

Review Artikel: Analisis Penggunaan Algoritma Dijkstra untuk Mencari Rute Terpendek di Rumah Sakit

Lailatul Musabbikhah¹, Risky Via Yuliantari^{2*}

^{1,2}Program Studi Teknik Elektro Fakultas Teknik Universitas Tidar
Jalan Kapten Suparman 39 Magelang, Jawa Tengah 56116 Indonesia

Lailatul.musabbikhah@students.untidar.ac.id¹, rviay@untidar.ac.id²

*Corresponding Author

Abstrak— Rumah sakit merupakan institusi yang melayani bidang kesehatan dan banyak dikunjungi masyarakat setiap harinya. Bangunan rumah sakit yang besar dan memiliki banyak ruang dan lorong seringkali menyulitkan pengunjung untuk menemukan suatu ruangan. Sedangkan *Google Maps*, aplikasi yang kerap digunakan untuk menunjukkan rute ke sebuah tempat hanya dapat digunakan di luar ruangan. Sehingga dibutuhkan sebuah aplikasi yang dapat menunjukkan rute menuju suatu ruangan di rumah sakit. Analisis pencari rute terpendek dengan algoritma Dijkstra untuk di dalam ruangan dilakukan dengan membandingkan hasil penelitian terdahulu tentang pencarian rute terpendek. Pembuatan aplikasi pencari rute terpendek dengan algoritma Dijkstra sudah banyak diteliti. Hasilnya adalah aplikasi berbasis web yang mengambil data dari *Google Maps* untuk penentuan letak titik awal dan tujuan kemudian pemanfaatan algoritma untuk penentuan rute terdekat. Dalam pengujian algoritma Dijkstra dan Astar menghasilkan rute yang sama dengan waktu eksekusi yang berbeda. Review artikel ini dibuat untuk mengetahui cocok tidaknya penggunaan algoritma Dijkstra untuk mencari rute terpendek di dalam bangunan khususnya rumah sakit.

Kata kunci—rumah sakit, rute terpendek, aplikasi, algoritma Dijkstra

Abstract— *The hospital is an institution that serves the health sector and is visited by many people everyday. Hospital building are large and have lots of rooms and passages often make it difficult for visitors to find a room. Meanwhile, Google Maps, an application that is often use to show a route to a place can only be used outdoors. So we need an application that can show the route to a room in the hospital. The analysis of finding the shortest route using Dijkstra's algorithm for indoors was carried out by comparing the result of previous research on the search of the shortest route. The making of the application for finding the shortest route with Dijkstra algorithm has been widely researched. The result is a web-based application that retrieves data from Google Maps for determining the location of the starting and destination points and then utilized algorithms for determining the closest route. In testing Dijkstra and Astar algorithms produce the same route with different execution times. This article review was made to find out wheather or not the use of Dijkstra's algorithm is suitable to find the shortest route in a building especially in hospital.*

Keywords— hospital, shortest route, application, Dijkstra algorithm

22ASW23V—————GB

I. PENDAHULUAN

Ketergantungan masyarakat dengan *Google Maps*, aplikasi *mobile* yang biasa digunakan untuk menunjukkan letak suatu objek maupun mencari rute menuju suatu tempat cukup tinggi. Untuk menuju ke suatu kota misalnya, kebanyakan orang lebih mengandalkan *Google Maps* daripada plang penunjuk jalan. Menampilkan lokasi *realtime* pengguna, memilih rute menuju suatu tempat, memberi peringatan seperti jalan tergenang banjir merupakan kelebihan yang menjadi alasan aplikasi ini banyak diminati. Aplikasi pencari rute seperti ini sangat berguna baik digunakan di luar ruangan maupun di dalam gedung.

Pemanfaatan *Google Maps* saat ini masih terbatas pada penggunaan di luar ruangan, baik untuk mencari suatu objek maupun menunjukkan rute menuju objek tersebut. Padahal kebutuhan adanya aplikasi semacam ini sama pentingnya untuk digunakan di dalam gedung terutama yang memiliki kompleksitas tinggi seperti rumah sakit.

