

ANALISIS PENENTUAN JARAK TEMPAT TAMBAL BAN TERDEKAT MENGUNAKAN METODE DIJKSTRA

Cornelesz Famor Barros Moniyung¹, Bonie Emphy Giri², James Adam Seo³, Ryan Peterzon

Hadjon⁴

Program Studi Teknologi Informasi Universitas Citra Bangsa^{1,2,3,4}

Email : barrosfamor@gmail.com¹, angibonie@gmail.com², james.adam@ucb.ac.id³,
ryanphd@ucb.ac.id⁴

Abstrak

Penelitian ini berjudul Analisis Penentuan Jarak Tempat Tambal Ban Terdekat Menggunakan Metode Dijkstra. Penelitian ini bertujuan untuk menganalisis penentuan jarak tempat tambal ban dengan memakai Metode Dijkstra dalam konteks penentuan jarak terdekat. Metode Dijkstra adalah algoritma yang sudah terbukti efektif dalam menentukan jarak terpendek antara dua titik dalam graf. Dalam studi ini, peneliti mengaplikasikan algoritma Dijkstra untuk menentukan jarak terdekat ke lokasi tambal ban, yang akan membantu pengendara menjumpai tempat tambal ban dengan efisien. Kata kunci dalam studi ini adalah tambal ban, jarak terdekat, dan algoritma Dijkstra. Peneliti melakukan analisis komprehensif pada hasil penelitian ini dan memberikan rekomendasi untuk pengembangan aplikasi praktis yang bisa membantu pengguna jalan dalam situasi darurat. Hasil dari penelitian ini diharapkan bisa meningkatkan efisiensi pencarian tempat tambal ban dan berkontribusi pada keselamatan dan kenyamanan berkendara.

Kata Kunci : Metode Dijkstra, Jarak Terdekat, Rute Terpendek

I. PENDAHULUAN

Saat ini, dunia sedang berkembang untuk menjadi lebih maju, dimana kemajuan itu tidak lepas dari peran inovasi yang semakin berkembang pula. Perkembangan ini juga berdampak positif pada inovasi yang mendorong manusia untuk membuat sesuatu yang bermanfaat baik untuk diri sendiri ataupun orang lain dengan tujuan untuk mempermudah manusia dalam merampungkan aktivitas sehingga pekerjaan manusia lebih cepat dan efektif dalam mendapat data. Saat ini, dunia sedang berkembang untuk menjadi lebih maju, dimana kemajuan itu tidak lepas dari peran inovasi yang semakin berkembang pula. Perkembangan ini juga berdampak positif pada inovasi yang mendorong manusia untuk membuat sesuatu yang bermanfaat baik untuk diri sendiri ataupun orang lain dengan tujuan untuk mempermudah manusia dalam merampungkan aktivitas sehingga pekerjaan manusia lebih cepat dan efektif dalam mendapat data. Kemajuan inovatif saat ini, terlebih telepon seluler, bisa dipakai sebagai cara untuk bekerja pada layanan terbuka, termasuk klien penasihat transportasi yang memiliki pilihan untuk mendapatkan data lalu lintas secara efektif. Aplikasi panduan pencarian lokasi diperlukan sebab dalam keseharian hidup masyarakat sering berpindah dari satu tempat ke tempat lain dengan mempertimbangkan produktivitas, waktu dan

biaya, sehingga diperlukan ketelitian dalam menentukan jalur yang paling singkat.

Kendaraan merupakan suatu alat transportasi yang dipakai untuk sampai pada suatu daerah tujuan. Kendaraan memungkinkan klien menghemat investasi tambahan yang signifikan untuk membantu aktivitas manusia. Tiap-tiap kendaraan memiliki satu bagian penting, terlebih ban. Ban memegang peranan penting sebab kemampuannya sebagai jaringan pendukung bobot kendaraan secara emosional, menjaga kesehatan kecepatan kendaraan, mengendalikan pijakan dan perlambatan, mengendalikan arah kendaraan dan juga bisa meredam getaran. Sifat ban juga harus menjadi perhatian utama bagi pengemudi untuk menghindari tumpahan ban. Ban pecah bisa disebabkan oleh berbagai sebab, misalnya ban luar yang aus, kondisi ban yang tidak baik, dan ban bocor akibat benda tajam. Saat ini, kita sering menjumpai toko perbaikan ban, bahkan di jalan kecil sekalipun. Kendaraan bermotor merupakan instrumen yang paling dibutuhkan untuk keperluan transportasi. Kemajuan zaman yang pesat membuat kendaraan bermotor menjadi sangat penting sebagai moda transportasi. Kendaraan mekanis menyisihkan waktu beberapa menit dan hemat energi sebab dibuat untuk membantu aktivitas manusia.

Problematika yang dihadapi pengemudi dalam mencari bengkel mekanik ban adalah harus terus menerus menelusuri jalan tanpa mengetahui di mana pengemudi bisa menjumpai bengkel itu. Pengemudi juga harus

bertanya pada tiap-tiap penghuni lingkungan yang mereka temui, yang akan menghasilkan proses pencarian yang lebih lama, sehingga mengurangi kegagalan waktu. Hambatan yang dihadapi pengemudi dalam mencari bengkel ban adalah harus terus menerus menelusuri jalan tanpa menyadarinya. di mana pengemudi bisa menjumpai bengkel mekanik. Pengemudi juga harus bertanya pada tiap-tiap penghuni lingkungan yang mereka temui, yang akan menghasilkan proses pencarian yang lebih lama, sehingga mengurangi kegagalan waktu.

Melihat problematika yang dihadapi maka perlu dibuat suatu kerangka media data untuk membantu pencarian lokasi bengkel ban mobil. Dalam studi ini dipakai perhitungan Dijkstra, teknik ini dipilih sebab bisa menjumpai jalan terdekat dengan menghitung semua beban terkecil yang mungkin untuk tiap-tiap titik. Sehubungan dengan hal itu, penghitungan ini dilakukan berlandaskan jarak perjalanan terpendek. Media data itu bisa berupa inovasi yang mengikuti perkembangan inovasi data, inovasi yang tepat untuk dimanfaatkan adalah inovasi Worldwide Situating Framework (GPS), salah satunya adalah telepon seluler dengan kerangka kerja Android. Untuk mempelajari bengkel ban diperlukan inovasi yang bisa menyimpan dan informasi bengkel ban yang sesuai. Rumusan masalah penelitian ini yaitu bagaimana pemanfaatan perhitungan Dijkstra dalam menghitung semua beban terkecil yang mungkin terjadi dari tiap-tiap titik area perbaikan ban dengan memakai inovasi (GPS)? Penelitian ini memakai metode Dijkstra penelitian ini mengambil sampel dari 15 tambal ban di Kupang NTT dengan memakai *Google Maps*. Tujuan dari pemeriksaan ini, yakni: Membantu masyarakat luas mendapatkan data pasti perihal area reparasi ban di web. Membedah kecukupan pelibatan perhitungan Dijkstra dalam membantu masyarakat umum dengan menentukan jalur yang paling singkat pada bidang perbaikan ban. Membedakan unsur-unsur yang mempengaruhi kemajuan kerangka perhitungan Dijkstra dalam menentukan arah yang paling terbatas. Manfaatnya memberikan usulan yang lebih tepat pada masyarakat umum dalam mengambil pilihan mencari tempat perbaikan ban terdekat. Dengan perhitungan Dijkstra dan pemanfaatan inovasi Worldwide Situating Framework (GPS) diyakini bisa membantu kesejahteraan masyarakat. Dan memberikan pemahaman yang lebih baik perihal pemanfaatan perhitungan Dijkstra dalam pengamatan lapangan melalui pemanfaatan teknologi Worldwide Situating Framework (GPS).

