampilkan *popup* berisikan nama Faskes pada titik Faskes terdekat. Bisa dilihat pada gambar 14 berikut.



Gbr 14. Aplikasi Mendeteksi Faskes Terdekat

3. Pengguna bisa menekan *popup* nama yang telah tampil untuk mengetahui rute menuju Faskes tersebut, atau jika pengguna menginginkan faskes lain bisa dilakukan dengan cara menekan salah satu data Faskes pada bagian rekomendasi Faskes pada *sidebar*.
4. Sistem akan menampilkan rute terdekat menuju Faskes yang hendak dituju oleh pengguna, berikut pada gambar 15 menampilkan aplikasi saat menunjukkan rute menuju Faskes terdekat.



Gbr 15. Aplikasi Menunjukkan Rute Menuju Faskes Terdekat

## B. Pengujian

Proses pengujian yang dilakukan meliputi pengujian *Alpha Testing* dengan Teknik *blackbox* yang bertujuan untuk memastikan fungsi dan antarmuka system telah berjalan dengan baik sebelum aplikasi dirilis. Selanjutnya dilakukan juga pengujian akurasi yang akan membandingkan hasil rute yang dihasilkan oleh aplikasi dengan yang ada pada *Google Maps* dan *Bing Maps* dengan tujuan untuk mengkaji system yang dibuat apakah sudah sesuai dengan sistem lain yang serupa.

Pada sistem aplikasi ini terapat 103 data Faskes yang ada di Sidoarjo terdiri dari 23 data Rumah Sakit, 26 data Puskesmas, 45 data Klinik, dan 7 data Dokter Spesialis. Pengujian akan dilakukan di 3 titik yang mewakili 3 Kecamatan dengan jumlah penduduk tertinggi tahun 2020 di Sidoarjo yang dihimpun dari [17] dan data alamat dari *Google Maps*, dengan uraian yang ditunjukkan pada tabel 1 berikut.

TABEL I  
TABEL TITIK AWAL PENGUJIAN

No	Koordinat (Lat, Lng)	Alamat
1	-7.351178038356194, 112.69303085680367	Kalijaten IB 36-10, Kalijaten, Kec. Taman, 61257
2	-7.452847646127449, 112.72069391336308	Jl. Kartini 103-115, Gabahan, Sidokumpul, Kec. Sidoarjo, 61218
3	-7.354081406641987, 112.74789337919418	Belahan, Wedoro, Kec. Waru, 61256

Proses pengujian dilakukan pada perangkat *notebook* dengan spesifikasi sebagai berikut :

- Processor i5-7200U 2.5 GHz
- RAM 12GB
- Penyimpanan 1TB
- Sistem Operasi Windows 10
- Resolusi 1336x780
- Browser : Google Chrome

### 1) Pengujian Alpha Test

Pada tahap ini aplikasi SIG Penentuan Jarak Terdekat Menuju Faskes di Sidoarjo berbasis web diuji untuk pertama kali sebelum aplikasi dirilis dengan menggunakan teknik *blackbox* untuk mengkaji kesesuaian fungsionalitas dan antarmuka dari aplikasi. Hasil dari pengujian alpha ditunjukkan pada tabel 2.

TABEL 2  
HASIL PENGUJIAN ALPHA APLIKASI

No.	Hal Yang Diuji	Hasil Pengujian	Pengamatan
1	Masuk ke <i>landing page web</i>	(✓) Berhasil ( ) Gagal	Diterima
2	Masuk ke halaman aplikasi	(✓) Berhasil ( ) Gagal	Diterima
3	Mendeteksi lokasi pengguna / tetapkan titik lokasi sesuai titik <i>default</i>	(✓) Berhasil ( ) Gagal	Diterima
4	Mendeteksi Faskes dengan radius jarak 1 km dari pengguna	(✓) Berhasil ( ) Gagal	Diterima
5	Memberikan data rekomendasi Faskes terdekat	(✓) Berhasil ( ) Gagal	Diterima

6	Mencari rute dari titik pengguna ke titik Faskes Rumah Sakit Ibu dan Anak Kirana	(✓) Berhasil ( ) Gagal	Diterima
7	Menampilkan rute dari titik pengguna ke titik Faskes Rumah Sakit Ibu dan Anak Kirana	(✓) Berhasil ( ) Gagal	Diterima