Rumah sakit merupakan sebuah institusi yang melayani kesehatan perorangan secara keseluruhan dengan penyediaan layanan rawat inap, rawat jalan dan juga layanan gawat darurat (Putra et al., 2020). Setiap harinya banyak orang yang keluar masuk rumah sakit baik orang sakit yang dirawat, keluarga dan kerabat yang mengunjungi, maupun orang-orang yang melakukan *medical check up*. Gedung rumah sakit

biasanya cukup besar dan memiliki banyak ruang dan lorong. Tanpa adanya petunjuk yang jelas mengenai ruang dan rute untuk menuju kesana, orang-orang akan merasa kesulitan. Oleh karena itu diperlukan aplikasi pencari rute dan penunjuk jalan untuk digunakan di dalam bangunan.

Algoritma Dijkstra banyak diteliti untuk diaplikasikan pada sistem pencarian rute terpendek. Algoritma ini ditemukan oleh Edsger Dijkstra, seorang ilmuwan komputer asal Belanda. Cara kerja algoritma Dijkstra adalah dengan strategi *greedy* yaitu pada setiap langkahnya memilih sisi dengan nilai terkecil yang menghubungkan simpul/node yang telah terpilih dan yang belum terpilih. Algoritma ini membutuhkan titik asal dan tujuan dengan hasil akhir jarak terpendek dari titik asal ke tujuan beserta rutenya (Sianturi, 2018). Algoritma Dijkstra merupakan salah satu algoritma yang biasa digunakan dalam menentukan jalur terpendek pada sebuah graf berarah (Dewi, 2010). Ada banyak jenis graf yang dikenal, namun dalam algoritma Dijkstra yang dipakai adalah graf berbobot dan berarah dimana setiap sisinya diberi orientasi arah dan memiliki bobot dengan nilai yang berbeda-beda. Dalam algoritma ini ruangan direpresentasikan dengan node dan jalur yang dapat ditempuh sebagai bobot. Secara sederhana, algoritma ini akan memilih jalur yang menghubungkan node satu dengan lainnya dengan nilai bobot yang paling kecil. Kemudian, jalur dan sudah dipilih akan dihilangkan dari pilihan tahap ke dua sehingga tidak akan memunculkan rute yang berbolak-balik antara satu titik dengan titik lain.

Melalui analisa terhadap penelitian mengenai penggunaan algoritma Dijkstra untuk mencari rute terpendek diharapkan dapat diidentifikasi cocok tidaknya penggunaan algoritma ini untuk menunjukkan rute terpendek untuk digunakan di dalam ruangan, khususnya di rumah sakit.

II. METODE PENELITIAN

Teknik yang digunakan dalam penyusunan artikel ini, adalah studi pustaka dengan mencari jurnal nasional sebagai rujukan utama dengan topik yang sama yaitu seputar penggunaan algoritma Dijkstra dalam mencari rute terdekat dan pengumpulan data sekunder melalui situs online seperti Google, *Google Maps*, dan situs jurnal untuk mendukung keakuratan analisa yang dibuat. Artikel-artikel tersebut kemudian dianalisa mulai dari metode yang digunakan sampai hasilnya untuk melihat apakah algoritma Dijkstra cocok dan sesuai untuk digunakan pada aplikasi pencari rute terdekat di dalam gedung khususnya rumah sakit.

Ada beberapa metode yang dapat digunakan dalam perancangan aplikasi penentu rute terdekat. Metode yang digunakan dalam penentuan rute dan perpindahan koridor dalam aplikasi pencarian rute bus Trans Semarang adalah dengan menggunakan metode pengembangan perangkat lunak model Waterfall yang meliputi tahapan pendefinisian kebutuhan, desain sistem dan perangkat lunak, implementasi dan pengujian unit, integrasi dan pengujian sistem, serta operasional dan perawatan. Aplikasi ini menggunakan pendekatan I dengan algoritma pemrograman berorientasi pada objek atau biasa disebut *Object Oriented Programming* (OOP) dan juga bahasa pemrograman UML yang biasa digunakan untuk penulisan blueprint perangkat lunak. Dalam

penentuan jarak antar titik berdasarkan letak lintang dan bujunya digunakan metode Harvesine. Metode Harvesine merupakan sebuah formula untuk menghitung jarak lokasi di bumi dengan memperhatikan garis lintang dan garis bujur sebagai masukannya. Data spasial diambil dari *Google Maps API* dan data non spasial diambil dari informasi detail *shelter* dan koridor Calon penumpang dapat memasukkan lokasi yang dituju untuk mendapatkan rute (Ardana and Saputra, 2016).

