

PENERAPAN ALGORITMA DIJKSTRA UNTUK Mencari Lokasi Rumah Sakit Rujukan COVID-19 (*CORONA VIRUS DISEASE 2019*) SECARA ONLINE TERDEKAT DI SUMATERA UTARA MENGGUNAKAN *GEOGRAPHIC INFORMATION SYSTEM (GIS)*

Aldhi Pratama, Darjat Paripurna, Oris Krianto Sulaiman

Program Studi Teknik Informatika, Fakultas Teknik
Universitas Islam Sumatera Utara, Medan, Indonesia
aldhipratama92@gmail.com

Abstrak

Aplikasi pencarian rumah sakit rujukan covid-19 merupakan sebuah sistem yang ditujukan untuk membantu para tenaga kesetanan yang bekerja dalam menangani pasien covid-19 untuk memberikan bantuan perjalanan rujukan ke lokasi rumah sakit rujukan covid-19 yang telah dipilih oleh Kementerian Kesehatan Republik Indonesia untuk secara khusus menangani pasien yang terkena ataupun tertular virus corona yang telah menjadi wabah belakangan ini. Aplikasi ini dirancang untuk secara otomatis mencari lokasi terdekat berdasarkan jarak dan waktu dengan mengimplementasikan algoritma shortest path yaitu djikstra yang akan menyeleksi dan menentukan jalur terpendek antara dua node atau parameter yaitu lokasi awal dan tujuan untuk memberikan hasil yang di inginkan. Algoritma Dijkstra merupakan algoritma yang digunakan untuk menentukan lintasan terpendek pada sebuah graf yang kemudian dengan adanya bantuan dari API Map Service maka dapat ditentukan jarak lintasan dan waktu yang ditempuh berdasarkan keadaan jalan yang terbaik yang dipilih oleh layanan tersebut.

Kata-Kata Kunci : *Algoritma Dijkstra, API Map Service, Lokasi, Rumah Sakit,*

I. PENDAHULUAN

Wabah Virus Corona merupakan sebuah pandemi yang telah mengkhawatirkan seluruh negara, termasuk Indonesia. Hingga saat ini jumlah negara yang telah terinfeksi virus tersebut sudah mencapai angka 213 negara dari 246 negara di dunia menurut data *International Organization for Standardization (ISO) 3166-1*.

Rumah sakit merupakan sarana pelayanan Kesehatan yang tentunya saat ini menjadi *central point* dalam menangani situasi pandemi ini. Berbagai regulasi terkait pun banyak dikeluarkan oleh pemerintah termasuk Peraturan Menteri Kesehatan Republik Indonesia Nomor 9 Tahun 2020 Tentang Pedoman Pembatasan Sosial Berskala Besar Dalam Rangka Percepatan Penanganan *Corona Virus Disease 2019* menyebutkan bahwa rumah sakit dan semua instansi terkait, termasuk instansi medis terkait dan lainnya harus tetap berfungsi sebagaimana mestinya agar pelayanan terhadap kesehatan dapat terus terlaksana dalam menangani wabah ini.

Saat ini penyebaran berita terkait dengan Virus Corona melalui banyak platform baik itu media sosial maupun portal-portal berita telah sangat membantu dalam menyampaikan informasi kepada masyarakat Indonesia dalam mengetahui perkembangan serta hal-hal yang perlu diwaspadai dan yang harus dilakukan untuk mencegah penularan virus yang dapat menyebabkan penyakit ringan sampai berat, seperti pilek dan jika sampai pada tahap serius seperti MERS dan SARS. Sistem informasi yang dibuat oleh Gugus Tugas Percepatan Penanganan COVID-19 juga menyebutkan bahwa gejala penyakit meliputi demam yang melebihi 38

derajat Selsius disertai dengan batuk, sesak napas yang membutuhkan perawatan di rumah sakit yang akan lebih berat lagi jika penderita merupakan seorang lansia dan memiliki penyakit lainnya seperti penyakit paru obstruktif menahun dan penyakit jantung. Oleh karena itu sistem informasi sangat penting dalam memberikan pelayanan yang berguna bagi semua orang.

Pemerintah telah mengeluarkan rumah sakit rujukan khusus untuk menangani virus yang hingga saat ini belum ditemukan vaksinnnya, Oleh karena itu butuh penanganan khusus agar virus tersebut tidak menyebar ke mana-mana. Rumah sakit rujukan Covid-19 tersebar banyak di seluruh Indonesia termasuk di Provinsi Sumatera Utara yang saat ini telah mencapai 970 kasus yang terpapar penyakit menular ini. Dalam kegiatan darurat saat ini tentunya ketepatan dan waktu merupakan hal yang penting untuk menyelamatkan penderita agar sesegera mungkin dapat tiba dirumah sakit untuk memperoleh pelayanan kesehatan yang tepat.

Peta sering kita gunakan untuk mengetahui sebuah tempat atau lokasi. Seiring dengan berkembangnya teknologi informasi, kini peta tidak lagi hanya dalam bentuk media cetak namun telah menjadi sebuah layanan digital atau peta virtual yang dapat kita akses melalui *smartphone* kita atau pun laptop. Walaupun peta membantu kita untuk dapat mengetahui lokasi kita berada dan lokasi yang kita tuju namun tidak dapat membantu kita dalam menentukan jalur terpendek.

Terdapat cara untuk dapat menentukan jalur terpendek yaitu dengan menginterpretasikan peta kedalam sebuah graf. Dalam graf, terdapat metode yang dapat digunakan untuk menentukan jalur

terpendek yaitu Algoritma Dijkstra. Algoritma yang ditemukan oleh Edsger Dijkstra ini digunakan dalam graf berarah dimana setiap titik dihubungkan oleh sisi yang memiliki bobot. Dengan memperhitungkan bobot pada setiap sisi, algoritma ini efektif dalam menentukan jalur terpendek dari lokasi awal ke lokasi tujuan.