### 2) Pengujian Akurasi

Pengujian akurasi dilakukan dengan membandingkan data yang didapatkan dari aplikasi SIG Penentuan Jarak Terdekat Menuju Faskes di Sidoarjo berbasis web dengan aplikasi serupa yaitu *Google Maps* dan *Bing Maps* yang bertujuan untuk mengkaji kesesuaian sistem yang dibangun dengan sistem yang sudah ada. Data lalu lintas yang ada di sistem merupakan data pada koordinat *random* yang diatur oleh system. Akurasi yang dimaksudkan di sini adalah perbandingan jarak antara aplikasi SIG dengan aplikasi yang sudah ada sebelumnya, tanpa memperhitungkan kesamaan pemilihan rute.

#### Percobaan Pada Titik 1

Percobaan pertama dilakukan pada titik 1 dengan koordinat Latitude, Longitude (-7.351178038356194, 112.69303085680367) yang beralamat di Kalijaten IB 36-10, Kalijaten, Kec. Taman, Kab. Sidoarjo 61257. Hasil dari percobaan pada titik 1 dapat dilihat pada tabel 3 untuk pengujian pada *Google Maps* dan table 4 untuk pengujian pada *Bing Maps*. Perbandingan jarak secara grafik ditunjukkan pada gambar 16.

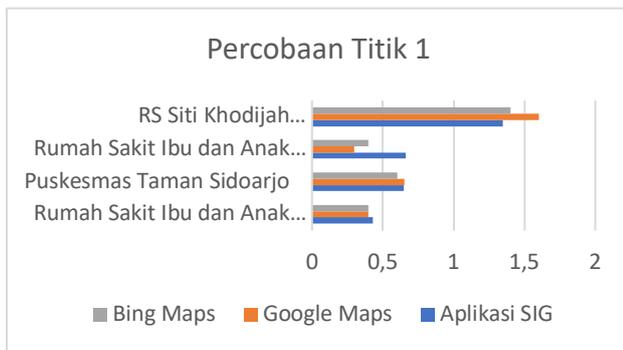
TABEL 3  
HASIL PENGUJIAN AKURASI PADA TITIK 1 GOOGLE MAPS

No	Nama Faskes	Aplikasi SIG	Google Maps	
			Jarak	Akurasi
1	Rumah Sakit Ibu dan Anak KIRANA	0,431 km	0,4 km	92,8 %
2	Puskesmas Taman Sidoarjo	0,647 km	0,65 km	99,5 %
3	Rumah Sakit Ibu dan Anak Soerya	0,662 km	0,3 km	45 %
4	RS Siti Khodijah	1,354 km	1,6 km	84 %

	Muhammadiyah			
--	--------------	--	--	--

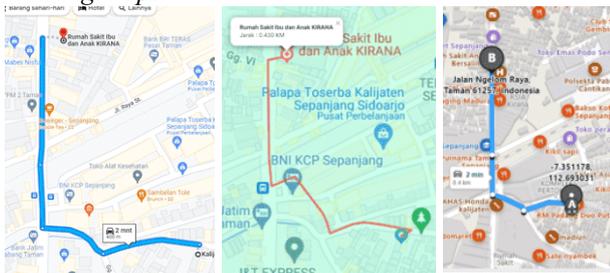
TABEL 4  
HASIL PENGUJIAN AKURASI PADA TITIK 1 BING MAPS

No	Nama Faskes	Aplikasi SIG	Bing Maps	
			Jarak	Akurasi
1	Rumah Sakit Ibu dan Anak KIRANA	0,431 km	0,4 km	92,8 %
2	Puskesmas Taman Sidoarjo	0,647 km	0,6 km	92,7 %
3	Rumah Sakit Ibu dan Anak Soerya	0,662 km	0,4 km	60 %
4	RS Siti Khodijah Muhammadiyah	1,354 km	1,4 km	96 %