Untuk mencari rute terpendek di Bali dengan menggunakan Algoritma Dijkstra adalah dengan membangun web berbasis sistem informasi geografis (SIG) dengan menggunakan skrip PHP dan MySQL. Gambar yang dihasilkan memiliki format file *Scalable Vector Graphics* (SVG) yang dapat menampilkan grafik dua dimensi pada web yang berbasis *eXtension Markup Language* (XML). Dalam pengembangan sistem, langkah pertama yang dilakukan adalah dengan melakukan analisis kebutuhan dengan metode kepastakaan, kemudian pengembangan sistem dengan diagram *use case* yang terdiri dari mencari rute terpendek dan memperbesar peta. *Database* yang digunakan berupa data tempat wisata yang ada di Bali beserta jarak tempuhnya (Dewi, 2010).

Dalam perancangan aplikasi penentuan rute rumah sakit digunakan algoritma Dijkstra dengan model UML yang dibangun berbasis web dengan script PHP dan *database* MySQL serta editor yang digunakan adalah Dreamweaver CS3. Tahap pertama yang dilakukan adalah studi literatur untuk mencari referensi baik dari buku maupun jurnal, kemudian dilakukan pengumpulan data dengan melakukan wawancara ke Dinas Kesehatan Kota Kisaran dan observasi secara langsung untuk menentukan data letak geografis rumah sakit. Langkah selanjutnya yang dilakukan adalah melakukan analisis sistem yang dibutuhkan kemudian dilakukan perancangan sistem dengan menggunakan model UML, perancangan *user interface*, perancangan *input* dan *output* dan langkah terakhir adalah penulisan kode program dan pengujian. Pengujian sistem digunakan dengan *blackbox testing* dengan tujuan untuk menemukan kesalahan pada sistem agar dapat diperbaiki. Parameter yang digunakan adalah jarak obyek dengan rumah sakit tujuan. Lokasi obyek dan rumah sakit yang dituju berperan sebagai masukan dengan rute menuju rumah sakit terdekat sebagai keluaran (Ginting and Barus, 2018).

Analisis perbandingan penggunaan algoritma Dijkstra dan Astar dalam penentuan rute terpendek pada studi kasus SPBU di Kota Medan adalah dengan membangun aplikasi menggunakan bahasa pemrograman C# dan SharpDevelop 5.1 sebagai perangkat bantu. Terdapat halaman *shortest path* yang digunakan pengguna untuk menentukan rute terpendek. Pengguna juga dapat memilih sendiri algoritma apa yang akan digunakan. Pengujian dilakukan dengan menghitung nilai heuristik secara manual menggunakan metode Euclidean Distance dan mencatat jarak sebenarnya antar SPBU. Bobot jarak yang digunakan diambil dari data yang tersedia dalam *Google Maps*. Adapun langkah-langkah yang dilakukan dalam perancangan sistem ini yang pertama adalah studi pustaka untuk memberikan gambaran mengenai sistem yang akan dibangun, analisis sistem untuk menentukan kebutuhan pengembangan sistem, dan perancangan sistem yang berupa pembuatan diagram Ishikawa, diagram umum sistem, diagram

use case, diagram aktivitas, diagram pewaktu, diagram alir serta perancangan tampilan aplikasi. Langkah selanjutnya adalah implementasi sistem yaitu pembuatan program, kemudian pengujian sistem untuk memastikan program dapat berjalan sesuai yang diharapkan dan yang terakhir adalah dokumentasi sistem untuk menunjukkan hasil dalam penelitian (Fisika et al., 2016).