II. DASAR TEORI

Konsep Dasar Algoritma Dijkstra

2.1 Algoritma Dijkstra

Aulia, R., Syahputra, E. R., & Dafitri, H. (2015) menuturkan bahwa Perhitungan Dijkstra adalah suatu

teknik untuk menentukan cara paling terbatas yang dipakai untuk grafik berbobot dan terkoordinasi. Dijkstra adalah nama pencipta perhitungan ini, terlebih Edsger Wybe Dijkstra. Perhitungan Dijkstra bisa menelusuri cara paling singkat dalam sejumlah tahap. Perhitungan ini memakai standar yang ketat, hal ini bertujuan pada tiap-tiap langkah akan dipilih sisi yang bobotnya paling kecil dan tetap dalam susunan yang ditetapkan. Andrew Goldberg seorang ilmuwan di Microsoft Exploration Silicon Valley menuturkan bahwa para ahli terus berfokus pada masalah pencarian jarak terpendek sebab sejumlah alasan. Memecahkan masalah dengan penerapan yang luas adalah cara tercepat. Seperti pengarah organisasi, permainan, rencana, sirkuit, dan perhitungan perencanaan Dijkstra bekerja dengan membuat satu hub ideal di tiap-tiap langkah. Jadi pada langkah ke-n, pada dasarnya ada n hub yang cara paling terbatasnya kita ketahui. Perhitungan Dijkstra diimplementasikan untuk melacak jalur terpendek dalam grafik terkoordinasi.

Robet, Yoshida Sary, Dearfan, (2021) menuturkan bahwa Perhitungan Dijkstra yang dijumpai oleh peneliti PC asal Belanda, Edsger Dijkstra pada tahun 1959, merupakan pencarian diagram perhitungan Dijkstra yang menangani masalah jalan terpendek dari satu sumber dengan nilai jalan yang tidak merugikan, dan menghasilkan pohon jalan yang paling terbatas. Perhitungan ini banyak dipakai dalam pencarian mata kuliah. Untuk simpul sumber (hub) khusus dalam diagram, penghitungan menghasilkan jalur dengan biaya paling sedikit (misalnya, jalur paling singkat) antara simpul itu dan simpul lainnya. Untuk mencari cara yang paling singkat, perhitungan

Dijkstra memiliki sejumlah rangkaian yang masuk akal, pengelompokannya, yakni:

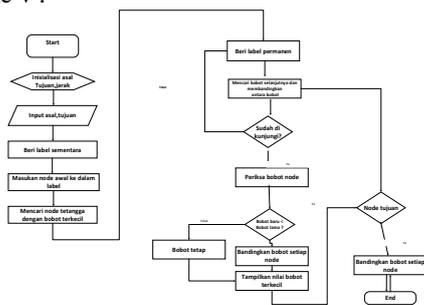
1. tiap-tiap titik harus memiliki nilai bobot (jarak) dari titik lainnya. Node awal harus diset dengan nilai 0, dan node lainnya (yang belum terisi) harus bernilai tak terhingga.
2. Node awal diset sebagai "Node Keberangkatan" dan semua node lainnya diset sebagai "belum Tersentuh".
3. Tentukan seberapa jauh jarak node tetangga yang belum dijelajahi dari titik awal. Misalnya jarak titik keberangkatan A ke B adalah 6 dan jarak B ke node C adalah 2, maka jarak melalui B ke C adalah $6+2=8$. Apabila jarak ini kurang dari jarak yang tercatat sebelumnya, hapus data lama dan simpan kembali data jarak itu memakai jarak baru.
4. Sesudah semua jarak ke node terdekat telah diperhitungkan, tentukan node yang disentuh sebagai "node terjamah". Jarak yang disimpan adalah jarak akhir dengan bobot paling kecil, dan node yang disentuh tidak akan pernah diperiksa lagi.
5. Lanjutkan dengan kembali ke langkah 3 dan tentukan atau set "Node belum terjamah"

yang paling dekat (dari node keberangkatan) sebagai "Node Keberangkatan" berikutnya.

Perhitungan Dijkstra merupakan perhitungan yang dipakai untuk menentukan pemisahan paling singkat dimulai dari satu titik kemudian ke titik berikutnya dalam suatu diagram berbobot. Nilai bobot tiap-tiap sisi pada diagram mewakili jarak antar node. Bobot ≥ 0 memperlihatkan bahwa bobot harus positif. Edger Wybe Dijkstra-lah yang menemukan perhitungan Dijkstra. Perhitungan Dijkstra disebut juga dengan perhitungan rakus, yakni perhitungan yang menangani masalah dengan menelusuri nilai yang paling ekstrim. Perhitungan Dijkstra dimulai dari satu titik dan dilanjutkan dari titik itu ke titik terdekat, titik berikutnya, dan seterusnya dengan maksud agar jarak terbatas bisa dicari.

Mayoritas masalah cara singkat bisa diselesaikan dengan memakai perhitungan Dijkstra, metode pencarian grafik yang pertama kali diusulkan pada tahun 1959. Lee (2006) dengan menyesuaikan perhitungan Dijkstra dan memasukkannya ke dalam kerangka arah kendaraan. Kerangka arah kendaraan dipisahkan menjadi dua cara, yakni perhitungan cara terpendek dan cara tercepat. Di lain sisi penghitungan cara tersingkat berpusat pada batas panjang lintasan dan menghitung lintasan yang paling terbatas, di lain sisi penghitungan cara tercepat berpusat pada lintasan dengan waktu tempuh paling sedikit.

Perhitungan Dijkstra pada umumnya dipakai untuk membuat aplikasi berbasis geolokasi, salah satu pemeriksaannya mengimplementasikan Perhitungan Dijkstra dalam eksplorasinya dalam mencari semako dengan memberikan cara yang paling terbatas pada klien, perbaikan aplikasi berbasis client-waiter, diperkenalkanlah aplikasi klien pada meja Android di lain sisi aplikasi pelayan dibuat memakai bahasa pemrograman C# dengan Microsoft Visual Studio Climate. Perhitungan Dijkstra memakai cara yang cermat untuk menangani problematika yang paling terbatas pada satu sumber, lebih dari satu kali memilih dari simpul-simpul yang tidak dipilih simpul v yang paling dekat dengan sumber, lalu, pada saat itu, melaporkan pemisahan terpendek yang sebenarnya dari s ke v .



Gambar 2.1. Flowchart Algoritma Dijkstra Pada *flowchart* diatas bisa dijelaskan proses algoritma Dijkstra, yakni:

1. Masukan: Graf berbobot

2. Proses:

- a) Inisialisasi vertex
- b) Inisialisasi jarak antar vertex
- c) Tentukan vertex awal (A) dan vertex tujuan(O)
- d) Beri label permanen = 0 ke vertex (A) dan label sementara = ∞ ke vertex lainnya.
- e) Untuk tiap-tiap vertex V yang belum mendapatkan label permanen, mendapat label sementara = $\min \{ \text{label lama } V \text{ (label lama } V+D) \}$.
- f) Cari nilai minimum di antara semua vertex yang masih berlabel sementara.
- g) Jadikan vertex minimum yang berlabel sementara menjadi vertex dengan label permanen, jika lebih dari satu vertex pilih sembarang.
- h) Ulangi langkah 5 sampai 7 hingga vertex tujuan mendapat label permanen.
- i) impan hasil perhitungan tampilan hasil perhitungan.

Sesudah dilakukan perancangan system, tahapan selanjutnya adalah implementasi dimana pada tahapan ini dilakukan pengkodean pada daftar.

2.2 Lintasan Terpendek

Astrid Noviriandini, Maryanah Safitri. (2017) menuturkan bahwa Persoalan menjumpai cara paling singkat dalam suatu bagan merupakan persoalan perbaikan. Bagan yang dipakai dalam mencari jalan terpendek adalah diagram berbobot, yakni bagan yang tiap-tiap sisinya diberi nilai atau bobot. Beban pada tepi grafik bisa mengkomunikasikan jarak antar wilayah perkotaan, waktu penyampaian pesan, biaya perbaikan, dll. Kecurigaan yang kami gunakan di sini adalah bahwa semua beban bertanda positif. "Yang paling singkat" tidak senantiasa harus dimaknai sebagai panjang dasar, sebab nilai "yang paling terbatas" berfluktuasi tergantung pada masalah umum yang ingin ditangani. Bagaimanapun, secara keseluruhan "paling singkat" bermakna membatasi beratnya suatu jalan dalam diagram.