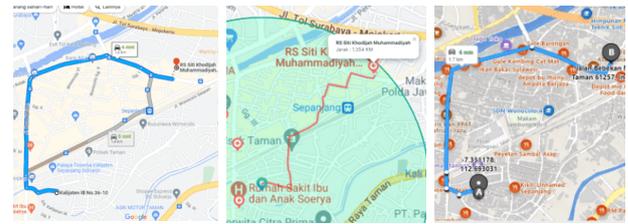


Gbr 16. Grafik Perbandingan Percobaan Titik 1

Berikut pada gambar 17 dan 18 merupakan tampilan perbandingan rute dari titik pengguna menuju Rumah Sakit Ibu dan Anak Kirana dan RS Siti Khodijah Muhammadiyah. Dengan keterangan mulai dari kiri *GMaps*, Aplikasi SIG, dan *Bing Maps*.



Gbr 17. Perbandingan rute menuju Rumah Sakit ibu dan Anak Kirana



Gbr 18. Perbandingan rute menuju RS Siti Khodijah Muhammadiyah

Dari contoh perbandingan yang ada pada gambar 17 dan 18 dapat dilihat bahwa perbedaan jarak yang terjadi akibat dari pemilihan rute yang berbeda dari aplikasi SIG dengan Gmaps maupun Bing Maps. Akurasi dapat lebih ditingkatkan dengan pembaruan data jalan yang digunakan sebagai dasar dari penentuan rute.

*Percobaan Pada Titik 2*

Percobaan kedua dilakukan pada titik 2 dengan koordinat Latitude, Longitude (-7.452847646127449, 112.72069391336308) yang beralamat di Jl. Kartini 103-115, Gabahan, Sidokumpul, Kec. Sidoarjo, 61218. Hasil dari percobaan pada titik 2 dapat dilihat pada tabel 5 untuk pengujian pada *Google Maps* dan table 6 untuk pengujian pada *Bing Maps*. Perbandingan jarak secara grafik ditunjukkan pada gambar 19.

TABEL 5  
HASIL PENGUJIAN AKURASI PADA TITIK 2 GOOGLE MAPS

No	Nama Faskes	Aplikasi SIG	Google Maps	
			Jarak	Akurasi
1	Dr. H. SUHARTONO, Sp.PD - Spesialis Penyakit Dalam	0,7 km	0,7 km	100 %
2	Dr. DONI PRIYO W. SpPD - Spesialis Penyakit Dalam	0,495 km	0,5 km	99 %
3	RSI Siti Hajar Sidoarjo	0,797 km	0,8 km	99,6 %
4	Dr.Hari Tjahjono Sp OT	1,066 km	1,2 km	88,3 %
5	Rumah Sakit Umum Jasem	1,18 km	1,4 km	84,2 %
6	dr. Yunanto Yosediputra, Sp.OG (Spesialis kebidanan & Kandungan)	1,314 km	1,1 km	83,7 %

7	Klinik Al Akbar	1,822 km	1,2 km	65 %
---	-----------------	----------	--------	------

TABEL 6  
HASIL PENGUJIAN AKURASI PADA TITIK 2 BING MAPS

No	Nama Faskes	Aplikasi SIG	Bing Maps	
			Jarak	Akurasi
1	Dr. H. Suhartono, Sp.Pd	0,7 km	-	-
2	Dr. Doni Priyo W. Sppd	0,495 km	-	-
3	Rsi Siti Hajar Sidoarjo	0,797 km	0,7 km	90,9 %
4	Dr.Hari Tjahjono Sp Ot	1,066 km	-	-
5	Rumah Sakit Umum Jasem	1,18 km	1,4 km	84,2 %
6	Dr. Yunannto Yosediputra, Sp.Og	1,314 km	-	-
7	Klinik Al Akbar	1,822 km	1,2 km	65 %



Gbr 21. Perbandingan rute menuju Klinik Al Akbar

Pada gambar 20 dari 3 aplikasi memiliki titik lokasi tujuan yang sama dan pemilihan rute yang sangat mirip, namun pada gambar 21 titik lokasi tujuan yang ada pada Bing Maps, hal ini tentu mempengaruhi pemilihan rute dan perbandingan jarak.