Dalam membandingkan antara kinerja algoritma Dijkstra dan Astar untuk pencarian rute terpendek pemetaan pariwisata Kota Sawahlunto, algoritma dibuat menggunakan MS4W yang terdiri dari server Apache, pHP, MapServer CGI, pHP/Mapscript, pustaka GDAL & OGR, program utility Mapserver dengan basis data PostgreSQL dan dukungan PostGIS dan pgRouting. Aplikasi tersebut menggunakan Framework Chameleon dengan data yang dibuat di ArcView kemudian dikonversi menggunakan bantuan aplikasi shp2pgsql dari PostGIS yang melalui *command prompt*. Sistem dibuat melalui proses perancangan DAD level konteks untuk melihat hubungan antara pengunjung, administrator, dan website SIG kemudian dibuatlah perancangan basis data, dan juga perancangan antar muka. Adapun komputer yang digunakan memiliki spesifikasi Prosesor Intel Pentium Dual-Core 1.9 GHz, harddisk 120 GB, dan RAM 1 GB (Pugas et al., 2011).

Metode dalam perancangan aplikasi berbasis sistem informasi geografis (SIG) dalam penentuan rute terdekat fasilitas umum adalah dengan menggunakan algoritma Dijkstra dan R-Tree menggunakan sistem yang dirancang dengan *Unified Modeling Language* (UML) yang dapat menggambarkan sistem perangkat lunak terkait dengan objek, dan bantuan GPS untuk menentukan lokasi. Aplikasi ini dibangun dengan spesifikasi *software* Windows 8 dan Basic4Android dengan Xiaomi Redmi 3X 4G Intel Core i3 2.27GHz dan kapasitas harddisk 250GB RAM 2GB. Tahap pertama dalam pembuatan sistem ini adalah dengan melakukan studi literature untuk mendapatkan informasi terkait algoritma Dijkstra dan Astar baik dari buku, jurnal, skripsi, maupun website. Tahap selanjutnya adalah melakukan analisis dan perancangan sistem berdasarkan metode dan *tools* yang akan digunakan. Setelah itu adalah tahap implementasi yaitu melakukan pengkodean dan membangun aplikasi sesuai analisis dan perancangan sebelumnya. Kemudian dilakukan pengujian untuk memastikan sistem bekerja dengan baik sesuai dengan tujuan dibuatnya. Langkah terakhir merupakan dokumentasi dan juga penyusunan laporan akhir (Sianturi, 2018).

III. HASIL DAN PEMBAHASAN

Hasil penelitian menunjukkan bahwa aplikasi pencarian rute bus Trans Semarang menghasilkan aplikasi yang dapat menampilkan rute perjalanan bus serta *shelter* terdekat dan juga perpindahan koridor pada titik transfer. Pengguna dapat memilih menu koridor, lokasi *shelter* dan jalur pada halaman utama aplikasi. Masukan pada algoritma Dijkstra berupa *node* awal dan *node* tujuan dan kemudian diproses dengan menghitung nilai fungsi dan nilai heuristic pada *node* yang memungkinkan untuk dilalui sehingga didapatlah suatu rute yang ditampilkan dalam peta. Logika yang terdapat dalam sistem menggunakan *use case diagram* yang memiliki 6

langkah, yaitu *login*, pengelolaan *shelter*, pengelolaan jalur, melihat rute, melihat *shelter*, melihat jalur, dan pengelolaan koridor. Selain itu, pemeran dalam perancangan diagram ada dua yaitu admin dan pengguna umum. Pengujian dilakukan dengan memilih titik awal baik secara manual dengan memasukkan nama tempat maupun mengeklik titik pada peta yang tersedia. Pencarian rute dimulai setelah ditemukan *shelter* terdekat kemudian penentuan *transfer point* selanjutnya apakah masih dalam satu koridor yang sama atau tidak. Transfer point dapat digunakan untuk beralih koridor. Aplikasi yang dibangun memiliki 6 menu, yaitu *home*, koridor, *shelter*, lokasi *shelter*, jalur, dan peta. Data yang ditampilkan dalam aplikasi bersifat dinamis sesuai kondisi di lapangan dengan manajemen data yang dilakukan oleh admin sistem (Ardana and Saputra, 2016).