Berdasarkan referensi penelitian yang telah diterbitkan oleh Windi, Dwiretno, Abdul (2015) yang dijadikan salah satu referensi oleh penulis maka dapat disimpulkan bahwa algoritma djikstra mampu dalam menentukan jalur terpendek dengan menggunakan graf menggunakan media peta virtual dengan menguji atau memperhitungkan setiap jalur-jalur yang ada dari lokasi awal ke lokasi tujuan yang diinginkan.

Untuk mempermudah dalam mengetahui lokasi-lokasi rumah sakit rujukan covid-19 di Sumatera Utara serta kecepatan dalam mengakses lokasi tersebut algoritma djikstra dirasa mampu untuk melakukan perhitungan untuk menentukan lokasi terdekat menuju rumah sakit rujukan covid-19 dengan menghubungkan dua buah titik pada suatu graf yang kemudian ditampilkan oleh peta virtual.

Oleh karena itu penulis merasa pembuatan sistem informasi pencarian lokasi rumah sakit rujukan covid-19 terdekat di Sumatera Utara dibutuhkan dalam membantu masyarakat khususnya masyarakat di Sumatera Utara yang juga hingga saat ini penulis belum menemukan aplikasi yang senada yang dibuat oleh pemerintah untuk menangani situasi wabah saat ini.

II. LANDASAN TEORI

2.1 Rumah Sakit

2.1.1 Defenisi Rumah Sakit

Menurut Keputusan Menteri Kesehatan Republik Kesehatan Republik Indonesia No. 340/MENKES/PER/III/2010 bahwa, Rumah sakit adalah institusi pelayanan kesehatan yang menyelenggarakan pelayanan kesehatan perorangan secara paripurna yang menyediakan pelayanan rawat inap, rawat jala dan gawat darurat (Setyawan & Supriyanto, 2019).

Rumah sakit merupakan salah satu bagian penting dalam sistem kesehatan. Dalam sistem kesehatan di seluruh negara tujuan utama yang ingin dicapai adalah tercapainya tingkat efisiensi yang optimal (Irwandy, 2019). Pencapaian efisiensi di rumah sakit menjadi suatu hal yang sangat penting untuk dicapai dikarenakan rumah sakit merupakan institusi yang padat biaya dan menghabiskan 50% - 89% pembiayaan dalam sebuah sistem kesehatan. Pentingnya efisiensi sebagai salah satu indikator kinerja sebuah rumah sakit juga telah diakui secara global oleh *World Health Organization* (WHO).

Pertumbuhan rumah sakit di Indonesia setiap tahunnya terus mengalami peningkatan. Menurut profil Indonesia, sejak tahun 2005 hanya terdapat 1.268 RS di Indonesia dan terus bertambah hingga

akhir 2018 sebanyak 2.813 RS terdiri dari 2.269 RS Umum dan 544 RS Khusus.

2.1.2 Jenis Rumah Sakit

Berdasarkan pasal 18 UU 44/2009 yang mengatur bahwa rumah sakit mempunyai tugas memberikan pelayanan kesehatan perorangan secara paripurna. Kemudian disambung dengan pasal 19 dan 20 ayat 1-2 menyatakan bahwa jenis rumah sakit dibedakan menjadi 2 yaitu :

1. Rumah Sakit Umum, memberikan pelayanan kesehatan pada semua bidang dan jenis penyakit.
2. Rumah Sakit Khusus, memberikan pelayanan utama pada satu bidang atau jenis penyakit tertentu berdasarkan disiplin ilmu, golongan umur, organ, jenis penyakit, atau kekhususan lainnya.

2.2 Pengertian Website

Website adalah kumpulan informasi/kumpulan page yang biasa diakses lewat jalur internet. Setiap orang di berbagai tempat dan segala waktu bisa menggunakannya selama terhubung secara online di jaringan internet. Secara teknis, Website adalah kumpulan dari *page*, yang tergabung kedalam suatu *domain* atau *subdomain* tertentu (Junirianto, 2018).

Sebuah situs web biasanya ditempatkan setidaknya pada sebuah server web yang dapat diakses melalui jaringan internet. "*Website* atau disingkat web, dapat diartikan sekumpulan halaman yang terdiri dari beberapa laman yang berisi *text*, gambar, video, audio, dan animasi lainnya.

2.3 Geographic Information System (GIS)

Sistem informasi Geografis (GIS) diartikan sebagai sistem untuk menyimpan, memeriksa, mengintegrasikan, memanipulasi, menganalisis dan memaparkan data yang berkaitan dengan semua ruang yang berhubungan dengan keadaan bumi (Awangga, 2019).

GIS adalah sistem informasi geografi yang digunakan untuk mengelola data informasi keruangan (Satria, 2016). Tujuan utama pengoperasian GIS adalah untuk memudahkan informasi yang diperoleh, diolah dan disimpan sebagai sebuah data yang mampu menggambarkan sebuah lokasi ataupun obyek tertentu.

2.4 Web Browser

Peramban web atau lebih dikenal dengan web browser merupakan perangkat lunak yang berfungsi untuk menerima dan menyajikan sumber informasi di internet. Mudahnya, browser digunakan untuk menampilkan halaman-halaman web. Seluruh komponen web termasuk teks, gambar, dan komponen lain yang dibangun dengan teknologi *client-side scripting* dapat ditampilkan di *web browser* (Solichin, 2016).

Menurut (Pahrudin, 2017), fungsi web browser yang paling menonjol mungkin ada pada kemampuannya untuk mengarahkan pengguna pada alamat yang dituju. Dengan menggunakan web

browser seseorang bisa langsung menuju alamat dimana ia akan mengakses data sementara jika menggunakan *search engine* maka ia hanya akan mendapatkan beberapa *website* yang sesuai dengan kata kunci yang dimasukkan.

