

RANCANG BANGUN APLIKASI PENENTUAN LOKASI TAMBAL BAN MOTOR DI KABUPATEN BLITAR MENGGUNAKAN METODE WATERFALL BERBASIS ANDROID

Muhammad Ibnu Zurbaninanto, Abdi Pandu Kusuma, Wahyu Dwi Puspitasari

Program Studi Teknik Informatika, Universitas Islam Balitar

Dibnuzx4@gmail.com

ABSTRAK

Aplikasi penentuan lokasi tambal ban motor di kabupaten blitar menggunakan metode pengembangan yaitu metode waterfall. Perancangan aplikasi ini dibuat untuk kebutuhan masyarakat akan solusi yang praktis dan efisien dalam mencari lokasi-lokasi dari tambal ban motor. Aplikasi ini juga mengintegrasikan teknologi GPS untuk menentukan lokasi pengguna dan menghitung rute terpendek ke tambal ban terdekat. Selanjutnya, dilakukan perancangan antarmuka aplikasi menggunakan framework Flutter dan IDE Visual Studio Code serta Android Studio. Pengujian white box dilakukan untuk mengukur kompleksitas kode program, yang menghasilkan nilai cyclomatic complexity sebesar 1 dengan terdapat 1 jalur independen, termasuk sebuah prosedur sederhana, dan telah teridentifikasi memiliki risiko rendah atau jumlah kesalahan yang kecil. Hasil dari pengujian kuesioner menunjukkan bahwa aplikasi ini mendapatkan penilaian sebesar 81.25% dari ahli IT dan 84.25% dari pengguna aplikasi. Hal ini menunjukkan bahwa aplikasi telah mendapatkan respon positif dari kedua kelompok pengguna, menandakan bahwa aplikasi ini dapat memberikan nilai tambah dan memenuhi kebutuhan pengguna dengan baik. Dengan demikian, aplikasi ini diharapkan dapat memberikan kontribusi yang signifikan dalam mempermudah pengguna dalam mencari lokasi tambal ban motor di Kabupaten Blitar.

Kata kunci: Aplikasi Android, Tambal Ban Motor, Metode Waterfall, GPS.

1. PENDAHULUAN

Pada era modern ini, peta telah mengalami perkembangan yang signifikan dalam beberapa tahun terakhir, terutama dengan munculnya teknologi yang memungkinkan penggunaan peta secara real-time. Ada banyak alasan mengapa era modern mengenai peta sangat penting, seperti mobilitas yang tinggi, penggunaan teknologi yang tinggi, perkembangan transportasi, perkembangan kota dan wilayah, dan bisnis yang terkait dengan peta.

Penggunaan teknologi yang tinggi, perkembangan transportasi, perkembangan kota dan wilayah, dan bisnis yang terkait dengan peta. Peta yang terbaru dan akurat sangat penting untuk membantu orang menemukan jalur terbaik dan mempermudah perjalanan mereka, serta membantu perusahaan menjalankan bisnis dengan efisien [1] Meningkatnya pembelian sepeda motor menjadi penyebab utama kemacetan lalu lintas. Peningkatan jumlah kendaraan bermotor tidak diimbangi dengan peningkatan kepadatan penduduk dan kapasitas jalan yang tersedia. Pada 2017, totalnya adalah 1,46 juta, tidak termasuk kendaraan bermotor lainnya [2] Perancangan aplikasi ini dibuat untuk kebutuhan masyarakat akan solusi yang praktis dan efisien dalam mencari lokasi-lokasi dari tambal ban motor. Adanya teknologi Sistem Informasi Geografis (SIG) dapat mendukung keperluan penyebaran suatu informasi baik dalam bentuk data atribut maupun peta, yang berguna untuk meningkatkan koordinasi dan keterpaduan suatu data tertentu [3]

2. TINJAUAN PUSTAKA

2.1. Rute Terdekat

Menurut Tumanggor [4] Rute terdekat merupakan usaha untuk mencari rute yang paling dekat dari awal hingga akhir dengan beban paling ringan atau sedikit dibandingkan dengan seluruh rute yang ada, atau suatu perjalanan yang akan dilalui ke arah-arah tertentu untuk menuju suatu tempat yang ingin dituju..