*Percobaan Pada Titik 3*

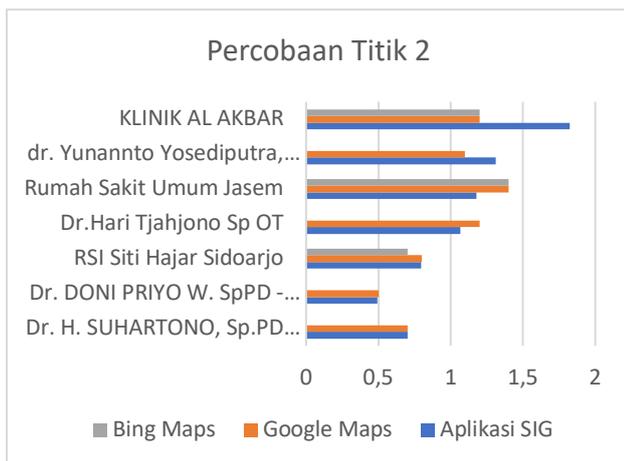
Percobaan ketiga dilakukan pada titik 3 dengan koordinat Latitude, Longitude (-7.354081406641987, 112.74789337919418) yang beralamat di Belahan, Wedoro, Kec. Waru, 61256. Hasil dari percobaan pada titik 3 dapat dilihat pada tabel 7 untuk pengujian pada Google Maps dan table 8 untuk pengujian pada Bing Maps. Perbandingan jarak secara grafik ditunjukkan pada gambar 22.

TABEL 7  
HASIL PENGUJIAN AKURASI PADA TITIK 3 GOOGLE MAPS

No	Nama Faskes	Aplikasi SIG	Google Maps	
			Jarak	Akurasi
1	Klinik Keluarga Sehat Wedoro	0,244 km	0,24 km	98,3 %
2	Klinik Graha Medika	0,581 km	0,5 km	86 %
3	Klinik geo medika	1,131 km	1 km	88,4 %
4	Klinik Hidayah Waru IV	1,689 km	1,9 km	88,8 %
5	Citra Gading Medika	1,296 km	1,4 km	92,6 %
6	Puskesmas Waru	1,53 km	2 km	76,5 %

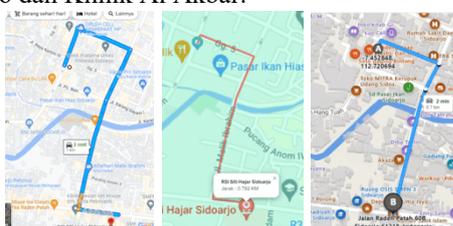
TABEL 8  
HASIL PENGUJIAN AKURASI PADA TITIK 3 BING MAPS

No	Nama Faskes	Aplikasi SIG	Bing Maps	
			Jarak	Akurasi
1	Klinik Keluarga Sehat Wedoro	0,244 km	-	-
2	Klinik Graha Medika	0,581 km	1,8 km	32,7 %



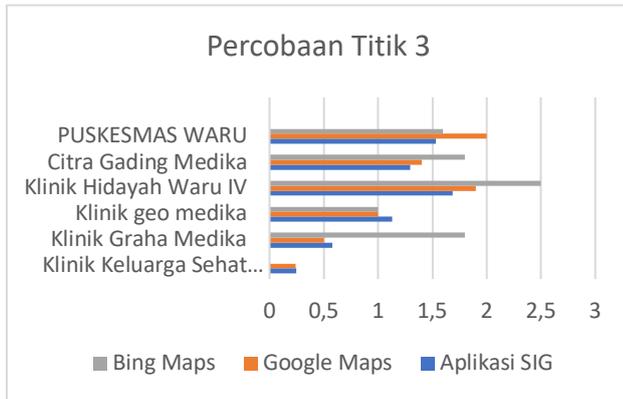
Gbr 19. Grafik Perbandingan Percobaan Titik 2

Berikut pada gambar 20 dan 21 merupakan tampilan perbandingan rute dari titik pengguna menuju RSI Siti Hajar Sidoarjo dan Klinik Al Akbar.



