



Penerapan Algoritma Dijkstra untuk Menemukan Tempat Ibadah Terdekat di Kota Sukabumi Menggunakan Metode Agile

Fauzia Dewi Amalia^{1*}, Asep Budiman Kusdinar²

¹Universitas Muhammadiyah Sukabumi,
Sukabumi, Jawa Barat, Indonesia
fauziadewi0401@gmail.com

²Universitas Muhammadiyah Sukabumi,
Sukabumi, Jawa Barat, Indonesia
asep.budiman.k@gmail.com

*Corresponding Author

Abstrak:

Dalam era kemajuan teknologi, navigasi dan peta telah menjadi kebutuhan sehari-hari, terutama dalam pencarian tempat ibadah di kota berkembang seperti Kota Sukabumi. Penelitian ini mengeksplorasi peran Algoritma Dijkstra dalam memfasilitasi pencarian tempat ibadah dengan beragam jenis, seperti masjid, gereja, pura, dan vihara. Algoritma ini diimplementasikan dalam pencarian untuk mempermudah akses masyarakat, menghitung jarak dan rute terdekat dari lokasi pengguna secara efisien. Pentingnya memasukkan informasi fasilitas, seperti parkir yang memadai, juga ditekankan untuk memberikan pengalaman lengkap. Tidak hanya membantu mencari lokasi tetapi juga mendukung kebutuhan umum pengguna, menciptakan lingkungan inklusif untuk menjalankan ibadah dan bepergian dengan nyaman. Integrasi teknologi dalam pencarian lokasi dengan memperhatikan ketersediaan fasilitas menjadi kunci untuk menciptakan lingkungan yang mendukung masyarakat dalam menjalankan ibadah dengan baik.

Kata kunci:

Algoritma Dijkstra, Fasilitas, Kota Sukabumi, Peta, Tempat ibadah

Abstract:

In the era of technological advancements, navigation systems and maps have become essential in our daily lives, especially in the search for places of worship in developing cities like Sukabumi. This research explores the pivotal role of the Dijkstra algorithm in facilitating the search for various places of worship, such as mosques, churches, temples, and viharas. The algorithm is implemented in a search system to streamline public access, efficiently calculating the shortest distance and route from the user's location. Emphasizing the importance of including facility information, such as adequate parking, is highlighted to provide a comprehensive user experience. This system not only aids in location searches but also supports the general needs of users, creating an inclusive environment for practicing worship and traveling comfortably. The integration of technology in location searches, coupled with a focus on facility availability, is crucial in creating an environment that supports the community in practicing their worship effectively.

Keywords:

Dijkstra Algorithm, Facilities, Sukabumi City, Map, Places of Worship

1. Pendahuluan

Dengan kemajuan teknologi, navigasi dan peta kini menjadi bagian penting dalam kehidupan sehari-hari. Pertumbuhan populasi yang cepat menambah tantangan dalam menyediakan informasi yang memudahkan pencarian lokasi penting, seperti tempat ibadah [1]. Menemukan tempat ibadah yang relevan sangat penting, terutama di kota-kota berkembang seperti Sukabumi, yang memiliki berbagai tempat ibadah seperti masjid, gereja, pura, dan vihara yang tersebar di berbagai wilayah. tempat-tempat ini sering dikunjungi oleh penduduk lokal maupun wisatawan[2].

Menggunakan Algoritma Dijkstra dalam pencarian tempat ibadah di Sukabumi dapat mempermudah akses masyarakat ke lokasi-lokasi tersebut[3]. Algoritma ini memungkinkan perhitungan jarak dan rute terpendek dari lokasi pengguna secara efisien, sehingga mempermudah penemuan tempat ibadah terdekat[4]. Namun, kemudahan akses bukanlah satu-satunya pertimbangan. ketersediaan fasilitas di tempat ibadah juga penting. Menyediakan informasi mengenai fasilitas seperti tempat parkir dan akomodasi lainnya sangat penting untuk memastikan pengalaman pengguna yang nyaman[5]. Mengatasi kebutuhan akan tempat ibadah dan kualitas lingkungan sekitar akan meningkatkan kenyamanan ibadah dan perjalanan. Oleh karena itu, integrasi teknologi untuk pencarian lokasi dan detail fasilitas sangat penting untuk menciptakan lingkungan inklusif bagi komunitas dalam menjalankan keyakinan mereka[6]. Penelitian sebelumnya oleh Satria menunjukkan bahwa wisatawan mengalami kesulitan menemukan tempat ibadah di Kota Mataram, dan ada kekurangan dalam pengelolaan data oleh Departemen Agama. Penelitian ini bertujuan merancang aplikasi informasi geografis yang efektif dan efisien menggunakan Metode *Extreme Programming* (XP) untuk memetakan lokasi tempat ibadah di Kota Mataram.[7]

Penelitian ini mengidentifikasi beberapa kendala, seperti kurangnya akses informasi mengenai lokasi tempat ibadah terdekat dan keterbatasan informasi tentang fasilitas di tempat ibadah. Dengan fokus pada pembuatan aplikasi Android, penelitian ini bertujuan memberikan akses yang lebih baik dan akurat terkait lokasi tempat ibadah, serta menyediakan informasi rinci mengenai fasilitas seperti ruang parkir dan aksesibilitas untuk penyandang disabilitas. Diharapkan, penelitian ini dapat meningkatkan aksesibilitas informasi mengenai lokasi tempat ibadah, menyediakan data rinci tentang fasilitas, dan memudahkan pengguna dalam memilih tempat ibadah yang sesuai dengan kebutuhan serta preferensi mereka, sehingga meningkatkan pengalaman keseluruhan dalam mencari tempat ibadah.