2.2. Sistem Informasi Geografis (SIG)

Menurut Afandi (2017) Sistem Informasi Geografis (SIG) atau Geographic Information System (GIS) adalah sistem informasi khusus yang mengelola data yang memiliki informasi spasial yaitu bereferensi keruangan. [5]

2.3. GPS (Global Positioning System)

Menurut Afandi (2017) GPS atau sistem pemosisi global merupakan sistem yang digunakan untuk menentukan posisi di permukaan bumi dengan sinkronisasi sinyal satelit. Dengan bantuan GPS seseorang dapat mengetahui posisi objek yang diinginkannya dengan bantuan perangkat yang memiliki sensor GPS di dalamnya.. [5]

2.4. Android

Menurut Fatkharrufiqi & Gata (2022) Android adalah sistem operasi untuk perangkat mobile berbasis Linux yang mencakup sistem operasi, middleware, dan aplikasi. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri. [6]

2.5. LBS (Location Base Service)

Menurut Syepanda dkk(2021) Location based services adalah layanan berbasis lokasi atau gambaran teknologi yang digunakan untuk menemukan lokasi perangkat yang pengguna gunakan.[7]

2.6. Waterfall

Menurut Qaedi dkk(2023) model air terjun atau yang sering disebut metode waterfall sering dinamakan siklus hidup, nama model ini sebenarnya adalah Linear Sequential Model dimana hal ini menggambarkan pendekatan yang sistematis dan juga berurutan pada pengembangan perangkat lunak, dimulai dengan spesifikasi kebutuhan pengguna lalu berlanjut melalui tahapan-tahapan perancangan, pembuatan kode program, pengujian, serta pemeliharaan.[8]

2.7. Whitebox Testing

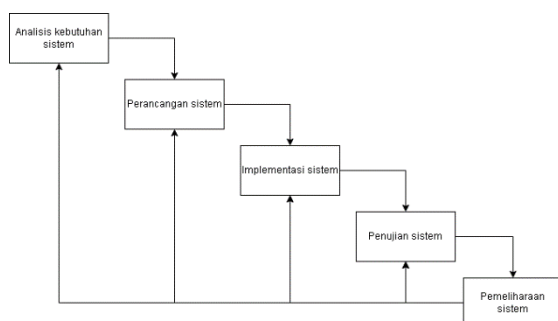
Menurut Harya Gusdevi (2022) White-box testing atau pengujian white-box dilakukan untuk menguji dan menganalisis kode program bilamana terjadi kesalahan atau tidak di sebut dengan pengujian white-box. Terdapat pendapat lain mengenai pengertian dari pengujian white box ini dilakukan dengan melihat pure kode tanpa melihat tampilan interface dari halaman aplikasi[9].

2.8. Maps API

Menurut Pratiwi dkk (2019) Google Maps merupakan suatu peta yang dapat dilihat dengan menggunakan suatu browser. Dengan menggunakan Mapbox API, kita dapat menghemat waktu dan biaya untuk membangun aplikasi peta digital yang handal, sehingga kita dapat fokus hanya pada data yang ditampilkan.[10]

3. METODE PENELITIAN

Pada pengambilan metode dari penelitian yang akan mencari lokasi tambal ban menggunakan metode pengembangan Waterfall. Penelitian ini akan dimulai dengan melakukan analisis terhadap masalah yang sering dihadapi oleh pengendara ketika mengalami kebocoran ban, seperti kesulitan menemukan lokasi tambal ban terdekat, informasi yang tidak akurat, dan tidak adanya informasi tentang jenis layanan yang disediakan oleh tambal ban.



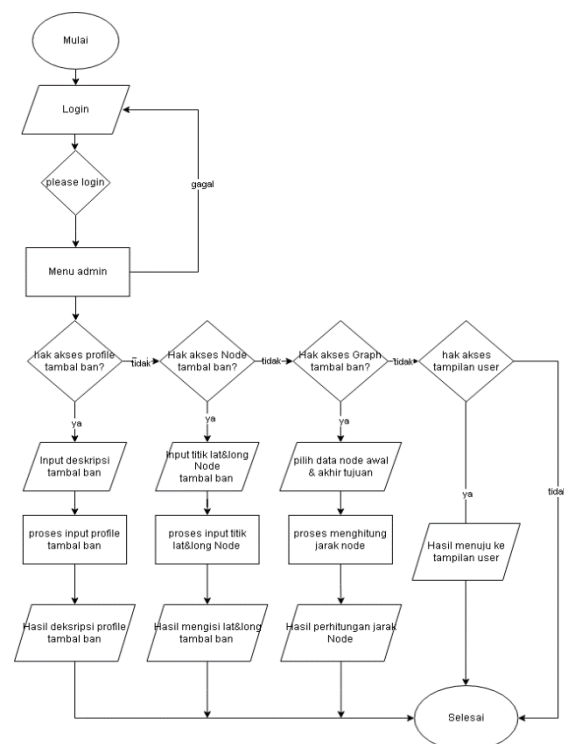
3.1. Analisis Kebutuhan Sistem

Melakukan analisis kebutuhan sistem aplikasi berdasarkan tujuan dan masalah yang ingin diselesaikan dalam penelitian. Beberapa analisis kebutuhan sistem yang perlu dilakukan antara lain: Analisis Kebutuhan fungsional, non-fungsional, pengguna, lingkungan, data, integrasi, dokumen, dan pengujian