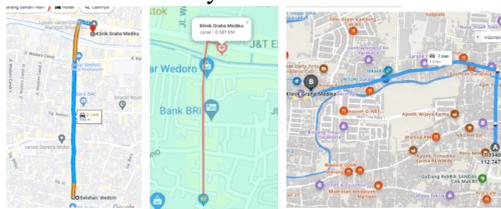
Gbr 20. Perbandingan rute menuju RSI Siti Hajar Sidoarjo

3	Klinik geo medika	1,131 km	1 km	88,4 %
4	Klinik Hidayah Waru IV	1,689 km	2,5 km	67,5 %
5	Citra Gading Medika	1,296 km	1,8 km	72 %
6	Puskesmas Waru	1,53 km	1,6 km	95,6 %

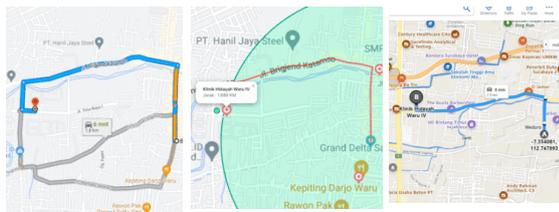


Gbr 22. Grafik Perbandingan Percobaan Titik 3

Berikut pada gambar 23 dan 24 merupakan tampilan perbandingan rute dari titik pengguna menuju Klinik Graha Medika dan Klinik Hidayah Waru IV.



Gbr 23. Pemilihan rute menuju Klinik Graha Medika



Gbr 24. Pemilihan rute menuju Klinik Hidayah Waru IV

Dapat dilihat pada gambar 23 dan 24, Aplikasi SIG mampu memberikan rute yang hampir sama bahkan lebih baik daripada *G Maps*, namun pada *Bing Maps* titik faskes belum diperbarui dengan baik sehingga menuju ke titik lain.

Pada tiga pengujian diatas didapatkan hasil akurasi rata-rata sebagai berikut :

- Percobaan 1: *Google Maps* 80,3% dan *Bing Maps* 85,3 %,
- Percobaan 2: *Google Maps* 88,5% dan *Bing Maps* 80,6 %,

- Percobaan 3: *Google Maps* 88,38% dan *Bing Maps* 59,3 %.

Dari hasil di atas dapat ditarik rata-rata untuk akurasi pada *Google Maps* 85,72 % dan *Bing Maps* 75 %. Pada hasil di atas juga dapat diketahui bahwa data Faskes yang ada pada *Bing Maps* belum lengkap untuk wilayah Sidoarjo, sehingga pada beberapa Faskes menunjukkan titik yang sangat berbeda dan mempengaruhi perbandingan jarak tempuh rute. Sedangkan pada *GMaps* memiliki hasil perbandingan jarak tempuh dan rute yang cukup baik.

Pada beberapa titik Faskes untuk rute yang ditentukan oleh aplikasi pada penelitian ini mampu memberikan jarak yang lebih pendek daripada *Google Maps* dan *Bing Maps*. Namun pada beberapa titik Faskes aplikasi ini juga menunjukkan rute yang terlalu jauh, hal ini dipengaruhi oleh data jalan yang didapatkan masih belum diperbarui sesuai dengan jalan saat ini dan data *traffict* yang didapatkan belum *real-time*. Tentunya hasil rute dan jarak tempuh yang didapatkan tidak selamanya sama karena perubahan yang dinamis. Algoritma ini menjadi salah satu yang terbaik untuk penentuan jarak terpendek dengan dinamis. Dengan pembaruan data jalan dan lalu lintas yang lebih baik algoritma ini bisa memberikan hasil yang lebih optimal.