2. Metode

2.1. Algoritma Dijkstra

Edsger Wybe Dijkstra mengembangkan Algoritma Dijkstra pada tahun 1959 untuk mengatasi masalah pencarian jalur terpendek dalam sebuah graf, dimana setiap sisi memiliki bobot yang mewakili jarak minimum antar node[8]. Algoritma ini mengidentifikasi rute terpendek dengan mengevaluasi jalur yang paling efisien dibandingkan dengan Algoritma Warshall, yang umumnya menawarkan efisiensi yang unggul. Meskipun dirancang khusus untuk graf berarah, Algoritma Dijkstra juga dapat digunakan dengan graf tidak berarah[9]. Pendekatan mendasar dari Algoritma Dijkstra adalah strategi serakahnya.

Dalam iterasinya, algoritma akan mencari satu titik yang jumlah bobotnya dari titik 1 terkecil. Titik-titik yang terpilih dipisahkan, dan titik-titik tersebut tidak diperhatikan lagi dalam iterasi berikutnya[10].

Dalam penelitian ini, Algoritma Dijkstra digunakan untuk mencari rute terpendek menuju tempat ibadah di Sukabumi berdasarkan lokasi pengguna. Penerapan algoritma ini dalam aplikasi Android memungkinkan pengguna untuk mengidentifikasi jalur tercepat dari titik awal menuju tempat ibadah yang dituju, dengan mempertimbangkan jarak antar titik yang tersimpan di Firebase.

Penggunaan Algoritma Dijkstra pada penelitian ini meliputi beberapa tahapan yaitu:

- a. Pengumpulan data lokasi tempat ibadah di Sukabumi, termasuk koordinat GPS tiap titik, disimpan dalam Firebase. Data ini di gunakan untuk membuat graf yang mewakili hubungan antar tempat ibadah dengan bobot yang menunjukkan jarak antar titik.
- b. Penentuan titik awal dan tujuan, titik awal di tentukan berdasarkan lokasi Pengguna saat ini yang di dapatkan melalui GPS dan memilih tempat ibadah sebagai tujuan.
- c. Penerapan Algoritma Dijkstra untuk menghitung jarak terpendek dari titik awal ke semua titik dalam graf, mengidentifikasi jalur terpendek ke tempat ibadah yang dipilih.
- d. Visualisasi Rute, Hasil Algoritma Dijkstra divisualisasikan pada peta di dalam aplikasi, menampilkan rute terpendek yang harus diambil pengguna untuk mencapai tujuannya.

2.2. Metode Agile



Gambar 1: Metode Agile

Pada Gambar 1 Agile adalah pendekatan pengembangan perangkat lunak yang cepat dan efektif yang mendukung pengembangan modular dengan memisahkan produk dari proses, sehingga memfasilitasi penyelesaian yang cepat[11]. Kekuatannya terletak pada fokusnya pada pengujian berkelanjutan, yang membantu mengurangi kesalahan dan meningkatkan kualitas produk. Agile menerapkan pengembangan berulang, membagi pekerjaan menjadi siklus pendek dan dapat dikelola yang dikenal sebagai *sprint*. Setiap *sprint* menghasilkan versi produk yang dapat diuji dan ditinjau, memungkinkan perbaikan dan penyesuaian cepat berdasarkan masukan pengguna. Metode ini memastikan bahwa produk terus beradaptasi untuk memenuhi kebutuhan dan harapan pengguna dengan lebih baik[12].

Metode Agile digunakan dalam penelitian ini untuk memastikan pengembangan aplikasi yang cepat, efisien, dan responsif terhadap perubahan kebutuhan pengguna. Proses Agile diterapkan melalui tahapan-tahapan berikut:

a. *Planning*

Fase perencanaan memulai setiap *sprint*, di mana pemangku kepentingan menentukan tujuan *sprint*, memprioritaskan fitur, dan menguraikan rencana yang jelas untuk mencapai tujuan tersebut[13]. Fase ini meliputi analisis tujuan penelitian, penetapan tujuan spesifik, identifikasi masalah, dan pengumpulan informasi kebutuhan pengguna dalam pencarian lokasi ibadah. Rencana pengembangan terperinci dibuat, mencakup fitur utama seperti fungsi pencarian, detail fasilitas, dan integrasi peta. Analisis persyaratan sangat penting, dengan fokus pada kebutuhan pengguna dan mengumpulkan data tempat ibadah di Kota Sukabumi untuk menentukan persyaratan fungsional dan non-fungsional untuk pengalaman pengguna yang optimal.