3.2. Perancangan Sistem

Perancangan sistem pada flowchart adalah proses menggambarkan secara visual alur kerja atau prosedur dari suatu sistem atau proses. Ini berfungsi untuk memberikan representasi grafis yang jelas tentang bagaimana data atau informasi mengalir dan diolah dalam sistem. Selain itu, flowchart juga memungkinkan untuk menganalisis, mengidentifikasi, dan memecahkan potensi masalah atau kelemahan dalam proses tersebut sebelum implementasi dilakukan. Selain sebagai alat dokumentasi yang kuat, flowchart juga berperan dalam komunikasi efektif antar anggota tim pengembang serta sebagai panduan untuk mempersiapkan rencana implementasi. Dengan demikian, perancangan sistem pada flowchart memainkan peran kunci dalam memastikan sistem atau proses berjalan dengan efisien dan sesuai dengan kebutuhan pengguna.

Merancang sistem aplikasi berdasarkan analisis kebutuhan yang telah dilakukan, termasuk desain antarmuka pengguna, metode pengembangan waterfall digunakan untuk menentukan lokasi tambal ban, dan integrasi data geospasial untuk menampilkan peta lokasi tambal ban.



Gambar 1. Flowchart Admin

Pada Gambar 1 merupakan *flowchart* dari admin yang mempunyai hak akses dari aplikasi dan dapat

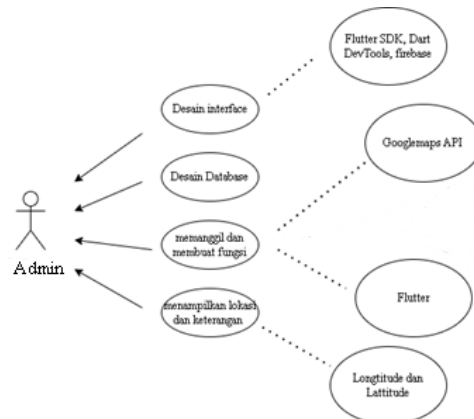
mengelola data lokasi, jarak, dan sistem. Langkah awal dari admin dengan memasukkan *username* dan *password* admin terlebih dahulu, jika terjadi kesalahan maka akan diminta kembali *login* sampai benar, sedangkan jika benar admin akan diarahkan pada *menu* admin yang mendapatkan hak akses dari sistem.

Pada menu admin akan memiliki akses untuk menambah yang pertama yaitu, *profile* tambal ban yang berisikan deskripsi dari tambal ban serta lokasinya, kedua yaitu menambahkan titik node dari tambal ban dengan lokasi yang sesuai *latitude* dan *longitude* dari data tambal ban yang sudah ditentukan oleh admin, ketiga yaitu memilih lokasi node agar djikstra dapat menghitung jarak terdekat agar dapat memilih waktu dan jarak yang lebih dekat dari tambal ban lain. Terakhir, ada fitur “tampilan *user*” yang berfungsi agar *admin* dapat mengakses tampilan *menu* user tanpa *logout* dan melihat hasil yang sudah diubah pada tampilan *admin*.



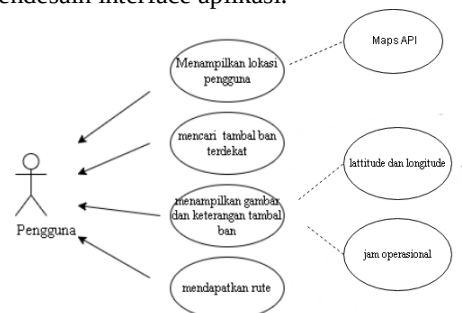
Gambar 2. Flowchart Sistem

Pada gambar 2 menunjukkan pengguna membuka aplikasi dan masuk ke halaman utama. Kedua, pengguna memasukkan lokasi awal dan lokasi tujuan. Ketiga, aplikasi akan menentukan lokasi tambal ban, deksripsi serta gambar tambal ban daln menampilkan lokasi tambal ban beserta jarak dan estimasi waktu tempuh. Keempat, pengguna dapat melihat detail lokasi tambal ban pada rute terdekat dan memilih untuk menavigasi ke lokasi tersebut menggunakan aplikasi peta yang tersedia pada perangkat pengguna. Kelima, pengguna dapat menyelesaikan navigasi dan kembali ke halaman utama untuk mencari lokasi tambal ban untuk lokasi berikutnya.