#### IV. KESIMPULAN

Setelah dilakukan percobaan dan pengujian pada penelitian ini dapat diperoleh kesimpulan sebagai berikut :

1. Penelitian ini dapat diterapkan untuk menentukan rute terdekat menuju Faskes yang ada di Sidoarjo. Dengan data Faskes dan data jalan yang ada, dapat diolah untuk menentukan rute menggunakan algoritma *Dynamic Dijkstra* yang diterapkan dalam sebuah aplikasi Sistem Informasi Geografis. Selain menentukan rute, algoritma ini juga bisa digunakan untuk memberi rekomendasi Faskes terdekat.
2. Penerapan *Dynamic Dijkstra* membutuhkan titik-titik Faskes dan jalan yang berupa *Latitude* dan *Longitude* yang diolah oleh untuk menentuka rute maupun jarak menuju Faskes. Peningkatan algoritma *Dijkstra* yang awalnya statis menjadi dinamis membutuhkan data pembaruan secara berkala ketika algoritma dieksekusi. Karena hal itu algoritma ini membutuhkan pembaruan bobot jalan secara terus menerus saat dieksekusi, maka perlu dibuat batasan kapan waktu eksekusi yang dibutuhkan dan berapa banyak maksimal data pembaruan bobot yang digunakan untuk mengurangi beban kinerja algoritma ini. Data dinamis pada aplikasi yang dibuat dalam penelitian ini belum benar-benar dinamis sesuai dengan kehidupan nyata, dinamisasinya dibuat menggunakan simulasi internal sistem dengan tujuan untuk mengetahui kinerja dan kemampuan algoritma *Dynamic Dijkstra*. Diharapkan kedepannya sudah ada data lalu lintas dinamis yang dapat diterapkan.
3. Hasil pengujian menyimpulkan bahwa aplikasi SIG yang memakai algoritma *Dynamic Dijkstra* dapat bekerja dengan baik. Hal ini dapat dilihat dari pengujian

Alpha dengan Teknik *blackbox* yang menghasilkan aplikasi ini dapat bekerja sesuai fungsinya dengan baik serta pada pengujian akuasi melalui 3 percobaan dalam beberapa titik Faskes aplikasi ini dapat memberikan jarak yang lebih pendek daripada *Google Maps* dan *Bing Maps*, akurasi rata-rata perbandingan jarak jika dibandingkan dengan *Google Maps* 85,72 % sedangkan pada *Bing Maps* 75 %. Jika dibandingkan dengan *Bing Maps*, aplikasi SIG memiliki data Faskes lebih lengkap dan akurasi yang lebih mumpuni. Namun jika dibandingkan dengan *Google Maps*, aplikasi SIG ini membutuhkan beberapa parameter lagi untuk penyesuaian bobot jalan sehingga bisa lebih baik.

#### V. SARAN

Penelitian ini dapat dikembangkan lagi, baik dari sisi algoritma maupun dari sisi peningkatan penentuan rute. Dari sisi algoritma diharapkan kedepannya terdapat metode ataupun pembaruan yang diterapkan pada *Dynamic Dijkstra* untuk mengurangi beban kerja yang dilakukan saat algoritma dieksekusi. Kemudian dari sisi peningkatan penentuan rute, diharapkan sudah ada data jalan dan lalu lintas yang dapat digunakan secara *real-time* oleh aplikasi ini tanpa menggunakan biaya yang terlalu besar.

#### UCAPAN TERIMA KASIH

Terimakasih dan syukur penulis panjatkan kepada Allah SWT. Atas rahmat, hidayah dan ridhonya penulis dapat menyelesaikan penelitian ini. Penulis juga berharap penelitian ini bermanfaat bagi khalayak ramai. Selain itu tak lupa penulis juga ucapkan terimakasih kepada :

1. Ibu Umi Abiba selaku ibu yang selalu mendukung kebutuhan dan perjalanan hidup penulis hingga sampai pada titik ini.
2. Bapak I Kadek Dwi Nuryana selaku dosen pembimbing yang memberikan bimbingan dan masukan yang berkaitan dengan keberlangsungan penelitian ini.
3. Keluarga dan seluruh rekan penulis yang memberikan dukungan, bantuan serta semangat untuk menyelesaikan penelitian ini.