b. *Design*

Setelah perencanaan, tahap desain melibatkan pengembangan solusi teknis untuk fitur atau perubahan. Termasuk di dalamnya pembuatan arsitektur sistem dan antarmuka pengguna yang akan diimplementasikan[14]. Proses ini menggunakan flowchart dan wireframe untuk merancang arsitektur aplikasi Android dengan integrasi Algoritma Dijkstra. Flowchart digunakan untuk memvisualisasikan alur logika dan proses dalam aplikasi, mulai dari input pengguna hingga output yang dihasilkan. Wireframe digunakan untuk mendesain antarmuka pengguna secara detail, menampilkan elemen-elemen visual seperti peta dan ikon untuk membantu navigasi dan detail lokasi. Desainnya mematuhi prinsip UI/UX, memastikan kejelasan, keterbacaan, dan daya tarik visual sekaligus memenuhi kebutuhan fungsional pengguna.

c. *Development*

Pada tahap *development*, desain diubah menjadi kode sebenarnya. Pengembang menulis kode untuk fungsi atau modul yang ditentukan dalam iterasi, memastikan kode tersebut mematuhi spesifikasi desain[15]. Pembuatan aplikasi melibatkan pelaksanaan desain dan spesifikasi yang direncanakan. Proses ini mencakup pengkodean dan penggabungan Algoritma Dijkstra untuk memungkinkan pencarian rute yang tepat berdasarkan koordinat geografis pengguna. Langkah-langkah ini sangat penting untuk memastikan aplikasi bekerja secara maksimal, memberikan informasi tempat ibadah yang akurat dan efisien bagi pengguna di Kota Sukabumi.

d. *Testing*

testing Ini melibatkan pengujian unit, pengujian integrasi, dan pengujian sistem untuk mendeteksi dan menyelesaikan masalah apa pun. pengujian unit untuk memverifikasi secara independen bahwa Algoritma Dijkstra berfungsi dengan benar dan memberikan hasil yang diharapkan. Pengujian integrasi dilakukan untuk memastikan bahwa komponen aplikasi yang berbeda bekerja sama secara lancar tanpa masalah. Selain itu, pengujian *widget* dan UI menilai antarmuka pengguna untuk memastikannya cocok dengan desain dan menawarkan pengalaman pengguna yang positif. Ini melibatkan pemeriksaan elemen seperti tombol, kolom input, dan navigasi layar untuk konsistensi fungsi dan tampilan.

e. *Deployment*

Setelah tahap pengujian berhasil diselesaikan, fitur atau perubahan yang dikembangkan siap untuk diterapkan di lingkungan produksi atau untuk pengujian mendalam lebih lanjut. Proses

penerapan ini mengonfirmasi bahwa aplikasi telah lulus semua persyaratan pengujian yang diperlukan dan siap untuk penggunaan aktual atau evaluasi tambahan.

f. *Review*

Evaluasi aplikasi dilakukan melalui usability testing untuk mengukur kecepatan, responsivitas, dan pengalaman pengguna secara keseluruhan. Penilaian difokuskan pada fungsi inti seperti pencarian tempat ibadah dan penghitungan rute terdekat, serta antarmuka pengguna (UI/UX) untuk memastikan kejelasan, kemudahan navigasi, dan konsistensi desain di seluruh perangkat. Potensi perbaikan diidentifikasi untuk memastikan aplikasi memberikan pengalaman pengguna yang optimal dan memenuhi kebutuhan pengguna yang mencari tempat ibadah di Kota Sukabumi.

g. *launch*

Setelah aplikasi melewati evaluasi akhir dan dianggap siap, aplikasi diluncurkan ke lingkungan produksi. Ini melibatkan pengujian putaran terakhir sebelum aplikasi diterapkan ke *server* atau platform target.

3. Hasil Dan Pembahasan

3.1. *Planning*

Dalam tahap perencanaan ini, langkah-langkah penting diambil untuk memastikan keberhasilan aplikasi. Awalnya, identifikasi tempat ibadah di Kota Sukabumi dilakukan dengan mengumpulkan informasi dari berbagai sumber. Pendataan meliputi rincian lokasi, jenis tempat ibadah, dan fasilitas yang tersedia di setiap lokasi. Hal ini memastikan aplikasi tidak hanya akurat dan efektif dalam memberikan informasi tetapi juga memenuhi beragam kebutuhan pengguna. Informasi yang dikumpulkan menjadi dasar utama penerapan Algoritma Dijkstra untuk mencari tempat ibadah terdekat.

A. Rencana pembuatan aplikasi

1. Pembuatan aplikasi direncanakan menggunakan flutter untuk *frontend* dan Firebase Firestore untuk pengelolaan data tempat ibadah.
2. Spesifikasi teknisnya meliputi MapBox untuk itegrasi peta dan Algoritma Dijkstra untuk *routing*.