2.5 Web Server

Server adalah pemilik informasi yang menyediakan dirinya untuk memberikan *service* atau layanan. Sedangkan Web server merupakan penyimpanan data web. Misalnya sebuah komputer menyimpan data web miliknya sendiri. Ia memberikan *service* kepada komputer lain yang ingin mengakses data tersebut. Komputer yang mengajukan permintaan untuk mengakses web itu disebut *client* (Frans, 2002).

Web server adalah sebuah aplikasi server yang menggunakan protokol HTTP (*HyperText Transfer Protocol*) atau HTTPS (*Hypertext Transfer Protocol Secure*) untuk memberikan layanan berbasis file dalam bentuk halaman web kepada pengguna. Salah satu contoh dari webserver adalah Apache (Husen & Surbakti, 2020).

2.6 XAMPP

XAMPP adalah perangkat lunak sumber terbuka yang dikembangkan oleh teman-teman Apache. Paket perangkat lunak XAMPP berisi distribusi Apache untuk server Apache, Maria DB, PHP, dan Perl. Pada dasarnya Xampp adalah tuan rumah lokal atau server lokal. Server lokal ini berfungsi di komputer desktop atau laptop anda sendiri. Penggunaan XAMPP adalah untuk menguji klien atau website anda sebelum mengunggahnya ke server web jarak jauh. Perangkat lunak server XAPP ini memberi anda lingkungan yang cocok untuk menguji proyek MySQL, PHP, Apache di komputer lokal. Bentuk lengkap XAMPP adalah X singkatan dari *Cross platform*, (A) server APache, (M) MariaDB, (P) PHP dan (P) Perl. *Cross-platform* biasanya berarti dapat berjalan dikomputer mana saja dengan sistem operasi apapun (Saputra & Aprillian, 2020).

2.7 PHP

PHP atau singkatan dari Hypertext Preprocessor adalah bahasa scripting Web HTML-embedded. PHP adalah bahasa scripting dengan tujuan umum *open source* yang banyak digunakan dan sangat cocok untuk pengembangan web dan dapat disematkan ke dalam HTML. Ini berarti kode PHP dapat dimasukkan ke HTML halaman web. Saat halaman PHP dikases, kode PHP dibaca atau “diurai” oleh server tempat halaman itu berada. Output dari fungsi PHP pada halaman biasanya dikembalikan sebagai kode HTML, yang dapat dibaca oleh *browser*. Karena kode PHP diubah menjadi HTML sebelum halaman dimuat, pengguna tidak dapat melihat kode PHP pada halaman. Ini membuat halaman PHP cukup aman untuk mengakses *database* dan informasi aman lainnya (Saputra & Aprillian, 2020).

2.8 Bootstrap

Bootstrap adalah *framework front-end* yang lebih intuitif dan *powerfull* untuk pengembangan aplikasi *web* yang lebih cepat dan mudah. Bootstrap menggunakan HTML, CSS dan Javascript (Enterprise, 2016).

Bootstrap adalah sebuah pustakan *open source* yang merupakan *framework* CSS dan Javascript untuk membuat *website* yang responsif (Sulistiono, 2018).

2.9 MySQL

Basis data (*data base*) adalah kumpulan informasi yang disimpan di dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut. Untuk mengelola *database* diperlukan suatu perangkat lunak yang disebut DBMS (*Data Base Management System*). DBMS merupakan suatu sistem perangkat lunak yang memungkinkan user (pengguna) untuk membuat, memelihara, mengontrol dan mengakses *database* secara praktis dan efisien.

MySQL merupakan salah satu perangkat lunak basis data yang sangat populer. Salah satu keunggulan MySQL adalah kemudahan penggunaan dan pengelolannya. Untuk mengakses *database* MySQL juga dapat dilakukan dengan berbagai MySQL *Client*. Selain menggunakan MySQL *Client* bawaan berbasis *command line*, saat ini banyak sekali *tools* yang dikembangkan oleh pihak ketiga untuk mempermudah pengelolaan *database* MySQL. Sebagian besar menyediakan fitur-fitur berbasis GUI yang sangat memudahkan penggunaannya (Solichin, 2016).

2.10 Konsep Algoritma

2.10.1 Pengertian Algoritma

Algoritma merupakan jantung dari ilmu komputer (teknik informatika). Ditinjau dari usul katanya, awalnya orang hanya menemukan kata “*algorism*” yang berarti proses menghitung dengan angka arab. Para ahli sejarah matematika akhirnya menemukan asal kata tersebut yang berasal dari nama penulis buku arab terkenal, yaitu Abu Ja’far Muhammad Ibnu Musa Al-Khuwarizmi.

Al-Khuwarizmi (oleh orang barat menjadi *Algorism*) menulis buku yang berjudul *Al Jabar Wal-Muqabala* yang artinya “Buku pemugaran dan Pengurangan”. Karena *Algorism* sering dihubungkan dengan *arithmetic*, sehingga akhiran -sm berubah menjadi -thm. Akhirnya lambar laun kata *algorithm* berangsur-angsur digunakan sebagai metode perhitungan (komputasi). Dalam Bahasa Indonesia *algorithm* diserap menjadi algoritma.

Algoritma merupakan serangkaian instruksi yang telah didefinisikan dengan baik untuk menghitung, yang dimulai dari suatu kondisi awal dan inputan awal, dimana instruksi-instruksi tersebut menjelaskan suatu komputasi yang bila dieksekusi dan diproses melewati sejumlah urutan kondisi terbatas yang terdefinisi dengan baik, yang pada

akhirnya menghasilkan suatu keluaran dan akan berhenti di kondisi akhir (Azis dkk, 2019). Sementara itu, definisi prosedur menurut Kamus Besar Bahasa Indonesia adalah :

1. Tahap-tahap kegiatan untuk menyelesaikan suatu aktivitas ;
2. Metode langkah demi langkah secara eksak dalam memecahkan suatu masalah

Dalam bidang matematika dan informatika, algoritma adalah serangkaian prosedur yang saling berinteraksi untuk mencapai suatu tujuan dalam memecahkan masalah tertentu. Sehingga algoritma dapat digunakan untuk perhitungan, pemrosesan data, penalaran otomatis, optimasi, dan sejenisnya untuk menyelesaikan sejumlah masalah. Agar dapat dilaksanakan oleh komputer, maka algoritma harus ditulis dalam notasi bahasa pemrograman sehingga dinamakan program komputer.