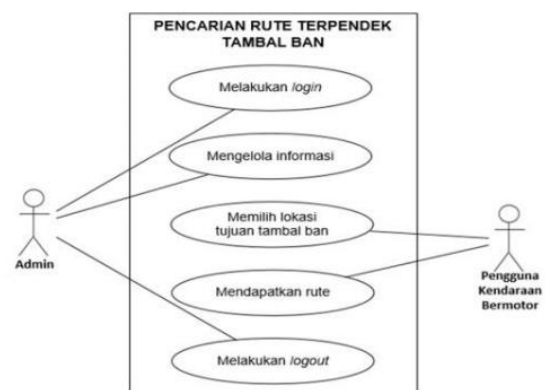
Gambar 3. Use Case Diagram Admin

Pada gambar 3 menjelaskan diagram admin yang mengelola aplikasi yang berkaitan dengan pemrograman *backend* maupun *frontend* yang bertugas untuk menambah fitur-fitur dari aplikasi dan mendesain interface aplikasi.



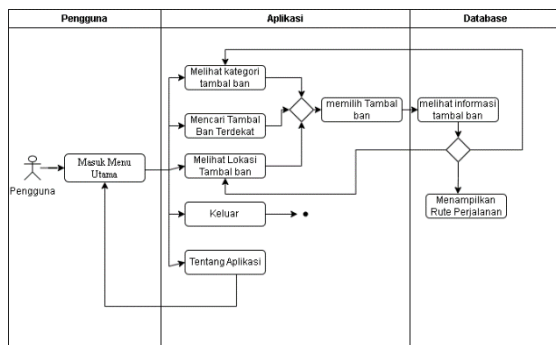
Gambar 4. Use Case Diagram Pengguna

Pada gambar 4 diagram pengguna yang hanya dapat melihat dan mencari lokasi-lokasi tambal ban yang sudah diinput oleh admin, pada tampilan lokasi terdapat latitude dan longitude dan juga jam operasional yang sudah disediakan admin



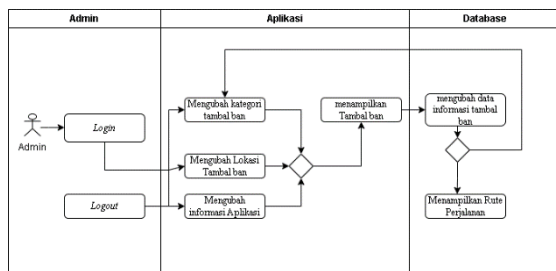
Gambar 5. Use Case Diagram Sistem

Pada gambar 5 terdapat admin yang dapat melakukan login dan informasi agar pengguna dapat melihat data yang sudah ditampilkan, untuk pengguna dapat memilih lokasi dari tambal ban agar mendapatkan rute menuju lokasi yang sudah dipilih.



Gambar 6. Activity Diagram Sistem

Pada gambar 6 diagram sistem pengguna terdapat menu-menu yang hanya dapat ditampilkan untuk pengguna dari aplikasi yaitu kategori tambal ban, lokasi tambal ban, dan juga informasi dari aplikasi, pada database terdapat data-data dari lokasi tambal ban yaitu gambar dari tambal ban, latitude dan longitude, dan juga jam operasional yang tidak bisa diolah oleh pengguna.



Gambar 7. Activity Diagram Admin

Pada gambar 7 diagram admin yang hanya dapat dikelola admin. Terdapat login dan logout agar informasi serta data lokasi tambal ban sesuai, pada database admin dapat mengubah apapun agar pengguna mendapatkan informasi yang sesuai.

4. HASIL DAN PEMBAHASAN

4.1. Database

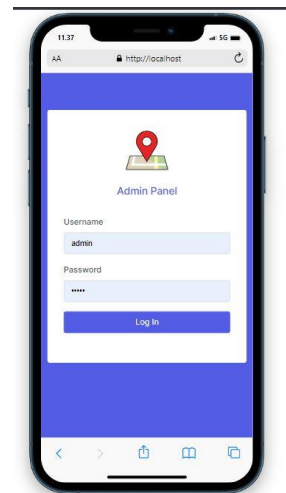
Tabel	Tindakan	Basis	Jenis	Penyortiran	Ukuran	Beban
graph	Struktur	Struktur	Struktur	Struktur	Struktur	Struktur
node	Struktur	Struktur	Struktur	Struktur	Struktur	Struktur
user	Struktur	Struktur	Struktur	Struktur	Struktur	Struktur

Gambar 8. Database

Pada Gambar 6 terdapat database dari aplikasi yang terdapat 3 poin yaitu table graph, node, dan user. Data diatas disesuaikan sesuai lokasi-lokasi yang sudah ditentukan dalam aplikasi agar dapat menemukan lokasi yang akan dituju.