Penelitian pengembangan sistem pencarian rute terpendek tempat wisata di Bali menghasilkan website dengan tampilan yang *user-friendly* dengan *database* yang berisi tempat wisata beserta jaraknya. Uji coba program dilakukan sebanyak 5 kali yang mencakup 3 kali pengujian rute terpendek dengan memasukkan lokasi asal dan tujuan, pengujian perbesaran dengan mengklik ikon perbesaran pada navigasi, dan pengujian pergeseran peta dengan mengklik ikon panning di sekitar peta. Pengujian algoritma Dijkstra yang dilakukan dengan memilih titik asal Tanah Lot dan titik tujuan Sanur menghasilkan jarak terpendek 33.33 km dengan banyak titik yang dilalui sejumlah 9. Kualitas gambar yang dihasilkan cukup baik karena menggunakan format file SVG dan mudah dalam akses serta memiliki jangkauan yang luas. Akan tetapi sistem yang dibangun ini belum dapat mengakomodir lebih dari satu titik tujuan dan tampilan masih sederhana (Dewi, 2010).

Algoritma Dijkstra digunakan untuk menentukan jalur terpendek diantara dua titik berdasarkan bobot terkecilnya. Data rumah sakit dikumpulkan dengan proses wawancara ke Dinas Kesehatan Kota Kisaran dan juga pengamatan secara langsung. Proses yang terjadi pada pemilihan rute adalah pengguna memulai masuk ke menu, memilih rumah sakit tujuan dan sistem akan memproses untuk menentukan rute terdekat yang dapat dipilih. Sistem yang dihasilkan akan mengarahkan pengguna untuk masuk ke menu *project* kemudian memilih rumah sakit tujuan dengan meletakkan ikon *user* pada peta. Pengguna harus mengklik tombol cari untuk mendapatkan rute. Sementara itu, sistem akan memproses pengambilan rute pada *database* untuk ditampilkan pada program. Apabila pengguna ingin mengganti rumah sakit tujuan caranya sama yaitu meletakkan ikon *user* pada peta dan klik tombol cari. Setelah itu, pengguna dapat langsung mengikuti rute yang telah ditampilkan pada program. Diagram *use case* yang digunakan untuk memodelkan kelakuan sistem informasi terdiri dari pengolahan menu, pengolahan petunjuk, validasi dan tampilan peta dengan dua pemeran yaitu admin yang bertugas mengolah data dan memiliki akses sepenuhnya pada aplikasi dan pengguna yang memanfaatkan sistem aplikasi tersebut. Aplikasi terdiri dari halaman utama website, halaman *login* admin, halaman admin, halaman data admin, dan halaman data rumah sakit. Pengujian dilakukan untuk menjamin kualitas sistem dan mengetahui kelemahan perangkat lunak untuk dapat diperbaiki sebelum peluncuran. Kriteria pengujian

dimaksudkan untuk memeriksa kebenaran dan kesesuaian perangkat lunak berdasarkan spesifikasinya. Sistem dinyatakan berhasil jika sesuai dengan spesifikasi yang telah ditentukan sebelumnya. Sistem yang dibangun sudah sesuai dengan tujuan awal dibuat sehingga dapat membantu mobilitas sehari-hari dan juga memiliki pembatasan akses untuk meningkatkan kontrol internal agar pihak-pihak yang tidak berwenang tidak leluasa mengakses data krusial (Ginting and Barus, 2018).