Menurut (Sitorus, 2015), suatu algoritma merupakan suatu *sequence* (runtunan) beberapa aksi baris terakhir sesuai dengan urutan aksi tanpa ada pengulangan pelaksanaan aksi ataupun lompatan akibat suatu kondisi tertentu. Misalkan suatu algoritma terdiri dari 4 (empat) baris aksi secara berurutan t1, t2, t3 dan t4, maka urutan pelaksanaan aksi sesuai dengan urutan aksi. Aksi t2 akan dilaksanakan aksi t1 selesai dilaksanakan, aksi t3 akan dilaksanakan setelah aksi t2 selesai dilaksanakan, aksi t4 akan dilaksanakan setelah aksi t3 selesai dilaksanakan dan akan berhenti karena aksi t4 adalah aksi terakhir.

2.10.2 Jalur Terpendek

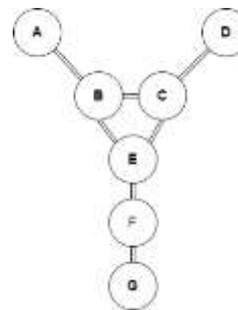
Penentuan sebuah jalur terpendek merupakan hal yang penting dan dibutuhkan sehubungan dengan optimasi waktu yang digunakan serta beberapa penghematan dibidang lainnya. Dengan jalur terpendek yang dilalui, membuat pekerjaan lebih efektif, cepat dan dapat terntunya terjadi penghematan biaya. Jalur terpendek dapat diartikan sebagai nilai minimal dari suatu lintasan, yaitu jumlah nilai dari keseluruhan bentuk lintasan (Harahap & Khirina, 2017) .

Konsep *shortest path* sebenarnya kerap kali kita pakai sehari-hari. Kalau anda ingin menghubungi lurah dikampung untuk mengurus suatu dokumen, Anda bisa mengikuti prosedur resmi, dengan mendaftar sampai ke tangan Pak Lurah, Tetapi jika Anda kebetulan kenal dengan sekertaris kelurahan atau kenal dengan orang yang dekat Pak Lurah, mungkin Anda cukup melakukan dua langkah hingga dokumen tersebut sampai ke tagan Pak Lurah.

Jalur terpendek (*shortest path*) dihitung untuk masing-masing aktor (node) ketika menghubungi semua aktor lain dalam jaringan. Lihat dalam ilustrasi Gambar 1 aktor A membutuhkan satu jalur (path) untuk menghubungi B, butuh 2 langkah untuk menghubungi C dan E, tiga langkah (jalur) untuk menghubungi F dan 4 langkah untuk menghubungi G. Jalur yang diperhitungkan disini yakni jalur terpendek.

Tabel 1. Jumlah jalur

	A	B	C	D	E	F	G
A		1	2	3	2	3	4
B	1		1	2	1	2	3
C	2	1		1	1	2	3
D	3	2	1		2	3	4
E	2	1	1	2		1	2
F	3	2	2	3	1		1
G	4	3	3	4	2	1	



Gambar 1. Jalur

Sebagai misal, aktor A untuk menghubungi E bisa menggunakan beberapa jalur :

- a) Jalur A-B-E
- b) Jalur A-B-C-E.

Karena yang terpendek yaitu A-B-E (2 Jalur), maka yang diperhitungkan yaitu jalur ini. Perhatikan, tidak dibolehkan jalur A-B-C-D-C-E. Karena dalam jalur tidak di perkenankan pengulangan aktor (node).

2.10.3 Algoritma Dijkstra

Algoritma Dijkstra adalah sebuah algoritma untuk menentukan lintasan terpendek dalam graf tak berarah (atau graf berarah) berbobot tanpa mengenumerisasi secara eksplisit semua lintasan yang mungkin. Algoritma ini didasarkan pada sebuah metode yang dikenal sebagai pemrograman dinamik.

Algoritma Dijkstra menentukan lintasan terpendek diantara pasangan-pasangan titik dalam suatu graf. Jika terdapat n titik dalam graf, kita perlu mengeksekusi algoritma nC2 kali (Marsudi, 2016).

Secara umum cara kerja algoritma Dijkstra adalah :

1. Memberi nilai bobot (jarak) dari satu titik ke titik lainnya, kemudian tentukan nilai 0 pada titik awal dan nilai tak terhingga (∞) pada titik yang lain atau titik yang belum terisi.
2. Tentukan semua titik awal yang belum dilewati dan aturlah titik awal sebagai titik keberangkatan.
3. Dari titik keberangkatan, pertimbangkan titik lain yang belum terlewati dan hitung jaraknya dari titik keberangkatan. Contohnya titik X ke titik Y memiliki (jarak) 10 dan dari titik Y ke titik Z bernilai 5. Maka, jarak ke titik C melewati titik B adalah $10 + 5 = 15$. Jika jarak ini lebih kecil

- dari sebelumnya yang telah dicatat, hapus data lama dan simpan ulang nilai (jarak) yang baru.
4. Ketika setiap jarak terhadap titik lain telah dipertimbangkan, tandai titik yang telah dilewati sebagai “titik terlewati”. Titik terlewati tidak akan pernah diperiksa kembali, dan jarak yang disimpan adalah jarak terakhir dari titik keberangkatan dan yang paling minimal nilainya (jaraknya).
 5. Tentukan titik belum terlewati dengan nilai (jarak) terkecil dari titik keberangkatan, sebagai titik keberangkatan selanjutnya, dan lanjutkan dengan mengulang kembali langkah ke-3.