4.2. Tampilan Aplikasi Admin

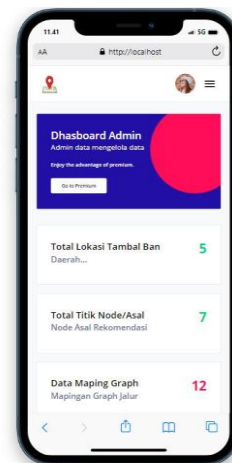
4.2.1. Halaman Login Admin



Gambar 1. Tampilan Halaman Login

Tampilan pada gambar 7 merupakan halaman login untuk admin agar membedakan antar tampilan user dan admin yang memiliki akses pada data aplikasi. Admin akan diberikan input username dan password jika admin sudah menginputkan dua hal tersebut tombol "Log In" akan ditekan agar admin dapat masuk ke halaman utama admin.

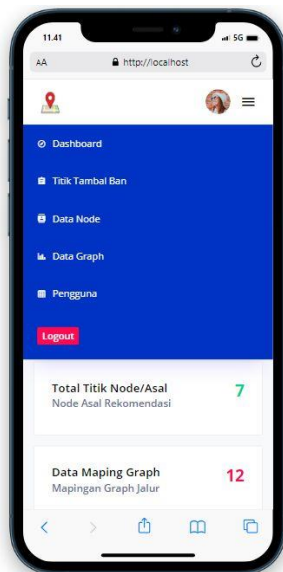
4.2.2. Halaman Dashboard Admin



Gambar 2. Tampilan Dashboard

Pada gambar 8 ditampilkan beberapa data dari tambal ban yang sudah diinputkan pada database. Admin mempunyai hak akses yang besar pada penginputan data tambal ban agar user dapat mencari tambal ban terdekat saat mengalami bocor pada ban motor milik user.

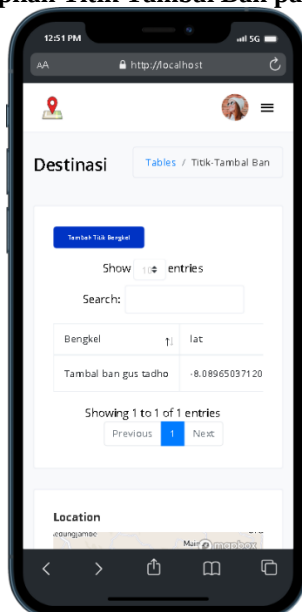
4.2.3. Halaman Menu Admin



Gambar 3. halaman Menu admin

Pada gambar 9 menampilkan tampilan dashboard dari Admin setelah berhasil login dengan akun khusus admin agar dapat mengolah data dari node, titik dari tambal ban, maupun data graph. Setelah Login Admin bisa keluar dari akun (Logout) dengan menekan tombol dashboard dan memilih tombol Logout berwarna merah yang berada pada urutan terbawah dashboard.

4.2.4. Tampilan Titik Tambal Ban pada Admin

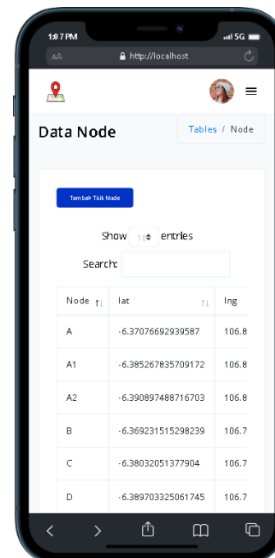


Gambar 4. Tambah titik tambal ban

Pada Gambar 10 diatas menampilkan hak Admin yang dapat menambahkan lokasi Tambal ban sesuai yang diinginkan, maupun hal yang perlu diisi pada tambah tambal ban adalah keterangan dari tambal ban

seperti nama tambal ban, jam operasional, latitude dan longitude, dan gambar dari lokasi tambal ban.

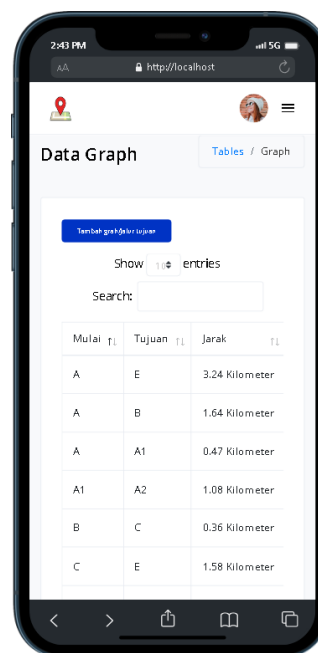
4.2.5. Tampilan Tambah Data Node Pada Admin



Gambar 5. Tambah Data Node

Pada Gambar 11 Data Node disini berfungsi sebagai rute awal dari lokasi yang diinputkan atau data yang diisi secara manual untuk menentukan perjalanan menuju tambal ban rute yang akan dituju. Pada tombol Tambah titik Node terdapat Nama dari node serta latitude dan longitude dari lokasi awal yang sudah ditentukan.