B. Kebutuhan fungsional

1. Integrasi dengan layanan peta diperlukan untuk memvisualisasikan lokasi dan rute terpendek menggunakan Algoritma Dijkstra. Hal ini melibatkan evaluasi API pemetaan yang tersedia dan perencanaan integrasi teknis dengan aplikasi.
2. Aplikasi harus memberikan informasi detail mengenai fasilitas di setiap tempat ibadah, seperti tempat parkir, toilet, dan fasilitas terkait lainnya. Ini melibatkan pencatatan dan pengorganisasian informasi untuk setiap lokasi.
3. Fitur penting antara lain tombol akses cepat berbagai jenis tempat ibadah di halaman utama, serta perancangan dan penetapan desain antarmuka pada setiap tombol.

C. Kebutuhan non-fungsional

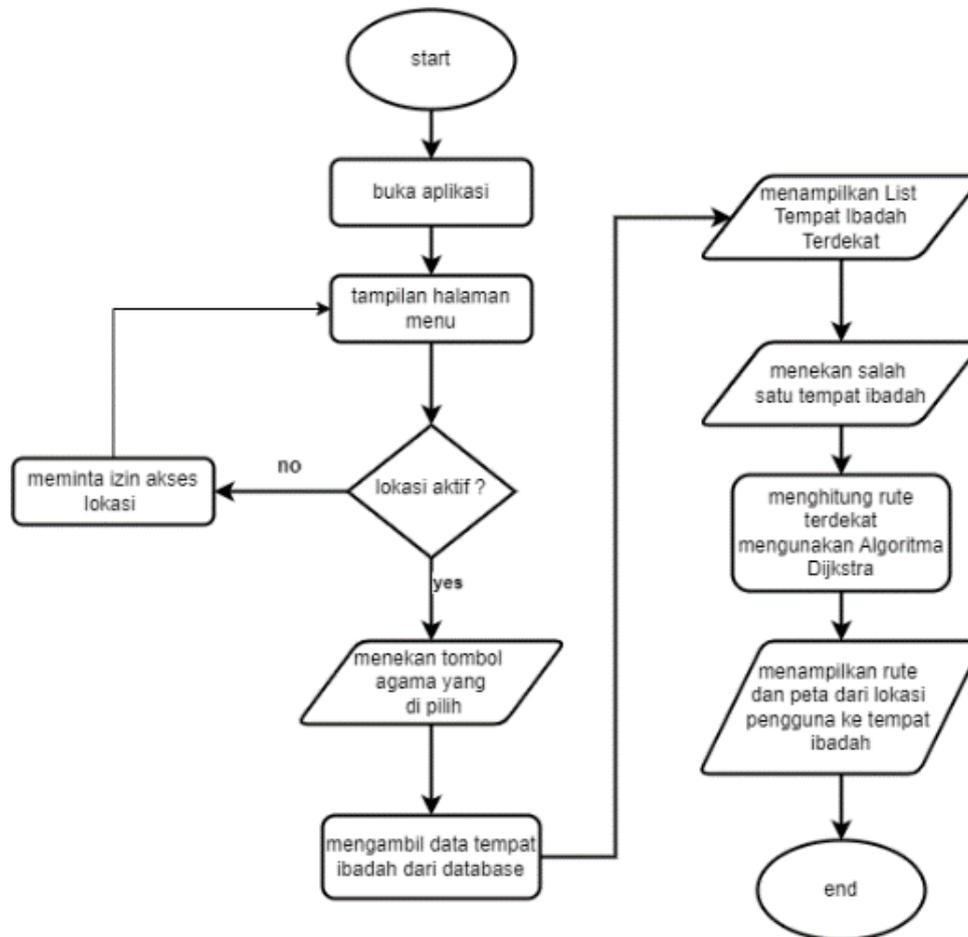
1. Memastikan akurasi lokasi dengan mengintegrasikan data GPS yang akurat dengan layanan pemetaan yang andal.
2. Desain antarmuka yang ramah pengguna sangat penting untuk pengalaman pengguna yang baik. Hal ini termasuk membuat wireframe, merancang antarmuka pengguna, dan melakukan pengujian pengguna awal untuk mengumpulkan umpan balik.

3. Persyaratan kinerja harus diidentifikasi untuk memastikan aplikasi berjalan lancar di berbagai perangkat Android. Hal ini mencakup pengujian kinerja, pengoptimalan kode, dan memastikan kompatibilitas dengan versi Android yang berbeda.

3.2. Design

Pada tahap desain, penekanan utama adalah pada pengembangan arsitektur sistem dan antarmuka pengguna yang efisien melalui penggunaan diagram alur. Flowchart secara visual menggambarkan alur kerja aplikasi, termasuk integrasi Algoritma Dijkstra untuk menghitung rute terpendek.

A. Flowchart



Gambar 2: flowchart

Pada Gambar 2 Aplikasi yang dirancang untuk menemukan tempat ibadah terdekat menggunakan algoritma Dijkstra ini mengikuti alur kerja sistematis untuk memastikan pengguna dapat dengan mudah menemukan tujuan yang diinginkan. Saat meluncurkan aplikasi, pengguna akan disambut oleh halaman menu utama yang menampilkan beberapa tombol untuk memilih agama yang berbeda. Pada titik ini, aplikasi meminta izin untuk mengakses lokasi pengguna. Jika layanan lokasi belum diaktifkan, aplikasi akan berulang kali meminta izin hingga lokasi diaktifkan.