2.10.4 Notasi Algoritma Dijkstra

Terdapat beberapa notasi utama dalam pengerjaan Algoritma Dijkstra ini :

- Inisiasi
- $N = \{s\}$
 $D_j = C_{sj} \forall j \neq s$
 $D_s = 0$

Proses inisiasi dibentuk suatu array/himpunan N dengan anggota s (s adalah lambang untuk suatu node sumber). Nilai D adalah jarak yang akan tersedia pada tabel hasil. Dan C adalah nilai jarak map yang tersedia. Maka pada tahap inisiasi ini nilai Dj (jarak pada hasil tabel antara node s dengan node j tidak sama dengan s) dimasukkan nilai yang sebenarnya. Jika tidak tersambung secara langsung maka akan dianggap tak terdefenisi. Untuk Ds tentu saja bernilai 0.

- Temukan simpul tetangga (node selain sumber)

$$D_i = \max_{j \in N} D_j$$

Untuk perulangan tiap baris dimasukkan node i yang belum termasuk pada array/himpunan N untuk nanti node i tersebut dijadikan sebagai “perpanjangan” dari node s, dengan node i juga merupakan node tetangga dari node s. Node i yang dimasukkan pada himpunan N berdasarkan pada jarak terkecil dengan node s. Dan jika seluruh node sudah masuk dalam himpunan N, maka iterasi akan berhenti.

- Update untuk setiap $j \in N$
- $D_j = \min \{D_j, D_i + C_{ij}\}$

Untuk setiap node j (dalam tabel hasil: tiap kolom) diperbaharui nilainya yang paling kecil yaitu membandingkan antara nilai Dj sebelumnya dengan hasil penjumlahan ($D_i + C_{ij}$), yaitu penjumlahan yaitu penjumlahan jarak node s ke node i dengan jarak sebenarnya dari node i ke node j.

Langkah – langkah Algoritma Dijkstra :

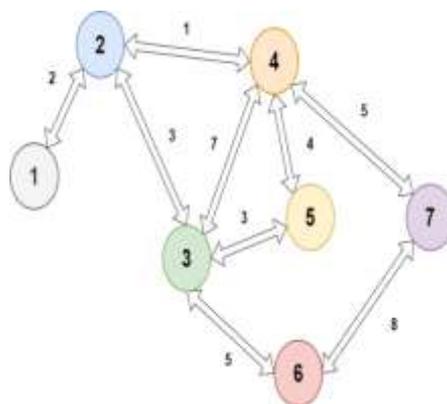
a. Pengambilan Data

Ketika menyelesaikan Algoritma Dijkstra, dibutuhkan data untuk melakukan perhitungan nilai (jarak) minimum. Dalam hal ini, data diambil berdasarkan nilai (jarak) atau biaya yang dapat diperoleh melalui posisikoordinat suatu wilayah. Posisi koordinat dapat diambil melalui Google Maps.

Dalam pengambilan data ini, perlu diketahui bahwa setiap koordinat yang diambil harus bernilai positif. Jika koordinat tersebut bernilai negatif, maka pencarian tidak dapat dilakukan. Setelah mendapatkan titik koordinat tersebut, dilakukan pencarian nilai (jarak) atau waktu (biaya) antar titik yang berhubungan atau yang ada disekitarnya.

b. Membuat Skema

Skema merupakan suatu bentuk rancangan atau kerangka secara garis besar yang membuat gambaran umum dari apa yang akan diperhitungkan. Skema yang dibuat berupa gambar antara titik yang saling terhubung oleh garis sesuai dengan posisi koordinat yang telah didapat sebelumnya.



Gambar 2. Contoh Skema Algoritma Dijkstra

Gambar 2 merupakan contoh dari sebuah skema yang menggambarkan nilai (jarak) antar titik bernilai positif dan titik awal keberangkatan dimulai dari titik 1.

c. Membuat Tabel Perhitungan

Langkah berikutnya ialah membuat tabel perhitungan, dimana langkah ini sangat penting karena tabel ini akan menentukan proses secara keseluruhan (notasi dalam tabel Algoritma Dijkstra memiliki format (s-j,D), dimana s-j menunjukkan rute dari node s menuju node j, sementara D menunjukkan jarak total antara kedua node tersebut).

Tabel 2. Perhitungan 1

N	D2	D3	D4	D5	D6	D7
{1}	(1-2, 2)	∞	∞	∞	∞	∞

Baris pertama merupakan inisiasi, yaitu Dj akan memiliki nilai jika tersambung langsung dan tidak memiliki nilai jika tidak tersambung langsung.

Karena node 1 hanya memiliki 1 tetangga yaitu node 2, maka i = 2 dimasukkan pada himpunan N.

Tabel 3. Perhitungan 2

N	D2	D3	D4	D5	D6	D7
{1}	(1-2, 2)	∞	∞	∞	∞	∞
{1,2}	(1-2, 2)	(1-2-3,5)	(1-2-4,3)	∞	∞	∞

Node 2 sudah berperan sebagai “perpanjangan” node sumber (node 1), sehingga sekarang node-node yang terhubung dengan node 2 sudah bisa “dijangkau” oleh node 1 via node 2. Diketahui node 3 dan node 4 terhubung langsung dengan node 2, sehingga rutenya ditulis (1-2-3) dan (1-2-4). Untuk langkah selanjutnya dipilih node i yang telah tersambung dengan node s namun belum masuk dalam himpunan N. Diketahui yaitu node 3 dan 4. Node yang dipilih adalah yang memiliki jumlah jarak yang paling minimum, yaitu node 4. Sehingga didapat baris tabel berikutnya :

Tabel 4. Perhitungan 3

N	D2	D3	D4	D5	D6	D7
{1}	(1-2, 2)	∞	∞	∞	∞	∞
{1,2}	(1-2, 2)	(1-2-3,5)	(1-2-4,3)	∞	∞	∞
{1,2,4}	(1-2, 2)	(1-2-3,5)	(1-2-4,3)	(1-2-4-5,7)	∞	(1-2-4-7,8)

Seperti langkah sebelumnya, sekarang node 4 ikut berperan sebagai “perpanjangan” dari node 1 sehingga node 5 dan node 7 yang terhubung langsung dengan node 4 bisa “dijangkau” oleh node 1 dengan rute yang terampil.