4.2.6. Tampilan Tambah Graph Pada Admin



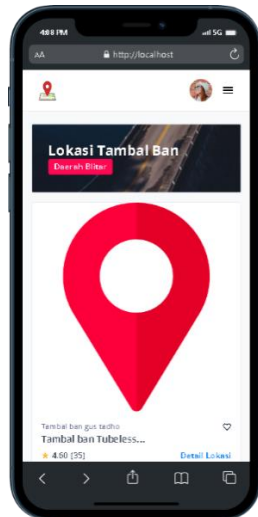
Gambar 6. Tambah Data Graph

Pada tampilan 13 atau graph merupakan bagian yang dipakai pada pencarian rute lokasi, Pada tombol

Graph jalur tujuan inputan jika di tekan akan muncul tampilan dimana admin dapat memilih tempat awal dan akhir tujuan sistem akan otomatis akan menampilkan output-nya yaitu jarak dari posisi awal menuju tempat dari tujuan yang sudah dipilih.

4.3. Tampilan Aplikasi User

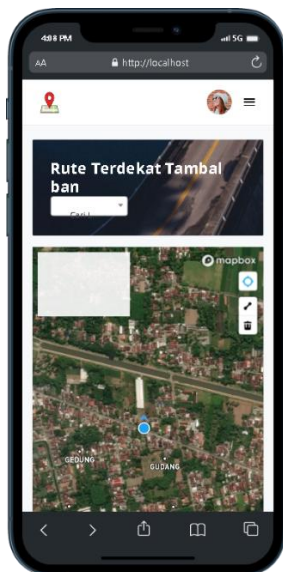
4.3.1. Tampilan Menu Tambal Ban pada User



Gambar 7. Menu Tambal Ban pada User

Pada gambar 13 menampilkan profil-profil dari lokasi tambal ban yang sudah diinputkan pada halaman tambal ban di sistem *admin*, jika pada teks-teks detail lokasi – lokasi ditekan profil tambal ban yang sudah diinputkan berisikan deskripsi yang diinputkan secara manual dan tempat lokasi dari tambal ban dari *maps*.

4.3.2. Pencarian Pada Lokasi User

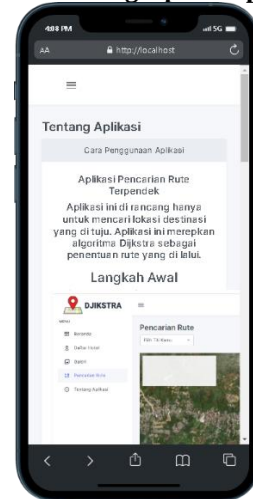


Gambar 8. Nilai Alternatif

Pada Gambar 14 merupakan sistem utama saat pencarian lokasi tambal ban dari data-data yang sudah diinputkan oleh admin pada daerah tertentu. Pada

halaman ini juga bisa menampilkan lokasi *Real-Time User* jika menekan tombol pada kanan atas yang berbentuk bulat agar otomatis *maps* akan melacak lokasi dari user, sebelum itu *user* akan diminta menyetujui agar aplikasi dapat mengakses lokasi *user*

4.3.3. Tampilan Tentang Aplikasi pada User



Gambar 9. Tentang Aplikasi

Pada Gambar 4.8 menampilkan informasi tentang aplikasi serta menjelaskan hanya beberapa fungsi-fungsi penting yang digunakan pada user, Halaman ini juga menampilkan beberapa gambar untuk menunjukkan langkah-langkah pada aplikasinya.

4.4. Pengujian White Box Testing

Metode Pengujian *Whitebox Testing* adalah sebuah metode yang dipakai untuk menguji sebuah *software* tanpa harus memperhatikan detail dari *software*. Pengujian *whitebox* ini akan dilakukan oleh Penulis sebagai *developer* untuk aplikasi penentuan lokasi tambal ban motor. Berikut langkah langkah dari pengujian *Whitebox*:

4.4.1. Pemetaan Source Code

Pada pemetaan source code dalam pengujian *whitebox* adalah metode untuk mengukur sejauh mana kode sumber dari program telah dieksekusi selama pengujian. Ini memberikan gambaran tentang seberapa baik pengujian telah mencakup seluruh kode yang ada. Terdapat beberapa tipe pemetaan seperti Line Coverage, Branch Coverage, dan Path Coverage yang masing-masing fokus pada aspek tertentu dari kode. Fungsi utama dari pemetaan source code adalah untuk memastikan kualitas dan keandalan perangkat lunak. Hal ini membantu dalam mengidentifikasi titik lemah atau bug yang mungkin terlewatkan selama pengujian, meningkatkan kualitas kode, dan memastikan kepatuhan dengan persyaratan regulasi dan standar tertentu. Selain itu, pemetaan source code juga memudahkan pengembang dalam proses debugging dan memberikan gambaran visual tentang kemajuan pengujian.