Setelah lokasi aktif, pengguna dapat memilih tombol yang sesuai dengan agama pilihan mereka. Aplikasi kemudian mengambil data tentang tempat ibadah yang relevan dari basis data. Data ini digunakan untuk membuat daftar tempat ibadah terdekat berdasarkan lokasi pengguna saat ini, yang dapat dipilih pengguna.

Aplikasi kemudian menggunakan algoritma Dijkstra untuk menghitung rute terpendek dari lokasi pengguna ke tempat ibadah yang dipilih. Rute ini ditampilkan pada peta, yang menawarkan navigasi visual kepada pengguna. Proses ini berakhir setelah rute dan peta berhasil disajikan, yang memungkinkan pengguna untuk mencapai tempat ibadah dengan mudah dan efisien.

3.3. Development

Tahap Pengembangan melibatkan perubahan konsep dari tahap desain menjadi aplikasi yang berfungsi. Hal ini mencakup penerapan algoritme, pembuatan fitur, dan pelaksanaan pengujian berkelanjutan untuk memastikan kualitas dan konsistensi di seluruh aplikasi.

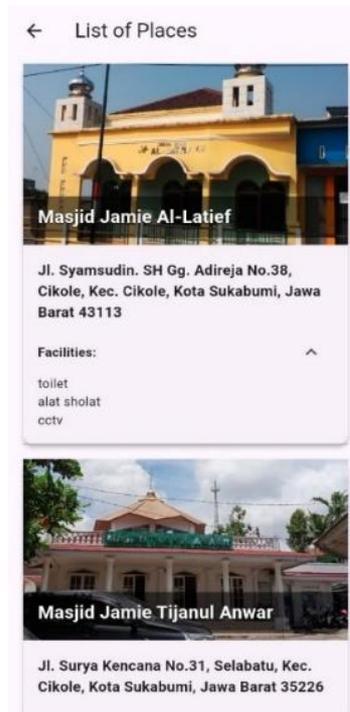
A. Halaman utama



Gambar 3: Halaman utama

Pada Gambar 3 Ketika halaman utama muncul, ini menunjukkan antarmuka pilihan utama aplikasi, dan sistem memeriksa apakah izin akses lokasi telah diberikan. Jika tidak, aplikasi akan meminta pengguna untuk mengizinkan akses. Setelah izin diberikan, aplikasi mengambil data lokasi perangkat saat ini dan menggunakannya dalam aplikasi.

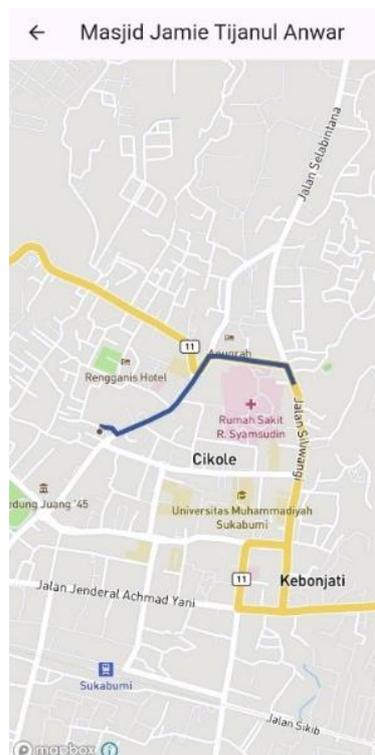
B. Halaman list tempat ibadah terdekat



Gambar 4: list tempat ibadah terdekat

Pada Gambar 4 Ketika halaman daftar tempat ibadah yang dipilih ditampilkan, pengguna melihat daftar tempat ibadah terdekat berdasarkan data lokasi yang diambil. Daftar ini mencakup informasi tentang fasilitas yang tersedia di setiap lokasi.