Ada beberapa macam persoalan lintasan terpendek, antara lain:

1. Lintasan terpendek antara dua buah simpul khusus.
2. Lintasan terpendek antara semua pasangan simpul.
3. Lintasan terpendek dari simpul khusus ke semua simpul yang lain.
4. Lintasan terpendek antara dua buah

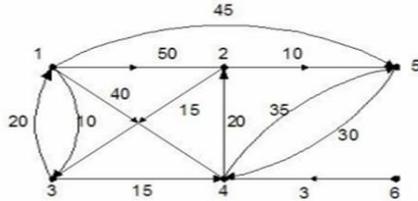
Kata paling singkat dalam masalah jalan menyiratkan metode yang berkaitan dengan pembatasan beban pada suatu jalan. Beberapa jenis masalah cara paling singkat memuat:

- a. Lintasan terpendek antara dua buah simpul.
- b. Lintasan terpendek antara semua pasangan simpul.
- c. Lintasan terpendek dari simpul khusus kesemua simpul yang lain.

d. Lintasan terpendek antara dua buah simpul yang melalui sejumlah simpul khusus .

Dalam pemecahan persoalan perihal graf berbobot $G = (V,E)$

dan sebuah simpul a . Penentuan lintasan terpendek dari a ke tiap-tiap simpul lainnya di G . Asumsi yang kita buat adalah bahwa semua sisi berbobot positif.



Gambar 2.2 Graf contoh persoalan lintasan terpendek

Dalam suatu organisasi, metode yang paling populer untuk melacak jalur terpendek adalah metode yang menempuh jarak paling sedikit atau biaya paling sedikit antara pusat awal dan tujuan. Ada dua jenis siklus dalam proses komputasi kursus terpendek: proses pemeriksaan hub dan sistem penamaan. Strategi penandaan adalah teknik untuk memberikan bukti yang bisa dikenali pada tiap-tiap pusat dalam organisasi. Dalam perhitungan perkiraan jalur paling singkat, ada 3 tanda data yang diawasi untuk tiap-tiap hub I dalam sistem penamaan, yakni: tanda jarak $d(I)$, simpul induk $I(I)$, dan status hub $S(I)$. Perhitungan yang bisa dipakai untuk menjumpai cara yang paling terbatas sudah dieksplorasi secara luas. Sejumlah perhitungan yang bisa dipakai untuk merampungkan jaminan cara tersingkat adalah perhitungan Dijkstra, perhitungan Bellman-Portage, perhitungan pencarian A^* , dan perhitungan Floyd-Warshall.

2.3 Bengkel

Bengkel adalah wirausaha skala kecil hingga menengah yang menyediakan jasa reparasi mobil dan sepeda motor. Yoga, J. A. (2013).

2.4 Tambal Ban

Memperbaiki adalah memperbaiki sesuatu (rumah, ban, dan sebagainya) yang tidak lengkap (hanya bagian yang rusak saja). Dalam deskripsi tambal ban bermakna suatu benda berbentuk bulat yang terbuat dari bahan elastis yang dipasang melingkar pada sebuah roda (sepeda, kendaraan, dan lain sebagainya). Ban memuat 3 macam, diantaranya:

1. Ban dalam adalah ban yang berada di dalam lebih tipis, tempat udara dipompakan.
2. Ban luar adalah ban yang berada di luar, tebal dan keras sebagai lapis penutup ban dalam.
3. Ban tubless adalah ban yang dirancang tanpa memakai ban dalam.

2.7 Algoritma

Muhammad Khoiruddin Harahap, Nurul Khairina, (2017) menuturkan bahwa Kata perhitungan diambil dari kata algoritma yang berasal dari nama seorang peneliti ternama asal Timur Tengah, terlebih Abu Ja'Far Muhammad Ibnu Musa Al Khuwarizmi dan diartikulasikan oleh orang barat dengan sebutan

Algorisme. Terlebih lagi deskripsi perhitungan sendiri adalah sekumpulan pedoman atau perintah yang dibuat secara jelas dan efisien yang dilandasi oleh suksesi yang sah untuk mengurus suatu problematika.

Komponen Algoritma

Perhitungan memiliki 5 bagian aturan yaitu batasan, kepastian, informasi, hasil dan kecukupan. Dalam merencanakan suatu perhitungan ada tiga bagian yang harus diperlukan, yakni:

- a. Komponen Masukan (*Input*)
- b. Komponen Keluaran (*Output*)
- c. Komponen Proses (*Processing*)

1. Notasi (Tata cara penulisan)

Tidak jarang, Algoritma Penulisan algoritma sukar untuk dipahami, begitu pula maksudnya. Tetapi, menulis algoritma juga merupakan sebuah tantangan. Notasi algoritmik bisa dipakai untuk memudahkan sejumlah hal. Suatu algoritma (desain penyelesaian masalah) yang ditulis dalam notasi disebut notasi algoritma (tata cara penulisan). Tiga kategori notasi algoritma yang sering terlihat, yakni:

- a. diagram alir (*flowchart*)
- b. *deskriptif*
- c. *pseudo-code*.

2.5 Aturan Penulisan Teks Algoritma

Perhitungan adalah gambaran sejumlah langkah untuk mengatasi suatu problematika. Penggambaran ini bisa ditulis dalam jenis dokumentasi apa pun, yang penting tidak sulit untuk dibaca dan dipahami. Tidak ada dokumentasi standar untuk menyusun teks perhitungan. Tiap-tiap individu bisa membuat aturan dan perhitungannya sendiri-sendiri. Meskipun demikian, agar dokumentasi perhitungan bisa secara efektif diubah menjadi dokumentasi bahasa pemrograman, dokumentasi perhitungan sebaiknya dibandingkan dengan dokumentasi bahasa pemrograman secara keseluruhan. Teks perhitungan didokumentasikan menjadi tiga bagian (blok): judul perhitungan (header), pengumuman, dan penggambaran.

2.6 Global Positioning System (GPS)

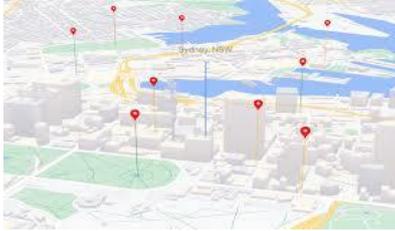


Gambar 2.3 Global Positioning System (GPS)

Syaiful Ahdana, Setiawansyah, (2015) menuturkan bahwa *Global Positioning System (GPS)* adalah kerangka rute satelit yang memuat 24 satelit fungsional dan 3 satelit penguat. Ke-24 satelit itu mengelilingi bumi dengan jarak 20.200 km dan musim orbit 12 jam, sekaligus mengirimkan sinyal berita

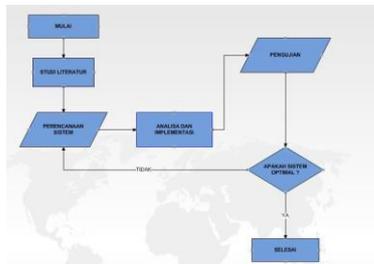
gelombang radio yang diterima oleh penerima. Di lain sisi, alat untuk mendapatkan sinyal satelit yang bisa dimanfaatkan oleh klien kebanyakan disebut dengan GPS Tracker atau GPS. Selanjutnya dengan memakai alat ini pengguna bisa mengikuti lokasi kendaraan, armada atau kendaraan secara terus menerus.

2.6 API Google Maps



Gambar 2.4 API Google Maps

Windi Eka Yulia R., Dwiretno Istiadi, Abdul Roqib. (2015) menuturkan bahwa Antarmuka Pemrograman *Google Guides* merupakan bantuan yang diberikan Google pada klien untuk memakai *Google Guides* dalam membuat aplikasi. Antarmuka Pemrograman Panduan Google memberikan sejumlah sorotan untuk mengontrol panduan, dan menambahkan konten melalui berbagai jenis administrasi, serta



memungkinkan klien untuk membuat aplikasi proyek di situs mereka.

Syaiful Ahdana, Setiawansyah, (2015). Google map adalah alat perencanaan yang awalnya dikembangkan oleh Lars dan Jens Rasmussen, dua bersaudara asal Denmark. Lalu, pada bulan Oktober 2004, Google membelinya. Google Guides adalah sistem perencanaan tingkat tinggi yang memungkinkan kantor lain menggabungkan perspektif citra satelit, meningkatkan profil, meninjau lokasi, mencari alamat, dan menentukan arah mengemudi.

2.7 MySQL (My Structured Query Language)

Enterprise, J. (2014) menuturkan bahwa MySQL adalah Sistem Manajemen Basis Data Relasional (RDBMS) populer yang cepat dan gampang dipakai untuk sejumlah tujuan. MySQL AB Swedia mengembangkan MySQL. Sejumlah hal berikut berkontribusi pada meningkatnya popularitas MySQL:

1. Berlisensi open source, sehingga bisa memakainya secara gratis.
2. Merupakan program yang powerful dan menyediakan fitur yang lengkap.
3. Menggunakan bentuk standar data SQL.