4.4.2. Pembuatan Notasi Alir (Flowgraph)

Flowgraph, dalam konteks pengujian whitebox, adalah representasi grafis dari struktur alur program atau fungsi yang dieksekusi selama proses pengujian. Ini terdiri dari simpul-simpul yang mewakili blok-blok kode dan panah-panah yang menghubungkannya untuk menunjukkan alur eksekusi. Flowgraph membantu dalam memvisualisasikan bagaimana aliran kontrol program berpindah dari satu blok kode ke blok lainnya, sehingga memudahkan analisis terhadap kemungkinan jalur eksekusi. Dengan memahami flowgraph, tim pengujian dapat mengidentifikasi rute-rute yang belum tercakup dalam pengujian, memastikan bahwa setiap percabangan dan kemungkinan jalur eksekusi telah diuji, dan memperbaiki strategi pengujian untuk mencakup seluruh struktur program dengan lebih efektif.

4.4.3. Perhitungan Cyclomatic Complexity

Cyclomatic Complexity (CC) adalah metrik kuantitatif yang digunakan dalam pengujian whitebox untuk mengukur kompleksitas struktural dari kode sumber. Metrik ini membantu dalam menentukan seberapa banyak uji kasus yang diperlukan untuk mencakup semua jalur eksekusi yang berbeda dalam program. Secara singkat, CC dihitung berdasarkan jumlah simpul (P), jumlah sisi (E), dan jumlah komponen terhubung (C) dalam flowgraph dengan menggunakan rumus berikut:

$$CC = E - P + 2C$$

Di mana:

- E** : jumlah sisi (edges) dalam flowgraph.
- P** : jumlah simpul (nodes) dalam flowgraph.
- C** : jumlah komponen terhubung (connected components) dalam flowgraph.

Rumus ini memberikan nilai numerik yang mengindikasikan tingkat kompleksitas dari program. Semakin tinggi nilai CC, semakin kompleks struktur programnya, dan oleh karena itu, memerlukan pengujian lebih komprehensif.

4.4.4. Penentuan Independent Path

Pengujian jalur independen (*Independent Path*) merupakan salah satu teknik penting dalam *whitebox testing* karena dapat membantu mengidentifikasi masalah yang mungkin terjadi dalam kode program, seperti bug atau kesalahan logika. Dengan menguji setiap jalur independen, pengembang dapat memastikan bahwa setiap bagian kode berjalan dengan benar dan tidak ada kesalahan yang terlewatkan. Berikut dibawah ini merupakan pengujian jalur independen (*Independent Path*) pada aplikasi penentuan lokasi tambal ban.

4.5. Hasil Pengujian

4.5.1. Hasil Pengujian Ahli

Dalam penelitian ini pengujian ahli dilakukan oleh tiga orang yang memiliki kemampuan di bidang teknologi informasi. Berikut hasil pengujiannya.

SS	S	C	K	T
4	9	3	-	-

Keterangan:

SS = 5, S = 4, C = 3, K = 2, T = 1

$$x = \frac{65}{80} \times 100\% = 81,25\%$$

Hasil dari pengujian validator mendapatkan nilai persentase yaitu 81,25%. Dengan saran dari pengujian yang masih mempunyai kekurangan pada fitur dan fungsi fungsi CRUD belum sempurna.

4.5.2. Hasil Pengujian Pengguna

Berikut adalah hasil dari jumlah pengujian yang sudah ditujukan untuk menilai aplikasi dari penelitian ini.

SS	S	C	K	T
26	45	9	-	-

Keterangan:

SS = 5, S = 4, C = 3, K = 2, T = 1

$$x = \frac{337}{400} \times 100\% = 84,25\%$$

Pada kuisisioner diatas mempunyai hasil persentase yaitu 84,25%, dan terdapat pengecekan terhadap pengguna aplikasi yang dimana terjadi beberapa *error* pada tombol CRUD dan fitur dari aplikasi yang belum sesuai dengan yang diharapkan.