Sebenarnya disini mulai terjadi perbandingan jarak. Dengan dimasukkannya node 4 dalam himpunan N, maka node 4 berlaku sebagai “penembak sementara” yaitu adalah node 4 dapat melakukan perhitungan minimum menuju node tertentu meskipun node tersebut telah diketahui rute dan jaraknya.

Dalam kasus ini dapat diambil node 3. Node 3 sudah diketahui rute dan jaraknya pada baris ke-2. Dengan masuknya node 4 pada himpunan N, maka node 4 akan menghitung jarak minimum antara entry D3 yang terdahulu dengan yang baru.

Perbandingan via node 2 dengan node 4 :

Via node 2 → $D2 + C23 = 2 + 3 = 5$

Via node 4 → $D4 + C43 = 3 + 7 = 10$

Maka entry yang dipertahankan adalah via node 2 dengan jarak 5.

Sebelumnya telah dipilih node 4 sebagai anggota N karena bertetangga dengan node 2. Sekarang dipilih tetangga node 2 yang lainnya yaitu node 3.

Tabel 5. Perhitungan 4

N	D2	D3	D4	D5	D6	D7
{1}	(1-2,2)	∞	∞	∞	∞	∞
{1,2}	(1-2,2)	(1-2-3,5)	(1-2-4,3)	∞	∞	∞
{1,2,4}	(1-2,2)	(1-2-3,5)	(1-2-4,3)	(1-2-4-5,7)	∞	(1-2-4-7,8)
{1,2,3,4}	(1-2,2)	(1-2-3,5)	(1-2-4,3)	(1-2-4-5,7)	(1-2-3-6,10)	(1-2-4-7,8)

Perbandingan yang terjadi ;

Pada D4

Via node 2 → $D2 + C24 = 2 + 1 = 3$

Via node 3 → $D3 + C34 = 5 + 7 = 12$

Maka entry yang dipertahankan adalah via node 2 dengan jarak 3.

Pada D5

Via node 4 → $D4 + C45 = 3 + 4 = 7$

Via node 3 → $D3 + C35 = 5 + 3 = 8$

Maka entry yang dipertahankan adalah via node 4 dengan jarak 7

Tabel 6. Perhitungan 5

N	D2	D3	D4	D5	D6	D7
{1}	(1-2,2)	∞	∞	∞	∞	∞
{1,2}	(1-2,2)	(1-2-3,5)	(1-2-4,3)	∞	∞	∞
{1,2,4}	(1-2,2)	(1-2-3,5)	(1-2-4,3)	(1-2-4-5,7)	∞	(1-2-4-7,8)
{1,2,3,4}	(1-2,2)	(1-2-3,5)	(1-2-4,3)	(1-2-4-5,7)	(1-2-3-6,10)	(1-2-4-7,8)
{1,2,3,4,5,7}	(1-2,2)	(1-2-3,5)	(1-2-4,3)	(1-2-4-5,7)	(1-2-3-6,10)	(1-2-4-7,8)

Karena hubungan antara node 5 dan node 6 tidak langsung sehingga nilai D6 tidak berubah.

Perbandingan yang terjadi :

Pada D7

Via node 4 = $D4 + C47 = 3 + 5 = 8$

Via node 6 = $D6 + C67 = 10 + 8 = 18$

Maka entry yang dipertahankan adalah via node 4 dengan jarak 8.

Tabel 7. Perhitungan 6

N	D2	D3	D4	D5	D6	D7
{1}	(1-2,2)	∞	∞	∞	∞	∞
{1,2}	(1-2,2)	(1-2-3,5)	(1-2-4,3)	∞	∞	∞
{1,2,4}	(1-2,2)	(1-2-3,5)	(1-2-4,3)	(1-2-4-5,7)	∞	(1-2-4-7,8)
{1,2,3,4}	(1-2,2)	(1-2-3,5)	(1-2-4,3)	(1-2-4-5,7)	(1-2-3-6,10)	(1-2-4-7,8)
{1,2,3,4,5,7}	(1-2,2)	(1-2-3,5)	(1-2-4,3)	(1-2-4-5,7)	(1-2-3-6,10)	(1-2-4-7,8)

Maka hasil yang diperoleh adalah rute node 1 → node 2 → node 4 → node 7, dengan jarak 8.

III. METODE PENELITIAN

3.1 Analisis Permasalahan

Analisis dan perancangan sistem adalah sebuah sistem mengenai penentuan lokasi terdekat dengan alur tercepat untuk mengakses rumah sakit rujukan covid-19 untuk pasien terpapar covid-19 dengan menggunakan layanan peta virtual. Sistem ini bertujuan untuk membantu penanganan terhadap pasien terpapar covid-19 untuk segera di rujuk ke rumah sakit rujukan covid-19 terdekat secara online yang nantinya dapat digunakan baik untuk tenaga medis rumah sakit maupun oleh masyarakat.