4. Dapat bekerja dengan banyak sistem operasi dan dengan bahasa pemrograman seperti PHP, PERL, C, C++, JAVA, dan lain-lain.
5. Bekerja dengan cepat dan baik, bahkan dengan data set yang banyak.
6. Sangat gampang dipakai dengan PHP untuk pengembangan aplikasi web.

Mendukung banyak database, sampai 50 juta baris atau lebih dalam suatu table (2014:1-3).

2.8 Hypertext Preprocessor (PHP)

Solichin, A. (2016) menuturkan bahwa *Pemrograman PHP dan MySQL untuk web*. Penerbit Budi Luhur. Pengembang web menciptakan PHP yang merupakan bahasa pemrograman berbasis web untuk pengembang web lainnya. Pengembang perangkat lunak dan anggota tim Apache Rasmus Lerdorf awalnya menciptakan PHP yang diterbitkan pada akhir tahun 1994. Tujuan awal pengembangan PHP adalah untuk melacak hanya pengunjung ke situs web pribadi Rasmus Lerdorf. Bahasa pemrograman web PHP dikembangkan khusus untuk membuat aplikasi web. Selain bisa diakses secara bebas, PHP juga gampang dipelajari oleh semua orang. (2016:11:23).

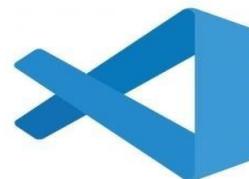
2.9 XAMPP

Riyadli menuturkan bahwa “Xampp merupakan *software* yang dirancang untuk membangun server lokal yang bisa diakses melalui web browser dan juga membuat database memakai PhpMyAdmin. Riyadli, H., Arliyana, A., & Saputra, F. E. (2020)

2.10 Database

Database adalah kumpulan kelompok data terkait yang sudah disusun untuk memfasilitasi pemakaian kembali secara cepat dan sederhana. Database secara luas dapat diklasifikasikan menjadi dua jenis, yakni database relasional dan database flat. Sebab database relasional lebih masuk akal dibandingkan database datar, maka database ini direkomendasikan. Database bisa dianggap sebagai susunan data yang dimungkinkan oleh komputer sehingga bisa diakses dengan cepat dan gampang selaras dengan apa yang dituturkan oleh Jantce TJ Sitingjak, D. D., Maman, ., & Suwita, J. (2020).

2.11 Visual Studio Code



Gambar 2.5 Logo Visual Studio Code

Pengembangan aplikasi bisa dilakukan dengan perangkat lunak *Microsoft Visual Studio*, yang bisa dipakai untuk membuat aplikasi konsol, Windows, dan Web serta aplikasi bisnis dan pribadi

serta komponen aplikasi.

III. METODE PENELITIAN

Adapun 3.1 Jenis Penelitian

Penelitian kuantitatif adalah jenis penelitian yang dipakai dalam studi ini, yakni informasi spesifik yang didapat dari pengujian penelitian dibedah berlandaskan metodologi yang dipakai kemudian diuraikan dalam struktur yang jelas.

3.2 Model Penelitian

Study Literatur

Study literatur adalah tahap menghimpun dan mengagregasi berbagai sumber tertulis, misalnya buku, makalah logika dan eksplorasi komparatif yang sudah ada sebelumnya untuk mengetahui apa yang sudah dilakukan orang lain, lalu melihat derajat perbedaannya dengan penelitian itu.

a. Perencanaan Sistem

Metode Algoritma Dijkstra akan dipakai untuk menganalisa Tempat Tambal Ban dan menentukan rute tercepat pada skripsi ini. Data yang dimasukkan berupa jarak antar Tempat Tambal Ban seperti terlihat pada *blok diagram* pada Gambar 3.1. Untuk mempermudah perhitungan, data masukan akan direpresentasikan sebagai persamaan matematika. Algoritma Dijkstra dipakai bersama dengan bahasa pemrograman web untuk melakukan perhitungan. Keluaran sistem berupa rute terpendek yang bisa ditempuh pengemudi untuk sampai ke tempat tambal ban.

b. Implementasi *System*

Sesudah analisis dan perancangan dilakukan, aplikasi antarmuka diberi kode dan diimplementasikan selama tahap implementasi selaras dengan analisis dan perancangan. Implementasi ini memuat:

1. Dengan memakai Google Maps, algoritma Dijkstra diimplementasikan untuk menentukan rute tercepat dalam skenario dunia nyata.
2. Implementasi antarmuka adalah cara untuk memeriksa apakah aplikasi yang dikembangkan mengikuti perancangan atau tidak.

3.3 Metode Pengumpulan Data

Metode yang dipakai untuk menghimpun data atau informasi terkait untuk eksplorasi atau studi. Teknik pengumpulan informasi yang berbeda bisa dipakai tergantung pada tujuan pemeriksaan, jenis informasi yang diperlukan, sumber informasi yang bisa diakses, dan lokasi eksplorasi.

a. Studi Kasus

Informasi atau data yang bisa mendukung penelitian ini dalam buku, jurnal, artikel, dan sumber online bisa didapat dengan melakukan studi pustaka.

b. Wawancara

Salah satu teknik pengumpulan data yang dipakai untuk mendapatkan informasi langsung dari sumbernya adalah wawancara. Wawancara dipakai untuk mendapatkan informasi yang lebih mendalam dari responden yang jumlahnya terbatas.

c. Observasi

Salah satu strategi atau metode pengumpulan data adalah observasi yang melibatkan pengamatan pada aktivitas yang sedang berlangsung. Observasi bisa dilakukan dengan atau tanpa keterlibatan partisipan. Pengamat mengambil bagian dalam kegiatan yang sedang berlangsung saat mereka mengimplementasikan observasi partisipatif. Di lain sisi, observasi non-partisipatif bermakna pengamat tidak terlibat dalam tindakan tersebut. Fungsi pengamat hanya untuk mengamati kegiatan.

d. Dokumentasi

Buku, peraturan, laporan kegiatan, gambar, dokumenter, dan data terkait lainnya adalah sejumlah data dalam dokumentasi yang dihimpun secara langsung dari lokasi penelitian.

3.4 Pengelolaan Data

Proses penyusunan kumpulan data dikenal sebagai pengolahan data, dan melibatkan sejumlah langkah, yakni memilih data yang tepat, pembersihkannya, mencari tahu atribut apa yang diperlukan, dan kemudian mentransformasikan data.

flowchart Algoritma Dijkstra, yakni:

Mulai Inisialisasi titik awal dan jarak tiap titik

1. Pemberian Label Memberikan label sementara untuk jarak antar titik
2. Menentukan jarak.
3. Jarak dihitung dan dibandingkan dengan maksud agar jarak minimum bisa dijumpai, lalu diberikan label sementara.
4. Label permanen
5. Sesudah ditentukan jarak minimum, maka diberikan label permanen pada titik itu untuk melangkah ke titik selanjutnya, ulangi langkah itu hingga sampai pada tujuan
6. Pencarian titik terpendek
7. Sesudah dibandingkan jarak menuju node tujuan, maka dijumpai node-node yang harus dilewati dalam menentukan rute terpendek
8. Selesai

3.5 Analisa *system*

Pencarian jalur terdekat ini akan dijalankan dengan kerangka memakai strategi Dijkstra yang memanfaatkan teknologi GPS untuk menentukan area dasar pengemudi memakai perhitungan Dijkstra untuk menentukan jalur terdekat. Organisasi jalan dijumpai dan proses estimasi Perhitungan Dijkstra diselesaikan oleh server, sehingga hasil perhitungan akan dikirimkan dari gadget pengemudi sebagai jalur tercepat menuju tujuan.