5. KESIMPULAN DAN SARAN

Proses perancangan aplikasi penentuan tambal ban motor di Kabupaten Blitar menggunakan metode *waterfall* berbasis *Android* melibatkan tahap identifikasi kebutuhan, desain antarmuka, implementasi kode, pengujian menyeluruh, peluncuran, dan pemeliharaan. Pendekatan ini memungkinkan pengembangan yang terstruktur dan hasil akhir yang berkualitas, serta aplikasi yang sesuai dengan perancangan dari penulis.

Hasil pengujian *Whitebox* pada aplikasi penentuan lokasi terdekat tambal ban berbasis *Android* menunjukkan bahwa kompleksitas kode rendah, dengan nilai *cyclomatic complexity* dan jalur independen (*Independent Path*) masing-masing sebesar 1 yang teridentifikasi sebagai *low risk* atau *low error count*. Pada pengujian kuesioner mendapatkan nilai masing-masing 81,25% dari Ahli IT dan 84,25% dari pengguna aplikasi, hal ini mengindikasikan bahwa

aplikasi memiliki struktur linear yang efisien dalam menerapkan metode *waterfall*. Dengan hanya satu jalur independen, setiap fitur dan fungsi dapat diuji dengan baik, memastikan kinerja yang lancar dan minim kesalahan. Sehingga, aplikasi ini dapat dianggap berhasil dalam mencapai tujuan utamanya, yaitu menemukan lokasi tambal ban pada platform *Android*.

DAFTAR PUSTAKA

- [1] V. L. Pratama and D. A. Darmawan, "Sistem Informasi Geografis Pencarian Rute Terdekat Bengkel Motor di Kota Surabaya Menggunakan Algoritma Bellman-Ford," *J. Informatics Comput. Sci.*, vol. 03, 2022.
- [2] S. C. G. Sihombing, T. R. Siringoringo, and V. Manullang, "Analisis kemacetan daerah terhadap peningkatan pembelian sepeda motor akibat dari kepadatan penduduk," vol. 1, no. 12, pp. 1531–1535, 2023.
- [3] M. R. Adyatma and T. S. Hadi, "Analisis Penentuan Lokasi Perumahan Oleh Developer Untuk Masyarakat Berpenghasilan Rendah Pada Sekitar Kawasan Industri," *J. Kaji. Ruang*, vol. 1, no. 2, p. 198, 2022, doi: 10.30659/jkr.v1i2.20025.
- [4] S. Tumanggor, "MENGUNAKAN ALGORITMA A*," 2019.
- [5] M. Afandi, "APLIKASI ANDROID UNTUK PENCARIAN RUTE TERDEKAT MINI MARKET MENGGUNAKAN METODE FLOYD-WARSHALL BERBASIS GIS," *Ubiquitous Comput. its Appl. J.*, vol. 2, no. 1, pp. 51–56, 2019.
- [6] A. Fatkharrofiqi and W. Gata, "Implementasi Algoritma Dijkstra dalam penentuan rute terdekat menuju Masjid di Perumahan Bona Indah Lebak Bulus," *J. Inf. Syst. Applied, Manag. Account. Res.*, vol. 6, no. 1, p. 87, Feb. 2022, doi: 10.52362/jisamar.v6i1.674.
- [7] M. Syepanda, Z. Zulhalim, and R. Haroen, "Perancangan Aplikasi Pencarian Rute Wisata Kuliner Berbasis Android Menggunakan Algoritma Dijkstra Di Kota Tangerang Selatan," *J. Manajemen Inform. Jayakarta*, vol. 1, no. 2, p. 117, 2021, doi: 10.52362/jmijayakarta.v1i2.448.
- [8] A. Qaedi, Y. Sholva, and N. Safriadi, "Rancang Bangun Aplikasi Pemilihan Rute Terpendek Menggunakan Metode Simulated Annealing Design and Development of the Shortest Route Selection Application Using the Simulated Annealing Method," vol. 01, no. 1, pp. 85–92, 2023, doi: 10.26418/juristi.v1i1.61131.
- [9] H. Gusdevi, S. Kuswayati, M. Iqbal, M. F. Abu Bakar, N. Novianti, and R. Ramadan, "Pengujian White-Box Pada Aplikasi Debt Manager Berbasis Android," *Naratif J. Nas. Riset, Apl. dan Tek. Inform.*, vol. 4, no. 1, pp. 11–22, 2022, doi: 10.53580/naratif.v4i1.147.
- [10] T. I. Pratiwi, Y. R. W. Utami, and S. H. Fitriasih, "PENENTUAN RUTE TERDEKAT LOKASI KLINIK BERSALIN DI KOTA SURAKARTA MENGGUNAKAN ALGORITMA FLOYD WARSHALL," *J. Teknol. Inf. dan Komun.*, vol. 6, no. 2, Jan. 2019, doi: 10.30646/tikomsin.v6i2.372.