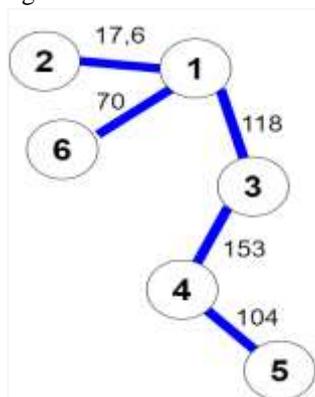
Sebagai contoh disebuah puskesmas terdapat seseorang yang baru saja teridentifikasi terinfeksi virus covid-19 setelah melakukan cek PCR (*polymerase chain reaction*) pada puskesmas tersebut. Dikarenakan puskesmas tersebut tidak

memadai dalam menangani pasien covid tersebut maka pasien tersebut harus segera dirujuk ke rumah sakit rujukan covid-19 terdekat untuk mendapatkan penanganan lebih lanjut juga untuk menghindari kontak dengan orang lain agar tidak ikut terkontaminasi oleh virus tersebut. Kemudian digunakanlah aplikasi pencarian rumah sakit rujukan terdekat untuk mengakses rumah sakit tersebut. Dengan mengandalkan algoritma djikstra sebagai pengukur jarak terdekat dan juga penggunaan layanan *gis(geographic informatic system)* yang akan melihat kondisi geografis pada alur perjalanan yang akan ditempuh kemudian aplikasi tersebut akan merekomendasikan lokasi rumah sakit rujukan covid-19 yang terdekat untuk di tuju.

Misalkan untuk mencari rumah sakit rujukan covid-19terdekat di sumatera utara dengan lokasi awal yang merupakan sebuah puskesmas di Gg. Puskesmas, Sigara Gara, Kec. Patumbak, Kabupaten Deli Serdang, Sumatera Utara 20361. Untuk menghitung jalur terpendek maka terlebih dahulu perlu adanya sebuah graf dengan bobot jarak dengan keterangan pada tabel berikut :

Tujuan	Dari Titik	Menuju Titik	Jarak	Lokasi
2	1	2	17,6	RSU H.Adam Malik
3	1	3	118	RSUP Djasamen Saragih
4	1	3	118	RSUD Tarutung
	3	4	153	
5	1	3	118	RSU Padang Sidempuan
	3	4	153	
	4	5	104	
6	1	6	70	RSU Kabanjahe

Kemudian berdasarkan data dari tabel di atas maka selanjutnya kita dapat membuat sebuah graf berbobot sebagai berikut :



Gambar 3. Graf Berbobot

Untuk setiap lokasi tujuan yang di peroleh kita lakukan perhitungan berikut :

- Untuk lokasi tujuan pada Node 2

N	D2	D3	D4	D5	D6
{1}	(1-2,17.6)	∞	∞	∞	∞

Pada lokasi tujuan Node 2 yang mana tidak melewati bagian dari node lain diperoleh hasil jarak = 17,6 km

- Untuk lokasi akhir pada Node 3

Tabel 3.1 Data JarakLokasi

N	D2	D3	D4	D5	D6
{1}	∞	(1-3,118)	∞	∞	∞

Untuk lokasi tujuan Node 3 yang sama seperti Node 2 diperoleh hasil jarak = 118 km

- Untuk lokasi akhir pada Node 4

N	D2	D3	D4	D5	D6
{1}	∞	(1-3,118)	∞	∞	∞
{1,3}	∞	(1-3,118)	(1-3-4,271)	∞	∞

Pada lokasi tujuan Node 4 dimana jalur yang ditempuh melewati node 3 maka diperoleh hasil jarak = D3 (Distance Between Node 1 and 3) + D4 = 118 + 153 = 271 km.

- Untuk lokasi akhir pada Node 5

N	D2	D3	D4	D5	D6
{1}	∞	(1-3,118)	∞	∞	∞
{1,3}	∞	(1-3,118)	(1-3-4,271)	∞	∞
{1,3,4}	∞	(1-3,118)	(1-3-4,271)	(1-3-4-5,375)	∞

Pada lokasi tujuan Node 5 terdapat beberapa Node yang berperan yaitu Node 3 dan Node 4 oleh karena itu jarak yang diperoleh = D3 + D4 + D5 = 118 + 153 + 104 = 375 km

- Untuk lokasi akhir pada Node 6

N	D2	D3	D4	D5	D6
{1}	∞	∞	∞	∞	(1-6,70)

Untuk lokasi tujuan pada Node 6 persis sama dengan Node 2 dan 3 yaitu= 70 km

Dari hasil perhitungan diatas maka diperoleh hasil :

- N2 = 17,6 km
- N3 = 118 km
- N4 = 271 km
- N5 = 375 km
- N6 = 70 km

Dengan demikian lokasi terdekat dari lokasi Gg. Puskesmas, Sigara Gara, Kec. Patumbak, Kabupaten Deli Serdang, Sumatera Utara 20361 adalah menuju RSU H.Adam Malik yang memiliki jarak tempuh terpendek yaitu 17,6 km.

IV. IMPLEMENTASI DAN HASIL

4.1 Implementasi

Tahap ini merupakan tahap menerapkan hasil analisis dan perancangan yang telah dibuat agar dapat berjalan sesuai dengan yang diharapkan yang kemudian dapat membantu pengguna aplikasi dalam menentukan lokasi rumah sakit rujukan covid-19 terdekat dan juga dapat memperkirakan jarak dan waktu yang ditempuh. Dalam mengimplementasikan Algoritma Dijkstra pada aplikasi berbasis web ini penulis melakukan pengkodean dengan menggunakan bahasa php sebagai bahasa pemrograman pembuatannya. Berikut merupakan script yang penulis gunakan untuk melakukan pencarian lokasi terdekat pada bagian user dan admin.

4.2 Pengujian Metode Black Box

Setelah sistem yang dirancang sesuai dengan kebutuhan selesai dibuat maka proses dilanjutkan dengan tahap menguji sistem tersebut. Pada pengujian menggunakan metode black box testing penulis melakukan pengujian terhadap fungsi – fungsi sistem yang mana pengujian black box adalah pengujian yang berfokus pada persyaratan fungsional sistem.