Aplikasi ini mengimplementasikan panduan dari Google (*Google Guides*) sebagai petunjuk arah area. Sebagai arahan dasar bagi klien, arahan area tujuan dan fasilitas jalan. Semua informasi arah diubah selaras dengan fasilitas yang ada di *Google Maps*. Aplikasi ini berencana untuk memberdayakan pengemudi untuk mendapatkan data perihal rute terdekat dari titik lokasi, Pemeriksaan dan konfigurasi dilakukan sesudah tahap penyajian masalah selesai, dimana dalam penyelidikan dan perencanaan ini ada sejumlah fokus terlebih uraian estimasi Perhitungan Dijkstra, rencana model aplikasi dan rencana pengumpulan data seperti yang diuraikan di bawah ini:

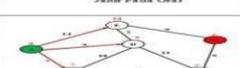
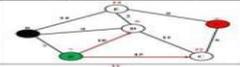
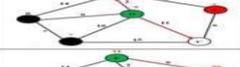
Berikut adalah sejumlah langkah yang dipaparkan pada deskripsi perhitungan *algoritma*,
1. Fase-fase yang terlibat dalam menentukan jalur terpendek, beserta contoh rute dalam bentuk grafik. Berikut adalah sejumlah langkah yang dipaparkan:

- a) Inisialisasi Node
- b) Membuat tabel nilai untuk nilai yang ada pada masing masing node
- c) Memberikan nilai bobot awal pada tiap-tiap node berlandaskan nilai jarak
- d) Rute tercepat bisa dicari dengan memakai rumus *Algoritma Dijkstra*
- e) Rute sukses ditemukan
- f) *Prototype* Antarmuka Aplikasi dirancang dengan maksud agar pengguna bisa merasakan kemudahan dalam memakai aplikasi
- g) Perancangan database aplikasi yang memuat data koordinat lokasi dan koordinat rute yang bisa ditempuh

3.6 Gambaran Umum Sistem

Ada juga gambaran umum system yang akan di implementasikan:

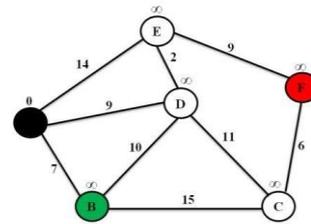
1. Admin masuk untuk login
2. Admin bisa menambah data Tambal Ban baru
3. Admin bisa menghapus data Tambal Ban

Langkah	Vertex Sumber	Jarak Vertex Lain	Jalur Pada Graf
1.	A	A → B = 7 A → D = 9 A → E = 14	
2.	B	B → C = 10 B → D = 22 B → E = 17	
3.	D	D → C = 10 D → E = 20 D → F = 11	
4.	E	E → F = 20	

ada kesalahan pada data Tambal Ban

3.7 Contoh pengimplementasian *Algoritma Dijkstra*

Gambar 3.4 memperlihatkan contoh jalur terpendek yang ditentukan dengan implementasi *Algoritma Dijkstra* dari node A ke F dan Tabel 3.2 memperlihatkan penyelesaiannya



Gambar 3.4 Graf penentuan jalur terpendek

Tabel Penyelesaian
graph metode *Algoritma Dijkstra*

Cara kerja perhitungan Dijkstra memakai metodologi rakus, dimana hub yang telah dipilih dihubungkan ke hub lain yang belum dipilih pada tiap-tiap langkahnya oleh edge dengan bobot paling rendah. Diperlukan batasan untuk perhitungan Dijkstra sebagai awal dan tujuan. Batasan ini memuat lintasan sebagai suatu sisi (vertex) atau simpul dalam struktur jamak yang perlu dipertimbangkan. Masing-masing sisi lintasan ini merupakan pasangan simpul (u,v) yang mengalami hubungan dari simpul u ke simpul v. Susunan semua sisinya kita namakan E. Beban semua sisi ditentukan dengan kemampuan: Persamaan Perhitungan Bobot dalam Perhitungan Dijkstra.

$$w : E \rightarrow [0, \infty]$$

Sehubungan dengan hal itu, jarak non-negatif antara simpul v dan simpul u adalah $w(u,v)$. Jarak antara dua *vertices* atau total jarak semua sisi dalam jalur bisa dianggap sebagai biaya (*cost*) sebuah sisi.

IV. HASIL DAN PEMBAHASAN

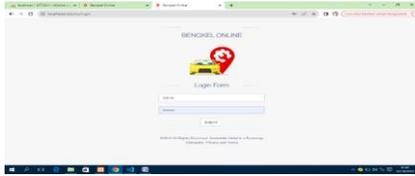
4.1 Implementasi Sistem

Pada fase ini kita akan membahas perihal kerangka data pemindaian kursus jarak ke tempat perbaikan ban terdekat memakai Dijkstra, dengan memanfaatkan teknik Dijkstra Anda bisa melacak jalan dengan beban dasar antara satu titik dengan titik lainnya.

4.2 Antar Muka Admin

a. Halaman Login

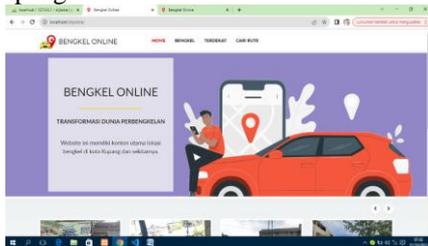
Halaman antar muka login merupakan halaman untuk admin mengakses dashboard *admin* dimana seluruh data *user* bisa dikelola oleh *admin*, tetapi pada antar muka ini lebih dahulu di input data email dan *password* yang sudah daftar di *database* kemudian *login*. Tampilan *login admin*.



Gambar Halaman Login Admin

b. Halaman Dashboard

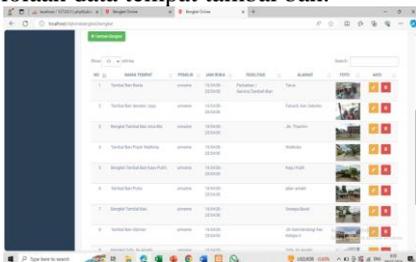
Halaman dashboard memperlihatkan informasi yang sudah ditempatkan oleh *admin*, data yang dimaksud yaitu memuat data admin dan data tempat tambal ban yang ada di Kota Kupang. Ini adalah pusat informasi yang memberikan gambaran holistic perihal pengolahan dan perkembangan tempat tambal ban di Kota Kupang.



Gambar Dashboard Admin,

c. Menu Tempat Tambal Ban

Halaman menu tempat tambal ban ada tabel yang memperlihatkan nama tempat tambal ban dan gambar. Halaman tempat tambal ban memberikan *admin* akses penuh untuk mengelola data tempat tambal ban. Dengan fitur pencarian, tombol tambah, edit dan hapus admin bisa dengan gampang menjalankan tugas pengelolaan data tempat tambal ban.

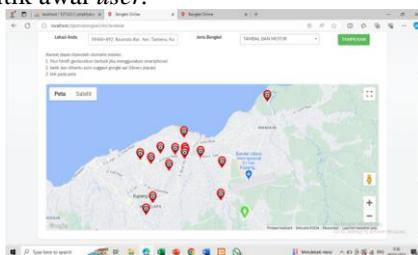


Gambar 4.3 Menu Admin

4.3 Antar Muka User

a. Halaman Peta

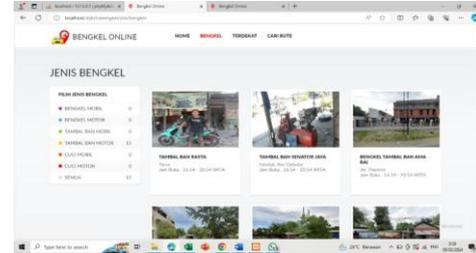
Halaman Peta *user*, pengguna dengan hak akses *user* bisa melihat informasi penting terkait sistem yaitu *user* bisa melihat lokasi tempat tambal ban terdekat di Kota Kupang. Misalnya, pada peta lokasi tempat tambal ban, jika *user* memasukkan lokasi titik awal makan akan muncul lokasi-lokasi terdekat di sekitaran titik awal *user*.



Gambar Halaman Peta

a. Halaman Tempat Tambal Ban

Halaman tempat tambal ban memperlihatkan informasi terkait tempat tambal ban yang terletak di Kota Kupang. Tambal Ban bisa menjumpai deskripsi rinci perihal tiap-tiap tempat tambal ban, memberikan mendalam perihal pesona dan daya tarik para pelanggan.