C. Halaman rute terdekat

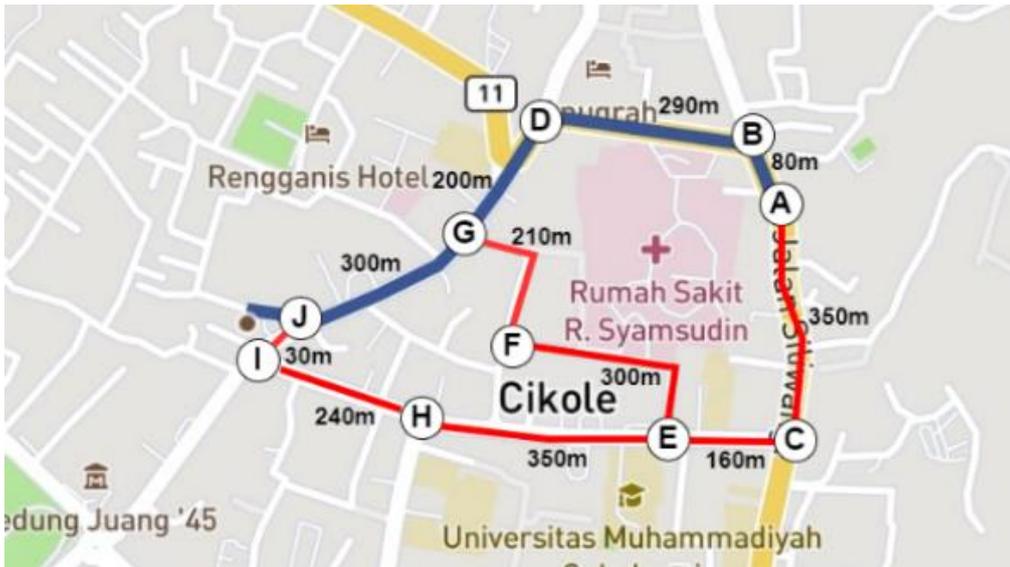


Gambar 5: Halaman rute terdekat

Pada Gambar 5 halaman rute terdekat, pengguna dapat melihat jalur tercepat menuju tempat ibadah pilihannya dari lokasinya saat ini. Aplikasi ini menggunakan Algoritma Dijkstra untuk menentukan rute tercepat, memastikan setiap keputusan didasarkan pada jarak terpendek untuk memberikan pengguna jalur paling efisien ke tujuan mereka.

3.4. Testing

Pengujian dilakukan dengan menggunakan ilustrasi node atau titik dan bobot yang terlibat dalam Algoritma Dijkstra untuk menentukan rute terdekat menuju masjid Jamie Tijanul Anwar.



Gambar 6: Penggambaran node

Pada gambar 6 merupakan ilustrasi berupa node atau titik serta bobot pada tiap node. Hasil pengujian node suatu tempat kearah masjid Jamie tijanul anwar dapat dilihat pada Tabel di bawah ini.

Tabel 1: Hasil pengujian Algoritma Dijkstra

Langkah	pilih	terkecil	A	B	C	D	E	F	G	H	I	J
1	A	0	0a	80a	350a	∞						
2	B	80a	0a	80a	350a	370b	∞	∞	∞	∞	∞	∞
3	D	370b	0a	80a	350a	370b	∞	∞	570d	∞	∞	∞
4	G	570d	0a	80a	350a	370b	∞	∞	570d	∞	∞	1080d
5	J	1080g	0a	80a	350a	370b	∞	∞	570d	∞	∞	1080d

Dari tabel pengujian dapat disimpulkan bahwa rute terpendek yang dihitung dengan menggunakan Algoritma Dijkstra adalah rute A-B-D-G-J, dengan total jarak 1080 meter. Ini merupakan jarak terpendek dibandingkan rute lainnya. Pengujian tambahan mencakup pengujian unit, pengujian antarmuka pengguna, pengujian integrasi, dan pengujian fungsional. Proses pengujian komprehensif ini dirancang untuk mengidentifikasi dan mengatasi kesalahan, memastikan aplikasi memenuhi persyaratan yang ditentukan.

Tabel 2: Pengujian aplikasi

No	Nama tes	Jenis pengujian	Deskripsi	Langkah pengujian	Status
1.	Tes Algoritma Dijkstra	Unit testing	Menguji perhitungan rute terpendek menggunakan algoritma Dijkstra	1. Inisialisasi graf 2. Jalankan Dijkstra 3. Verifikasi hasil jarak terpendek	Lulus
2.	Tes Pengambilan Data Firebase	Unit testing	Menguji fungsi pengambilan data dari Firebase	1. Hubungkan ke Firebase 2. Ambil data 3. Verifikasi data yang diambil	Lulus
3.	Tes Integrasi Firebase	Integration testing	Menguji integrasi aplikasi dengan Firebase	1. Hubungkan ke Firebase 2. Simpan data 3. Ambil data 4. Verifikasi data yang diambil	Lulus
4.	Tes Tampilan Daftar Tempat Ibadah	UI testing	Menguji tampilan daftar tempat ibadah	1. Buka aplikasi 2. Navigasi ke halaman daftar 3. Verifikasi tampilan daftar tempat ibadah	Lulus
5.	Tes Fitur aplikasi Tempat Ibadah	Functional testing	Menguji fungsi pencarian tempat ibadah berdasarkan lokasi	1. Buka aplikasi 2. pilih agama 3. muncul list tempat ibadah terdekat 4. pilih tempat ibadah yang sesuai 5. muncul rute ke tempat ibadah yang di pilih	Lulus
6.	Tes Integrasi Peta	Integration testing	Menguji integrasi dengan layanan peta	1. Hubungkan ke layanan peta 2. Tampilkan lokasi tempat ibadah 3. Verifikasi lokasi	Lulus
7.	Tes Tampilan Rute Tempat Ibadah	UI testing	Menguji tampilan rute terpendek menuju tempat ibadah	1. Pilih tempat ibadah 2. Klik untuk melihat rute 3. Verifikasi tampilan rute	Lulus

3.5. Deployment

Setelah memverifikasi bahwa aplikasi berfungsi dengan benar dan bebas kesalahan melalui pengujian ekstensif, langkah selanjutnya adalah membuat APK versi final. Hal ini melibatkan konfigurasi build Gradle untuk mode rilis dan menandatangani APK dengan kunci yang valid.