#### REFERENSI

- [1] Sagita N. K. K, "70 Persen Kematian Pasien Emergency Terjadi dalam Perjalanan ke RS," <https://health.detik.com/berita-detikhealth/d-4808814/70-persen-kematian-pasien-emergency-terjadi-dalam-perjalanan-ke-rs>, 2019. <https://health.detik.com/berita-detikhealth/d-4808814/70-persen-kematian-pasien-emergency-terjadi-dalam-perjalanan-ke-rs> (accessed Jun. 24, 2022).
- [2] D. Kadhim Muhsen and R. Flaih Hassan, "Spatial Decision Support System to Find the Nearest Hospital in Baghdad By using GIS and GPS." [Online]. Available: <https://sites.google.com/site/ijcsis/>
- [3] H. Ali, A. Alasadi, M. Talib Aziz, M. Dhiya, and A. Abdulmajed, "A Network Analysis for Finding the Shortest Path in Hospital Information System with GIS and GPS", doi: 10.23977/jnca.2020.050103.
- [4] Sunita and D. Garg, "Dynamizing Dijkstra: A solution to dynamic shortest path problem through retroactive priority queue," *Journal of King Saud University - Computer and Information Sciences*, vol. 33, no. 3, pp. 364–373, Mar. 2021, doi: 10.1016/j.jksuci.2018.03.003.
- [5] 2019 *IEEE 9th International Conference on Electronics Information and Emergency Communication (ICEIEC)*. IEEE.
- [6] N. Nurwatak, J. H. Hong, L. M. Jaelani, H. H. Handayani, A. B. Cahyono, and M. R. Darminto, "Using GIS to Understand Healthcare Access Variations in Flood Situation in Surabaya," *ISPRS International Journal of Geo-Information*, vol. 11, no. 4, Apr. 2022, doi: 10.3390/ijgi11040235.
- [7] S. Liu, H. Jiang, S. Chen, J. Ye, R. He, and Z. Sun, "Integrating Dijkstra's algorithm into deep inverse reinforcement learning for food delivery route planning," *Transportation Research Part E: Logistics and Transportation Review*, vol. 142, Oct. 2020, doi: 10.1016/j.tre.2020.102070.
- [8] A. Febriano, "Download Shapefile RBI Provinsi Jawa Timur Perwilayah (Kabupaten/Kota)," 2020. <https://www.indonesia-geospasial.com/2020/01/shp-rbi-provinsi-jawa-timur-perwilayah.html> (accessed Jun. 06, 2022).
- [9] Bakosurtanal, "Geospasial untuk negeri." <https://tanahair.indonesia.go.id/portal-web> (accessed Jun. 06, 2022).
- [10] V. Brestoiu, "Importing Data from Shapefiles and Pathfinding along Generated Nodes," *Journal of Student Science and Technology*, vol. 10, no. 1, Aug. 2017, doi: 10.13034/jsst.v10i1.126.
- [11] J. W. de Andrade and R. D. Seabra, "Fully retroactive priority queues using persistent binary search trees," *Journal of Computer Science*, vol. 16, no. 7, pp. 906–915, 2020, doi: 10.3844/JCSSP.2020.906.915.
- [12] E. D. Demaine, "Retroactive Data Structures." [Online]. Available: <http://john.poly.edu>.
- [13] U. A. Acar, G. E. Brelloch, and K. Tangwongsan, "Non-oblivious Retroactive Data Structures," 2007.
- [14] Humphrey, "Dijkstra's Algorithm in JavaScript," Jan. 2022. <https://www.codeunderscored.com/dijkstras-algorithm-in-javascript/> (accessed Jun. 10, 2022).
- [15] A. Rosano, "Pengujian Alpha dan Beta pada Pengembangan Sistem Internet Banking (Ibank) PT Bank Mega, Tbk," *Riset dan E-Jurnal Manajemen Informatika Komputer*, vol. 3, no. 2, 2019.
- [16] Y. Dwi Wijaya and M. Wardah Astuti, "Pengujian Blackbox Sistem Informasi Penilaian Kinerja Karyawan Pt Inka (Persero) Berbasis Equivalence Partitions Blackbox Testing Of Pt Inka (Persero) Employee Performance Assessment Information System Based On Equivalence Partitions," *Jurnal Digital Teknologi Informasi*, vol. 4, p. 2021.
- [17] BPS Sidoarjo, "Penduduk Menurut Kecamatan dan Jenis Kelamin (Jiwa), 2018-2020." <https://sidoarjokab.bps.go.id/indicator/12/59/1/penduduk-menurut-kecamatan-dan-jenis-kelamin.html> (accessed Jun. 18, 2022).