Gambar Dashboard

4.2 Pengujian Sistem

Pengujian sistem dilakukan untuk mengidentifikasi dan memperbaiki potensi kesalahan atau bug yang mungkin ada dalam sistem yang sudah dibuat/sebelum diperkenalkan pada pengguna. Hasil pengujian sistem ini juga dipakai sebagai dasar untuk mengambil keputusan terkait dengan pengembangan atau peningkatan lebih lanjut pada sistem yang sudah ada. Teknik pengujian sistem dipakai adalah teknik *black box* yaitu teknik pengujian perangkat lunak yang dilakukan tanpa mengetahui detail atau keadaan internal sistem yang diuji.

Tujuan pengujian *black box* adalah dihasilkan untuk pengujian operasi dan efisiensi sistem dengan mengambil *input* dan *output* dihasilkan, terlepas dari bagaimana sistem menanganinya. Tabel berikut memperlihatkan gambaran metode pengujian *black box* pada *website*.

4.3 Analisis Kelebihan dan Kekurangan Sistem

Tiap-tiap sistem yang dikembangkan pasti akan memiliki keunggulan ataupunkele mahan dari berbagai sudut pandang. Tetapi, hal ini dianggap sangat pentingdalam mengevaluasi sistem itu agar bisa diperbaiki dalam dan ditingkatkan kearah yang lebih baik dimasa depan.

Dibawah ini ada sejumlah kelebihan dari sistem, yakni:

Kelebihan Sistem

Berlandaskan hasil pengujian fungsi dari sistem yang dibuat kelebihan darisistem ini adalah sistem bisa membantu pengguna dalam mencari lokasi tempat tambal ban terdekat, sehingga bisa membantu para tempat tambal ban dalam melakukan perjalanan menuju tempat tambal ban yang di inginkan dengan gampang dan efisien.

Kekurangan Sistem

1. Sistem ini hanya melakukan proses perhitungan dijkstra
2. Sistem ini masih berbasis *website*.

4.4 Perhitungan Jalur Terdekat Algoritma Dijkstra

Pengguna harus terlebih dahulu menginput lokasi awal untuk menentukan rute, lalu harus mencari atau mencari tempat tambal ban terdekat dengan lokasi

awal. Jika demikian, sistem akan menghitung jalur menuju tujuan memakai metode Dijkstra.

4.5 Menentukan Titik Awal dan Akhir

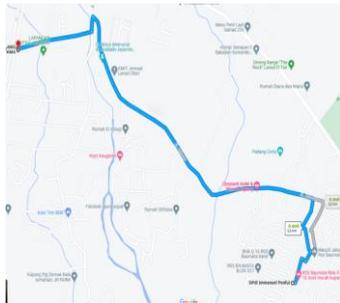
Titik awal adalah tempat yang menjadi awal perjalanan, di lain sisi titik tujuan adalah tempat tercapainya tujuan. Pada studi rumah dengan kordinat -10.189376733694733, 123.66585615851325 adalah titik awal yang diambil.

4.6 Menentukan Jalur

Menentukan tiap jalur dari titik awal ke titik tujuan

1) Bengkel A

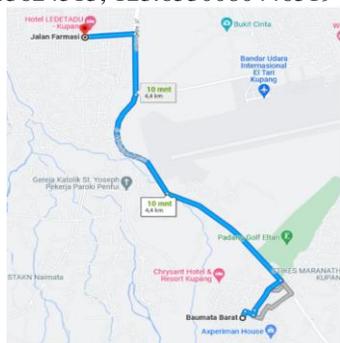
UMKM Gilang Pratama
10.180028, 123.652361



Gambar Bengkel A

2) Bengkel B

Tambal Ban Om Semy
10.16303303624513, 123.6530080446519



Gambar Bengkel B

3) Bengkel C

Tambal Ban Sebelum Jurusan Kesehatan Lingkungan (Liliba)

10.157715902035587, 123.64092371303003



Gambar Bengkel C

4) Bengkel D

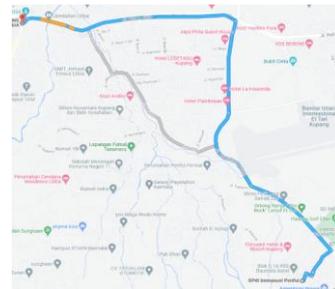
Tambal Ban Bundaran PU
-10.1559135783656, 123.63221751928798



Gambar Bengkel D

5) Bengkel E

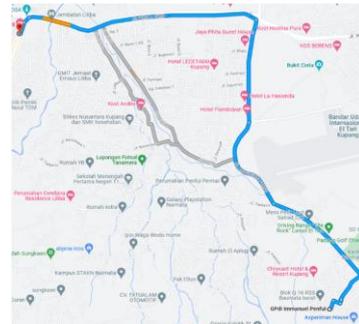
Tambal Ban Bundaran PU (Sesudah Hotel Amaris)
10.157046, 123.632625



Gambar 4.11 Bengkel E

6) Bengkel F

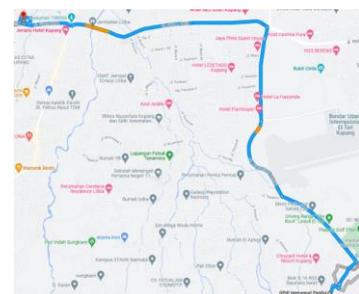
Tambal Ban (Sebelum Hotel Amaris)
10.158279882153042, 123.6321140756880



Gambar 4.12 Bengkel F

7) Bengkel G

Tambal Ban Idaman (Jln Samratulangi Kec Kelapa 5)
10.156050, 123.626765
No HP: 0823-4297-7937

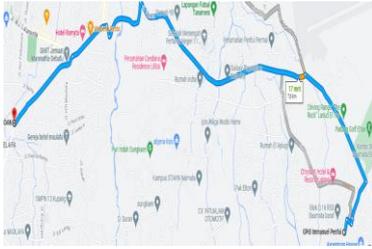


Gambar 4.13 Bengkel G

8) Bengkel H

Bengkel Tofa Jln Amabi

-10.181323717427992, 123.61426350380381



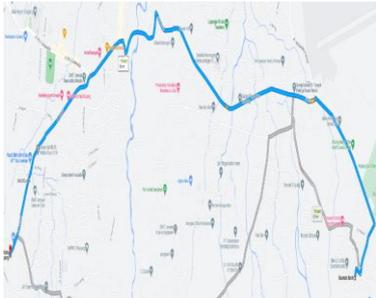
Gambar 4.14 Bengkel H

9) Bengkel I
Bengkel Tambal (Ban Oesapa Barat)
-10.147279, 123.631658



Gambar 4.15 Bengkel I

10) Bengkel J
Tambal Ban Putra
-10.187493199543344, 123.61044563360736



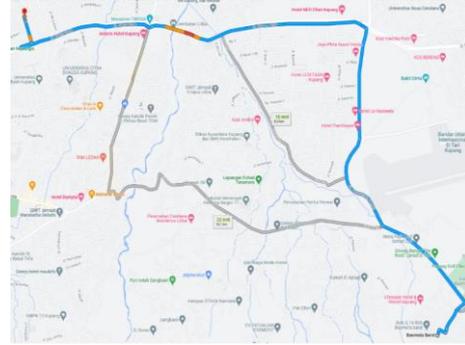
Gambar 4.16 Bengkel J

11) Bengkel K
Bengkel Tambal (Ban Kayu Putih)
-10.16461783,123.6232728



Gambar 4.17 Bengkel K

12) Bengkel L
Tambal Ban Pojok (Walikota)
-10.155672, 123.618432



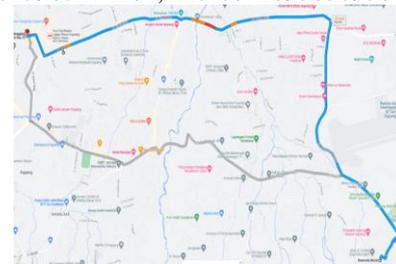
Gambar 4.18 Bengkel L

13) Bengkel M
Bengkel Tambal Ban Ama Rai (Jln. Thamrin)
-10.161093487398377, 123.6132090842845



Gambar 4.19 Bengkel M

14) Bengkel N
Tambal Ban Senator Jaya (Fatululi, Kec Oebobo)
-10.158813983274184, 123.6071893830938



Gambar 4.20 Bengkel N

15) Bengkel O
Tambal Ban Rasta Tarus
-10.128605729585415, 123.67841972721054

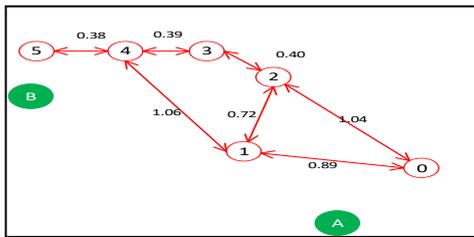


Gambar 4.21 Bengkel O

4.8 Contoh Perhitungan atau Penerapan Algoritma Dijkstra

Pada langkah ini peneliti memakai metode Algoritma Dijkstra untuk menghitung iterasi dari titik awal sampai ke tujuan. Jalur atau rute terpendek dari titik awal ke tiap-tiap titik tujuan menjadi pertimbangan pada tiap-tiap iterasi Algoritma Dijkstra.