Adapun fungsi – fungsi yang diuji adalah sebagai berikut :

1. Fungsi tambah data kota
2. Fungsi tambah data rumah sakit
3. Fungsi edit data kota
4. Fungsi edit data rumah sakit
5. Fungsi simpan data kota
6. Fungsi simpan data rumah sakit
7. Fungsi hapus data kota
8. Fungsi hapus data rumah sakit
9. Fungsi pilih kota
10. Fungsi cek uji

4.3 Hasil Uji Black Box

Tabel 8. Pengujian Black Box

NO	Pengujian	Skenario Pengujian	Hasil Yang Diharapkan	Hasil Pengujian
1	Fungsi tambah data kota	Admin memasukkan data kota baru	Data berhasil tersimpan database	baru ke Berhasil
2	Fungsi tambah data rumah sakit	Admin memasukkan data rumah sakit baru	Data berhasil tersimpan database	baru ke Berhasil
3	Fungsi edit data kota	Admin melakukan pembaharuan pada data kota	Data berhasil tersimpan database	baru ke Berhasil
4	Fungsi edit data rumah sakit	Admin melakukan pembaharuan pada data	Data berhasil tersimpan database	baru ke Berhasil

5	Fungsi simpan data kota	Admin melakukan penyimpanan dengan mengklik tombol “simpan”	Data berhasil tersimpan database	baru ke Berhasil
6	Fungsi simpan data rumah sakit	Admin melakukan penyimpanan dengan mengklik tombol “simpan”	Data berhasil tersimpan database	baru ke Berhasil
7	Fungsi hapus data kota	Admin menghapus data kota	Data berhasil dihapus database	baru dari Berhasil
8	Fungsi hapus data rumah sakit	Admin menghapus data rumah sakit	Data berhasil dihapus database	baru dari Berhasil
9	Fungsi pilih kota	Pengguna memilih kota yang tersedia pada halaman peta rumah sakit	Sistem menampilkan daftar rumah sakit pada kota yang telah dipilih	Berhasil
10	Fungsi cek uji	Pemilih melakukan uji pencarian lokasi terdekat pada halaman lokasi terdekat	Sistem menampilkan hasil dari pengujian lokasi terdekat	Berhasil

V, KESIMPULAN DAN SARAN

5.1 Kesimpulan

Kesimpulan dari penelitian ini disajikan sebagai berikut :

1. Algoritma Dijkstra yang di implementasikan untuk mencari lokasi rumah sakit rujukan covid-19 mampu menentukan lokasi terdekat dengan mengambil jarak terkecil antara lokasi awal dan lokasi-lokasi rumah sakit yang tertera di Provinsi Sumatera utara.
2. Aplikasi ini mampu memperhitungkan jarak antara lokasi awal dengan lokasi-lokasi rumah sakit rujukan covid-19 yang terdapat di Provinsi Sumatera Utara dengan layanan Map API dengan mempertimbangkan kondisi geografis terbaik dan juga memperkirakan waktu yang ditempuh untuk tiba dilokasi tujuan.

5.2 Saran

Berdasarkan kesimpulan diatas ada beberapa hal yang penulis sarankan bagi pihak-pihak yang ingin mengembangkan aplikasi ini yaitu untuk memperdetail informasi dan settingan jalur graph pada aplikasi ini dan juga aplikasi ini dapat mempercantik tampilan antarmuka serta menambahkan animasi-animasi dan juga petunjuk-petunjuk yang dapat mempermudah pengguna dapat menggunakan aplikasi tersebut.

DAFTAR PUSTAKA

- [1] Adiputra, N. P., 2020, *Dasar-Dasar Teknik Informatika*. Yogyakarta: Deepublish.
- [2] Awangga, R. M., 2019, *Pengantar Sistem Informasi Geografis*. Bandung: Kreatif Industri Nusantara.
- [3] Azis, A. I., & Mustofa, Y. A., 2019, *Fundamental Pemrograman*. Yogyakarta: Deepublish.
- [4] Enterprise, J.. 2016. *Pemrograman Bootstrap Untuk Pemula*. Jakarta: PT Elex Media Komputindo.
- [5] Gratier, T., Spencer, P., & Hazzard, E.. 2015, *Open Layers 3 Beginner's Guide*. Birmingham: Packt Publishing.
- [6] Harahap, M. K., & Khirina, N.. 2017, Pencarian Jalur Terpendek dengan Algoritma Dijkstra. *SinkrOn*.
- [7] Husen, Z., & Surbakti, M. S., 2020, *Membangun Server Dan Jaringan Komputer Dengan Linux Ubuntu*. Banda Aceh: Syiah Kuala University Press.
- [8] Junirianto, E., 2018, *Pemrograman Web Dengan Framework Laravel*. Ponorogo: Wade Group.
- [9] Marsudi, 2016, *Teori Graf*. Malang: UB Press.
- [10] Saputra, H. K., & Aprillian, L. V., 2020, *Belajar Cepat Metode Saw*. Bandung: Kreatif Industri Nusantara.
- [11] Satria, D., 2016, *Infografi dan Peta Tematik Data Sosial Ekonomi*. Malang: UB Press.
- [12] Setyawan, F. E., & Supriyanto, S. (2019). *Manajemen Rumah Sakit*. Sidoarjo: Zifatama Jawa.
- [13] Sitorus, L., 2015, *Algoritma Dan Pemrograman*. Yogyakarta: ANDI.
- [14] Solichin, A., 2016, *Pemrograman Web dengan PHP dan MySQL*. Jakarta: Penerbit Budi Luhur.
- [15] Sugiran, Daulay, P., Zaman, B., Effendy, F., & Amalia, L. 2016, Evaluasi Tutor Online untuk Meningkatkan Kualitas Layanan Tutorial Tatap Muka pada Pendidikan Jarak Jauh. *Journal of Information System Engineering and Business Intelligence*, 4.
- [16] Sulistiono, H., 2018, *Coding Mudah dengan CodeIgniter, jQuery, Bootstrap, dan Datatable*. Jakarta: Elex Media Komputindo.
- [17] Wicaksono, S.R. 2017, *Rekayasa Perangkat Lunak*. Jakarta: Seribu Bintang.