3.6. Review

Berikut adalah hasil *usability*. pugujian ini dilakukan untuk evaluasi tampilan, ketersediaan informasi, dan kegunaan fitur dalam aplikasi.

Tabel 3: Pertanyaan usability testing

No.	Pertanyaan	Pilihan
1.	Apakah Anda bisa menemukan tombol untuk agama yang Anda cari dengan mudah?	<ul style="list-style-type: none"> • Sangat mudah • Mudah • Sulit • Sangat sulit
2.	Bagaimana pengalaman Anda saat mencari tempat ibadah terdekat?	<ul style="list-style-type: none"> • Sangat mudah • Mudah • Sulit • Sangat sulit
3.	Apakah informasi fasilitas di tempat ibadah cukup lengkap?	<ul style="list-style-type: none"> • Sangat lengkap • Lengkap • Kurang lengkap • Sangat kurang lengkap
4.	Seberapa cepat aplikasi merespons perintah Anda?	<ul style="list-style-type: none"> • Sangat cepat • Cepat • Lambat • Sangat lambat
5.	Apakah tampilan antarmuka aplikasi menarik?	<ul style="list-style-type: none"> • Sangat menarik • Menarik • Tidak menarik • Sangat tidak menarik

Tabel 4: Hasil usability testing

Aspek	Hasil	Keterangan
Kemudahan Penggunaan	Pengguna merasa sangat mudah (67%) dan mudah (33%) dalam menemukan tombol agama yang dicari.	Sebagian besar pengguna menilai navigasi untuk memilih agama cukup intuitif.
Pencarian Tempat Ibadah	Semua pengguna (100%) merasa sangat mudah dalam mencari tempat ibadah terdekat.	Proses pencarian tempat ibadah dinilai efektif dan mudah dipahami.
Kelengkapan Informasi	Sebagian besar pengguna (83%) merasa informasi fasilitas di tempat ibadah sudah lengkap.	Namun, perlu perhatian lebih untuk memastikan semua informasi yang dibutuhkan tersedia secara detail.
Responsivitas Aplikasi	Separuh pengguna (50%) merasa aplikasi merespons dengan sangat cepat, dan separuhnya lagi (50%) merasa cepat.	Responsivitas yang baik merupakan faktor kunci dalam pengalaman pengguna yang positif.
Tampilan Antarmuka	Mayoritas pengguna (67%) menganggap tampilan antarmuka aplikasi sangat menarik.	Desain visual yang atraktif dapat meningkatkan daya tarik aplikasi.

Berdasarkan analisis pengujian kegunaan, aplikasi menerima tanggapan positif di sebagian besar area yang dievaluasi. Pengguna menilai aplikasi ini sangat tinggi karena kemudahan penggunaannya, pengambilan informasi, dan daya tanggapnya. Namun, ada beberapa hal yang memerlukan perbaikan, seperti memastikan kelengkapan informasi dan terus meningkatkan kinerja aplikasi untuk menjaga pengalaman pengguna yang optimal. Dengan mempertahankan standar kualitas ini dan mengatasi area yang teridentifikasi selama pengujian, aplikasi dapat lebih meningkatkan kepuasan pengguna dan kegunaan secara keseluruhan.

3.7. Launch

Setelah aplikasi dievaluasi dan memenuhi semua persyaratan, langkah selanjutnya adalah merilisnya ke pengguna. Pada tahap ini, aplikasi akan diluncurkan secara resmi dan tersedia untuk diunduh atau digunakan publik.

4. Kesimpulan

Setelah melakukan serangkaian proses pembuatan aplikasi Android untuk mencari tempat ibadah terdekat di Sukabumi menggunakan algoritma Dijkstra, beberapa kesimpulan dapat diambil dari penelitian ini:

1. Efektivitas Algoritma Dijkstra
Algoritma Dijkstra terbukti efektif dalam mencari rute terpendek dari lokasi pengguna menuju tempat ibadah. Dengan menggunakan data koordinat GPS yang tersimpan di Firebase, aplikasi dapat menghitung jarak dan menampilkan rute terpendek secara real-time.
2. Integrasi Layanan Berbasis Lokasi
Kombinasi Flutter dan Firebase dalam aplikasi ini memastikan integrasi layanan berbasis lokasi yang lancar. Firebase mengelola penyimpanan dan pengambilan data tempat ibadah secara efisien, sementara
3. Pengalaman Pengguna dan Kegunaan
Hasil pengujian menunjukkan bahwa aplikasi ini sangat ramah pengguna. Pengguna dapat dengan mudah menemukan tempat ibadah terdekat dan mengakses informasi terperinci tentang fasilitas yang tersedia.