- Contoh 1
1. Titik awal ditentukan adalah Baumata Bar., Kec. Taebenu, Kabupaten Kupang, Nusa Tenggara Tim., Indonesia. Koordinat 10.189412124454131, 123.66579936611267.
 2. Titik akhir ditentukan Tambal Ban UMKM Gilang Pratama. Koordinat -10.180028, 123.652361.
 3. Data node, path, dan jarak yang dipakai



Gambar Graph diagram contoh 1

A: Lokasi Pengguna

B: Lokasi Bengkel Tujuan 0,1,2,3,4,5: Node (persimpangan jalan)

2. Node yang terdekat dengan titik awal (berlandaskan rumus jarak garis lurus) adalah node 0 (0,56 km). Node yang terdekat dengan titik akhir (berlandaskan rumus jarak garis lurus) adalah node 5 (0,34 km).
3. Menentukan jalur terpendek memakai algoritma *dijkstra*.

Langkah algoritma *dijkstra*:

- a) Tentukan titik awal.
 - b) Pilih titik terdekat dengan bobot terkecil yang bisa dijadikan sebagai maksimum lokal.
 - c) Kita jadikan maksimum lokal itu sebagai titik awal di iterasi selanjutnya.
 - d) Jika bobot suatu titik di iterasi sebelumnya lebih besar daripada iterasi yang sedang berjalan, maka bobot digantikan oleh bobot terbaru yang lebih kecil.
 - e) Jika suatu iterasi dijumpai bobot minimal lebih dari satu, maka penentuan maksimum lokalnya dengan memilih salah satunya.
 - f) Kembali ke langkah 2 hingga semua titik yang bisa dilalui menjadi maksimum lokal.
4. Titik awal ditentukan adalah Baumata Bar., Kec. Taebenu, Kabupaten Kupang, Nusa Tenggara Tim., Indonesia dengan koordinat lokasi -10.189412124454131, 123.66579936611267.
 5. Titik akhir ditentukan Tambal Ban Om Semy, Jalan W.J. Lalamentik No.95. Koordinat -10.16303304, 123.653008.
 6. Data node, path, dan jarak yang dipakai diambil dari basis data

A: Lokasi Pengguna

B: Lokasi Bengkel Tujuan 0,1,2,3,4,7,33,34,35: Node (persimpangan jalan)

7. Node yang terdekat dengan titik awal (berlandaskan rumus jarak garis lurus) adalah node 0 (0,56 km). Node yang terdekat dengan titik akhir (berlandaskan rumus jarak garis lurus) adalah node 33 (0,38 km).
8. Menentukan jalur terpendek memakai algoritma *dijkstra*.

Langkah algoritma *dijkstra*:

1. Tentukan titik awal.
2. Pilih titik terdekat dengan bobot terkecil yang bisa dijadikan sebagai maksimum lokal.
3. Kita jadikan maksimum lokal itu sebagai titik awal di iterasi selanjutnya.
4. Jika bobot suatu titik di iterasi sebelumnya lebih besar daripada iterasi yang sedang berjalan, maka bobot digantikan oleh bobot terbaru yang lebih kecil.
5. Jika suatu iterasi dijumpai bobot minimal lebih dari satu, maka penentuan maksimum lokalnya dengan memilih salah satunya.
6. Kembali ke langkah 2 hingga semua titik yang bisa dilalui menjadi maksimum lokal.

Tabel Matrix Algoritma *Dijkstra*

Iterasi	Dari ... Ke >>>						Jalur	Total Jarak
		1	2	3	4	5		
1	0	0,89	1,04	~	~	~	0-1	0,89
2	1	0,89	1,61	~	1,95	~	0-1-2	1,61
		0	1		1		0-2	1,04
3	2	0,89	1,04	1,44	~	~	0-2-3	1,44
		0	0	2				
4	3	0,89	1,04	1,44	1,83	~	0-2-3-4	1,83
		0	0	2	3			
5	4	0,89	1,04	1,44	1,83	2,21	0-2-3-4-5	2,21
		0	0	2	3	4		

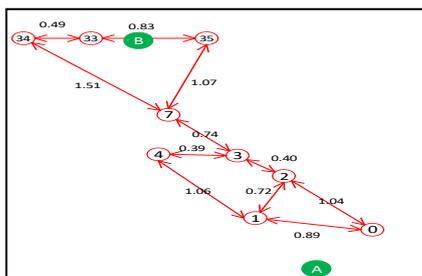
Tabel Keterangan Matrix *Dijkstra*

Iterasi	Path	Jarak	Keterangan
1	0-1 0-2	0-1 berjarak 0,89 0-2 berjarak 1,04	Maksimum lokal adalah titik 1 dengan total jarak 0,89
2	1-2 1-4	1-2 berjarak 0,72 1-4 berjarak 1,06	Maksimum lokal adalah titik 2 (jalur 0-1-2) dengan total jarak 1,61 Jika dibandingkan dengan jalur (0-2) dengan jarak 1,04 maka maksimum lokal ditetapkan titik 2 (jalur 0-2) dengan total jarak 1,04
3	2-3	2-3 berjarak 0,40	Maksimum lokal adalah titik 3 (jalur 0-2-3) dengan total jarak 1,44
4	3-4	3-4 berjarak 0,39	Maksimum lokal adalah titik 4 (jalur 0-2-3-4) dengan total jarak 1,83

5	4-5	4-5 berjarak 0,38	Maksimum lokal adalah titik 5 (jalur 0-2-3-4-5) dengan total jarak 2,21
---	-----	-------------------	---

Contoh 2

1. Titik awal ditentukan adalah Baumata Bar., Kec. Taebenu, Kabupaten Kupang, Nusa Tenggara Tim., Indonesia dengan koordinat lokasi -10.189412124454131, 123.66579936611267.
2. Titik akhir ditentukan Tambal Ban Om Semy, Jalan W.J. Lalamentik No.95. Koordinat -10.16303304, 123.653008.
3. Data node, path, dan jarak yang dipakai diambil dari basis data



Gambar *Graph* Contoh 2

Gambar 4.23 Graph diagram contoh 2

A: Lokasi Pengguna

B: Lokasi Bengkel Tujuan

0,1,2,3,4,7,33,34,35: Node (persimpangan jalan)

- a. Node yang terdekat dengan titik awal (berlandaskan rumus jarak garis lurus) adalah node 0 (0,56 km). Node yang terdekat dengan titik akhir (berlandaskan rumus jarak garis lurus) adalah node 33 (0,38 km).

b. Menentukan jalur terpendek memakai algoritma *dijkstra*.

Langkah algoritma *dijkstra*:

1. Tentukan titik awal.
2. Pilih titik terdekat dengan bobot terkecil yang bisa dijadikan sebagai maksimum lokal.
3. Kita jadikan maksimum lokal itu sebagai titik awal di iterasi selanjutnya.
4. Jika bobot suatu titik di iterasi sebelumnya lebih besar daripada iterasi yang sedang berjalan, maka bobot digantikan oleh bobot terbaru yang lebih kecil.
5. Jika suatu iterasi dijumpai bobot minimal lebih dari satu, maka penentuan maksimum lokalnya dengan memilih salah satunya.
6. Kembali ke langkah 2 hingga semua titik yang bisa dilalui menjadi maksimum lokal.