Penelitian untuk membandingkan Algoritma Dijkstra dan Astar dalam penentuan rute terpendek SPBU di Kota Medan dirancang dengan menggunakan *use-case diagram* untuk menggambarkan interaksi dengan sudut pandang pengguna, *activity diagram* untuk menggambarkan aktivitas sistem, dan *sequence diagram* untuk menggambarkan interaksi objek yangurut berdasarkan waktu. Pada aplikasi ini pengguna dapat mencari rute menuju SPBU terdekat dengan data SPBU asal dan tujuan yang berada dalam daftar 21 SPBU dalam sistem kemudian pengguna memilih algoritma apa yang akan digunakan. Sistem akan menampilkan total jarak, rute, dan waktu eksekusi program. Aplikasi ini berisi halaman awal untuk memulai masuk sistem dan halaman *shortest path* untuk memasukkan data *input* dan juga menampilkan *output*. Pengujian sistem dilakukan dengan menggunakan 3 PC yang berbeda dengan titik asal dan tujuan sebanyak 24 titik. Sistem menyediakan 2 pilihan algoritma yaitu Dijkstra dan Astar. Pada pengujian algoritma Dijkstra dari SPBU Gatot Subroto ke SPBU Ringroad menampilkan rute dengan jarak total 5.4 km dan waktu eksekusi 4.9956 ms pada PC_1, 5.3576 ms pada PC_2, 11.7469 ms pada PC_3. Pada pengujian dengan algoritma Astar dengan titik awal dan tujuan yang sama menghasilkan rute dengan jarak total 5.5 km dengan waktu eksekusi 1.7723 ms pada PC_1, 2.1741 ms pada PC_2, dan 2.4534 ms pada PC_3. Selain itu, pengujian juga dilakukan dengan penghitungan manual menggunakan metode heuristik dan menghasilkan jarak total 5.4 km. pemilihan *hardware* yang digunakan tidak mempengaruhi total jarak dan sangat mempengaruhi waktu eksekusi. Aplikasi yang dibuat masih secara statis sehingga diharapkan untuk kedepannya dibuat aplikasi yang dinamis untuk mengantisipasi apabila ada pembaruan data dan juga diharapkan kedepannya data peta dapat diambil dari *Google Maps* untuk memberikan bentuk jalur yang lebih baik (Fisika et al., 2016).

Sistem informasi yang dibuat dapat menampilkan visualisasi peta wisata Kota Sawahlunto dan menemukan rute terpendek antar obyek wisata. Aplikasi terdiri dari halaman *login*, halaman administrator, dan halaman pengunjung. Sebelum masuk ke halaman administrator, pengunjung diharuskan untuk memasukkan *username* dan *password*. Tampilan yang dihasilkan berupa tabel lokasi pengguna dan tujuan serta jalan mana saja yang harus dilalui. Selain itu, tampilan juga dapat diubah ke bentuk peta dengan menampilkan rute terdekat yang dapat dipilih. Pengujian halaman administrator dilakukan dengan memasukkan data spasial dan non spasial kemudian memperhatikan bagaimana data dimasukkan dalam *storage*. Pengujian halaman pengunjung dilakukan dengan memilih obyek wisata asal yaitu Museum Gudang Ransoem dan titik tujuannya adalah Masjid Agung Nurul Islam kemudian memilih algoritma yang akan

digunakan. Aplikasi akan menampilkan rute beserta jarak antar tempat wisata tersebut. Apabila ingin melihat peta, pengunjung dapat mengklik pada link peta. Dalam 5 kali uji sampel melalui *host to host*, algoritma Astar dan Dijkstra mampu menghasilkan rute yang sama. Perhitungan jarak dilakukan dengan peta analog dengan skala 1:40.000 dan menghasilkan perhitungan yang sedikit berbeda ketika menggunakan algoritma Dijkstra dan Astar dengan selisih jarak rata-rata adalah 0,016 km. Kecepatan algoritma Astar dalam pencarian rute terpendek lebih cepat daripada algoritma Dijkstra dengan selisih rata-rata 40ms. Aplikasi dapat menampilkan visualisasi data spasial Kota Sawahlunto dan menemukan rute terpendek pada 12 obyek wisata. Akan tetapi sistem mengalami beban berat karena menerima *query* setiap saat sehingga sangat dibutuhkan kestabilan server ketika sistem diakses banyak pengguna. Hasil pencarian akan lebih efektif jika ditambahkan parameter lain seperti biaya perjalanan, alternatif angkutan umum, dan peraturan lalu lintas (Pugas et al., 2011).