Pustaka

- [1] B. lutfi A. Rofiq, H. Februariyanti, J. S. Wibowo, and Z. Budiarmo, "Pencarian Tempat Ibadah Terdekat Menggunakan Algoritma Dijkstra," *JUSIM (Jurnal Sist. Inf. Musirawas)*, vol. 6, no. 1, pp. 19–28, 2021, doi: 10.32767/jusim.v6i1.1138.
- [2] M. K. Negara, R. L. Rahardian, and I. G. N. D. Widiastara, "Sistem Informasi Geografis Pemetaan Tempat Ibadah Pada Pulau Bali Berbasis Mobile," *J. Innov. Inf. Technol. Appl.*, vol. 2, no. 01, pp. 56–66, 2020, doi: 10.35970/jjinita.v2i01.198.
- [3] Y. N. et al Marlin, "Analisis Sistem Jalur Terpendek Menggunakan," *JEPIN (Jurnal Edukasi dan Penelit. Inform.)*, vol. 6, no. 1, pp. 54–60, 2020.
- [4] A. S. Salomo Silaban, A. Haidar Mirza, "Bina Darma Conference on Computer Science 2019," *Fak. ilmu Komput. Univ. Bina Dharma*, pp. 270–283, 2019.
- [5] A. K. Sonita and A. Dan Saroni, "Pencarian Lokasi Tempat Ibadah Terdekat di Kota Bengkulu Menggunakan Algoritma Ant Colony Optimization (ACO) Searching for the Nearest Places of Worship in Bengkulu City Using Ant Colony Optimisation Algorithm," vol. 13, no. 2, p. 13, 2023.

- [6] M. Muharrom, "Implementasi Algoritma Dijkstra Dalam Penentuan Jalur Terpendek Studi Kasus Jarak Tempat Kuliah Terdekat," *Indones. J. Bus. Intell.*, vol. 3, no. 1, p. 25, 2020, doi: 10.21927/ijubi.v3i1.1229.
- [7] R. Satria, R. Afwani, and E. Anjarwani, "Rancang Bangun Sistem Informasi Geografis untuk Pemetaan Lokasi Tempat Ibadah di Kota Mataram Menggunakan Metode Extreme Programming (Design of Geographic Information System for Mapping the Location of Worship in Mataram City Using Extreme Programming Met," vol. 5, no. 1, pp. 33–41, 2023, [Online]. Available: <http://jcosine.if.unram.ac.id/>
- [8] E. S. Arga, G. G. Firmansyah, K. Imam, and M. Fauzi, "Penerapan Algoritma Dijkstra pada Pencarian Jalur Terpendek," *J. Bayesian*, vol. 1, no. 2, pp. 134–142, 2021, [Online]. Available: <https://bayesian.lppmbinabangsa.id/index.php/home/article/download/15/19>
- [9] A. M. Inayah, N. C. Resti, and N. F. Ilmiyah, "Analisa Perbandingan Algoritma Floyd-Warshall dan Algoritma Dijkstra Untuk Penentuan Rute Terdekat," *J. Ilm. Mat. Realis.*, vol. 4, no. 2, pp. 146–155, 2023.
- [10] S. Panggabean and S. Panggabean, "Implementasi Algoritma Dijkstra Untuk Menentukan Jalur Terpendek Wilayah Pasar Minggu Dan STMIK Nusamandiri Jakarta," *Swabumi*, vol. 9, no. 1, pp. 78–85, 2021, doi: 10.31294/swabumi.v9i1.9574.
- [11] F. Anisa, S. Fauzi, H. Harahap, P. Al Khosyi, and Y. Sari, "Pengembangan Software Menggunakan Model SDLC Guna Mencapai Keselarasan dengan Kebutuhan Pengguna," *J. Informatics Business*, vol. 01, no. 04, pp. 229–232, 2024, [Online]. Available: <https://jurnal.ittc.web.id/index.php/jibs/index>
- [12] C. Rimbing, G. Rorimpandey, and V. Rantung, "Pengembangan sistem repositori skripsi di teknik informatika universitas negeri manado berbasis web," pp. 39–48, 2023.
- [13] S. Yoga Pamungkas and Rahmawati, "Implementasi Metode Agile dalam Perancangan Sistem User Request Form (Urf) Versi Digital Berbasis Web pada CV.Gunung Sewu Pariwisata," *Sci. Sacra J. Sains*, vol. 2, no. 2, 2022, [Online]. Available: <http://pijarpemikiran.com/index.php/Scientia>
- [14] I. Larasati, A. N. Yusril, and P. Al Zukri, "Systematic Literature Review Analisis Metode Agile Dalam Pengembangan Aplikasi Mobile," *Sistemasi*, vol. 10, no. 2, p. 369, 2021, doi: 10.32520/stmsi.v10i2.1237.
- [15] O. Adeboye Popoola, H. Ejiga Adama, C. David Okeke, and A. Emmanuel Akinoso, "Conceptualizing Agile Development in Digital Transformations: Theoretical Foundations and Practical Applications," *Eng. Sci. Technol. J.*, vol. 5, no. 4, pp. 1524–1541, 2024, doi: 10.51594/estj/v5i4.1080.