Tabel Matrix Algoritma *Dijkstra*

Ite Ra si	Da ri 0 Ke >> >	1	2	3	4	7	33	34	35	Jal ur	Jara k

1	0	0,8 9	1,0 4							0-1	0,9
2	1	0,8 9	1,6 1			1,9 5				0-1-2	1,61
3	2	0,8 9	1,0 4	1,4 4						0-2	1,04
4	3	0,8 9	1,0 4	1,4 4	1,8 3	2,1 8				0-2-3	1,44
5	7	0,8 9	1,0 4	1,4 4	1,8 3	2,1 8				0-2-3-4	1,83
6	34	0,8 9	1,0 4	1,4 4	1,8 3	2,1 8	4,1 9			0-2-3-7	2,18
7	35	0,8 9	1,0 4	1,4 4	1,8 3	2,1 8	4,0 8	3,2 5	7,7	0-2-3-35	3,25
		0	0	2	3	3	34				4,19
		0	0	2	3	3	35	7			4,08

Tabel Keterangan Matrix *Dijkstra*

Iterasi	Path	Jarak	Keterangan
1	0-1 0-2	0-1 berjarak 0,89 0-2 berjarak 1,04	Maksimum lokal adalah titik 1 dengan total jarak 0,89
2	1-2 1-4	1-2 berjarak 0,72 1-4 berjarak 1,06	Maksimum lokal adalah titik 2 (jalur 0-1-2) dengan total jarak 1,61 Jika dibandingkan dengan jalur (0-2) dengan jarak 1,04 maka maksimum lokal ditetapkan titik 2 (jalur 0-2) dengan total jarak 1,024
3	2-3	2-3 berjarak 0,40	Maksimum lokal adalah titik 3 (jalur 0-2-3) dengan total jarak 1,44
4	3-7	3-7 berjarak 0,74	Maksimum lokal adalah titik 7 (jalur 0-2-3-7) dengan total jarak 2,18
5	7-34 7-35	7-34 berjarak 1,51 7-35 berjarak 1,07	Maksimum lokal adalah titik 35 (jalur 0-2-3-7-35) dengan total jarak 3,25
6	35-33	35-33 berjarak 0,83	Jalur 0-2-3-7-35-33 memiliki total jarak 4,08
7	34-33	34-33 berjarak 0,49	Jalur 0-2-3-7-34-33 memiliki total jarak 4,19

4.9 Menarik Kesimpulan

Sesudah menjalankan studi, Peneliti menarik kesimpulan dari contoh 1 dalam penentuan rute terpendek perjalanan dari titik awal menuju semua titik tujuan. Rute terpendek dari titik awal Rumah Baumata Bar., Kec. Taebenu, Kabupaten Kupang, Nusa Tenggara Tim. Indonesia. Koordinat -10.189412124454131, 123.66579936611267. ke Tambal Ban UMKM Gilang Pratama. Koordinat -10.180028, 123.652361 rute menuju bengkel UMKM

Gilang Pratama dengan jalur terpendek dari titik 0 ke titik 5 berlandaskan perhitungan dijkstra adalah jalur 0-2-3-4-5. Jika digabungkan dengan lokasi pengguna aplikasi dan lokasi bengkel maka jalur akhir adalah lokasi pengguna-0-2-3-4-5-lokasi bengkel.

Di lain sisi kesimpulan dari contoh 2 dalam penentuan rute terpendek perjalanan dari titik awal menuju semua titik tujuan, Rute terpendek dari titik awal Rumah Baumata Bar., Kec. Taebenu, Kabupaten Kupang, Nusa Tenggara Tim., Indonesia. Koordinat -10.189412124454131, 123.66579936611267. Ke Tambal Ban Om Semy, Jalan W.J. Lalamentik No.95. Koordinat -10.16303304, 123.653008. Jadi jalur terpendek dari titik 0 ke titik 33 berlandaskan perhitungan dijkstra adalah jalur 0-2-3-7-35-33, Jika digabungkan dengan lokasi pengguna aplikasi dan lokasi bengkel maka jalur akhir adalah lokasi pengguna-0-2-3-7-35-33-lokasi bengkel

V. KESIMPULAN

5.1 Kesimpulan

Berlandaskan studi yang sudah pernah dijalankan pada algoritma Dijkstra dalam penentuan jalur terpendek (shortest path), memperlihatkan bahwa jalur terpendek antara dua titik dicari dengan memakai algoritma Dijkstra dengan memilih jalur yang memiliki bobot terendah. Algoritma Dijkstra bekerja dengan terlebih dahulu mengidentifikasi titik awal, lalu melakukan pembobotan jarak antara node pertama dan terdekat secara iteratif. Hal ini memungkinkan algoritme untuk memajukan pencarian secara bertahap dari satu titik ke titik berikutnya dan sebaliknya. Peneliti memakai sampel uji dengan lokasi tujuan yang berbeda berlandaskan uji coba algoritma Dijkstra, dimana hasil uji di aplikasi bisa memperlihatkan tujuannya.

Jalur dari titik koorinat posisi pengguna. Sebab tiap graf pada algoritma Dijkstra dipilih dengan bobot minimum dengan cara suatu node terpilih dihubungkan dengan node lain yang tidak terpilih, maka algoritma itu efektif dalam mencari jalur terpendek,

5.2 Saran

1. Untuk kedepannya, tingkat interaksi pengguna atau layout menu bisa ditingkatkan untuk mengembangkan aplikasi.
2. Metode pencarian jalur terpendek yang lebih efektif diharapkan bisa diciptakan dalam pengembangan aplikasi ini.

VI. REFERENSI

Astrid Noviriandini, Maryanah Safitri. (2017). "Implementasi Algoritma Dijkstra Untuk Menentukan Jalur Terpendek Wilayah Pisangan Dan Kampus Nusa Mandiri Tangerang". Jurnal Pilar Nusa Mandiri Vol. 13 (2).

Aulia, R., Syahputra, E. R., & Dafitri, H. (2015). Sistem Pencarian Rumah Sakit Terdekat Menggunakan Algoritma Dijkstra Berbasis Android (Studi Kasus: rumah Sakit di Kota Medan). In *Medan. Prosiding: SNASTIKOM (Seminar nasional teknologi informasi & komunikasi)*, jilid (Vol. 1, pp. 150-155).

Enterprise, J. (2014). *MySQL untuk pemula*. Elex Media Komputindo.

Jantce TJ Sijinjak, D. D., Maman, ., & Suwita, J. (2020). Analisa Dan Perancangan Sistem Informasi Administrasi Kursus Bahasa Inggris Pada Intensive English Course Di Ciledug Tangerang. *Insan Pembangunan Sistem Informasi Dan Komputer (IPSIKOM)*, 8(1).

<https://doi.org/10.58217/ipsikom.v8i1.164>

Sutriyono, (2019). "Implementasi Algoritma Dijkstra Pada Aplikasi Pencarian Bengkel Tambal Ban Terdekat Di Kota Medan Berbasis Android". *Jurnal Majalah Ilmiah Informasi dan Teknologi Ilmiah (INTI)*.

Muhammad Khoiruddin Harahap, Nurul Khairina, (2017). "Pencarian Jalur Terpendek dengan Algoritma Dijkstra". *Jurnal & Penelitian Teknik Informatika Volume 2 (2)*

Riyadli, H., Arliyana, A., & Saputra, F. E. (2020). Rancang Bangun Sistem Informasi Keuangan Berbasis WEB. *Jurnal Sains Komputer Dan Teknologi Informasi*, 3(1), 98–103. <https://doi.org/10.33084/jsakti.v3i1.1770>

Robet, Yoshida Sary, Dearfan, (2021). "Aplikasi Pencarian Bengkel Tambal Ban Dan SPBU Terdekat Di Kota Medan Menggunakan Metode Dijkstra Dan Haversine Berbasis Android". <http://ejournal.stmik-time.ac.id>.

Solichin, A. (2016). *Pemrograman web dengan PHP dan MySQL*. Penerbit Budi Luhur.

Syaiful Ahdana, Setiawansyah, (2015). "Pengembangan Sistem Informasi Geografis Untuk Pendonor Darah dengan Algoritma Dijkstra berbasis Android". *Jurnal Sains Dan Informatika Research Of Science And Informatic.V6.I2*.

Windi Eka Yulia R., Dwiretno Istiadi, Abdul Roqib. (2015). "Pencarian SPBU Terdekat Dan Penentuan Jarak Terpendek Menggunakan Algoritma Dijkstra (Studi Kasus Di Kabupaten Jember)". *Jurnal Nasional Teknik Elektro*.

Yoga, J. A. (2013). Perancangan Sistem Inventory Sparepart Motor Pada CV. Surya Jaya Jepara. *Universitas Dian Nuswantoro*.