Dalam aplikasi pencari rumah sakit terdekat, hal pertama yang dilakukan adalah pengguna menekan *panic button* pada aplikasi kemudian diambil titik awal yang sesuai pada GPS dan mendapat lokasi rumah sakit terdekat dengan *Google Maps*. Dengan data yang telah didapat, algoritma Dijkstra dan R-Tree bekerja untuk menentukan jarak terpendek menuju rumah sakit tujuan dan aplikasi menampilkan rute yang direkomendasikan. Kedekatan objek pada algoritma R-Tree didefinisikan dalam ukuran berdasarkan jarak Euclidean dengan menampilkan posisi berdasarkan titik *longitude* dan *latitude*. Dalam aplikasi tersebut juga berisi keterangan mengenai informasi rumah sakit sesuai dengan data yang sudah disimpan dalam *database* telah yang disimpan. Aplikasi menggunakan data dari 50 Rumah Sakit di Kota Medan yang mendukung BPJS dan UMUM pada tahun 2017. Aplikasi terdiri dari halaman awal, halaman pencarian, dan halaman hasil rute. Pengujian dilakukan sebanyak lima kali untuk memastikan aplikasi dapat bekerja sesuai dengan yang diinginkan. Aplikasi ini mampu menampilkan rekomendasi rumah sakit terdekat dan *direction* untuk mendapatkan jarak tempuh dari lokasi asal ke tujuan. Untuk kesepannya sebaiknya tidak membatasi jumlah rumah sakit dan juga titik jalan agar aplikasi semakin baik dan juga sebaiknya ditambahkan titik kemacetan, rumah sakit maupun klinik (Sianturi, 2018).

IV. PENUTUP

Algoritma Dijkstra cukup baik untuk digunakan dalam penentuan rute terdekat menuju suatu tempat. Dalam sistem yang dibuat dapat ditentukan terlebih dahulu *database* yang akan digunakan sehingga dapat mengurangi beban pemrosesan data. Algoritma Dijkstra akan menampilkan rute terpendek yang sama dengan algoritma Astar, dengan waktu eksekusi sedikit lebih lama. Meskipun begitu, algoritma Dijkstra tetap lebih baik dalam penentuan rute terdekat terutama apabila digunakan untuk penentuan rute dengan jalur dan titik yang banyak kompleks. Spesifikasi *hardware* yang digunakan tidak akan mempengaruhi hasil rute, tetapi akan sangat berpengaruh pada waktu eksekusi. Dengan demikian algoritma Dijkstra cocok untuk digunakan dalam aplikasi penentu rute terdekat di

rumah sakit dengan terlebih dahulu memasukkan data ruangan dan jarak antar ruang ke dalam *database* sistem.

REFERENSI

- Ardana, D. and Saputra, R. (2016) 'Penerapan Algoritma Dijkstra pada Aplikasi Pencarian Rute Bus Trans Semarang', Skripsi Jurusan Ilmu Komputer, Fakultas Sains Dan Matematika, Universitas Diponegoro, (Snik), pp. 299–306.
- Dewi, L. J. E. (2010) 'Pencarian Rute Terpendek Tempat Wisata Di Bali Dengan Menggunakan Algoritma Dijkstra', Seminar Nasional Aplikasi Teknologi Informasi 2010 (SNATI 2010), 2010(Snati), pp. 46–49.
- Fisika, D. et al. (2016) Universitas Sumatera Utara - Beranda.
- Available at: <https://www.usu.ac.id/id/>.
- Ginting, J. V. and Barus, E. S. (2018) 'Aplikasi Penentuan Rute Rumah Sakit Terdekat Menggunakan Algoritma Dijkstra', Jurnal Mantik Penusa, 2(2), pp. 1–8.
- Pugas, D. O. et al. (2011) 'Dijkstra1', 13(1), pp. 27–32.
- Putra, A. B. W. et al. (2020) 'Perbandingan Hasil Rute Terdekat Antar Rumah Sakit di Samarinda Menggunakan Algoritma A*(star) dan Floyd-Warshall', Jurnal Sisfokom (Sistem Informasi dan Komputer), 9(1), p. 59. doi: 10.32736/sisfokom.v9i1.685.
- Sianturi, A. H. (2018) 'Universitas Sumatera Utara Skripsi', Analisis Kesadahan Total dan Alkalinitas pada Air Bersih Sumur Bor dengan Metode Titrimetri di PT Sucofindo Daerah Provinsi Sumatera Utara, pp. 44–48.