

# PENERAPAN ALGORITMA DIJKSTRA DALAM MENENTUKAN RUTE TERPENDEK PADA TPA MRICAN

Diterima Redaksi: 7 Agustus 2024; Revisi Akhir: 24 Agustus 2024; Diterbitkan Online: 15 November 2024

**Fakih Kuncoro<sup>1)</sup>, Ismail Abdurrozzaq Zulkarnain<sup>2)</sup>, Ghulam Asrofi Buntoro<sup>3)</sup>**

<sup>1, 2, 3)</sup> Program Studi Informatika, Fakultas Teknik Universitas Muhammadiyah Ponorogo  
<sup>1, 2, 3)</sup> Jalan Budi Utomo No.10 Kec.Ponorogo, Kab.Ponorogo, Jawa Timur, Indonesia, kode pos: 63411  
e-mail: [fakihkuncoro1@gmail.com](mailto:fakihkuncoro1@gmail.com)<sup>1)</sup>, [ismail@umpo.ac.id](mailto:ismail@umpo.ac.id)<sup>2)</sup>, [ghulam@umpo.ac.id](mailto:ghulam@umpo.ac.id)<sup>3)</sup>

**Abstrak:** Perkembangan teknologi informasi yang pesat telah memudahkan pengelolaan pengumpulan sampah, yang menjadi tantangan utama bagi pemerintah daerah seiring dengan peningkatan populasi dan volume sampah. Penelitian ini bertujuan untuk mengoptimalkan rute pengumpulan sampah menggunakan Algoritma Dijkstra guna meminimalkan waktu tempuh dan biaya operasional di Kabupaten Ponorogo. Sistem ini dikembangkan menggunakan metode pengembangan sistem waterfall, yang meliputi tahapan analisis kebutuhan, desain sistem, implementasi, dan pengujian. Sistem ini dirancang untuk menghitung rute terpendek dari Tempat Pemrosesan Akhir (TPA) Mrican ke berbagai Tempat Penampungan Sementara (TPS) dan Pusat Daur Ulang (PDU) berdasarkan data lapangan. Dengan pendekatan ini, sistem dapat membantu mengoptimalkan proses pengangkutan sampah, mengurangi waktu perjalanan, serta menekan biaya operasional, sehingga pengelolaan sampah menjadi lebih efisien dan efektif. Hasil pengujian yang ditampilkan pada gambar 9 menunjukkan bahwa sistem berhasil menentukan rute pengumpulan sampah terpendek dengan total jarak 16,2 km melalui rute A-B-C-D-E-F. Simulasi ini membuktikan bahwa sistem dapat secara akurat mengoptimalkan rute pengumpulan sampah berdasarkan rute terpendek dengan total jarak 16,2 km melalui rute A-B-C-D-E-F. Dengan demikian, hasil sistem ini mampu mengurangi waktu dan biaya pengumpulan sampah secara signifikan serta memberikan kontribusi nyata pada peningkatan efisiensi pengelolaan sampah.

**Kata Kunci**—Algoritma Dijkstra, Metode Waterfall, Rute Terpendek.

**Abstract:** The rapid development of information technology has facilitated the management of waste collection, which has become a major challenge for local governments along with the increase in population and waste volume. This research aims to optimize waste collection routes using Dijkstra's Algorithm to minimize travel time and operational costs in Ponorogo Regency. The system was developed using the waterfall system development method, which includes the stages of requirements analysis, system design, implementation, and testing. The system is designed to calculate the shortest route from Mrican Landfill to various Temporary Shelters (TPS) and Recycling Centers (PDU) based on field data. With this approach, the system can help optimize the waste transportation process, reduce travel time, and reduce operational costs, so that waste management becomes more efficient and effective. The test results displayed in Figure 9 show that the system successfully determined the shortest waste collection route with a total distance of 16.2 km through route A-B-C-D-E-F. This simulation proves that the system can accurately optimize the waste collection route based on the shortest route with a total distance of 16.2 km through the A-B-C-D-E-F route. Thus, the results of this system are able to significantly reduce the time and cost of waste collection and make a real contribution to improving the efficiency of waste management.

**Keywords**—Dijkstra's Algorithm, Waterfall Method, Shortest Route.

## I. PENDAHULUAN

**P**ERKEMBANGAN teknologi informasi mengalami kemajuan pesat, memudahkan pengguna untuk mendapatkan dan mengelola informasi[1]. Seiring dengan kemajuan teknologi informasi, banyak instansi dan masyarakat maju yang telah memanfaatkan teknologi ini untuk meningkatkan efisiensi dan efektivitas berbagai kegiatan, termasuk dalam bidang pengelolaan sampah[2][3]. Salah satu tantangan utama dalam pengelolaan sampah adalah menentukan rute pengumpulan yang paling efisien. Penentuan rute yang efisien tidak hanya membantu mengurangi biaya operasional, tetapi juga mengurangi waktu tempuh dan bahan bakar[4]. Optimalisasi pengangkutan sampah dari Tempat Penampungan Sementara (TPS) ke Tempat Pemrosesan Akhir (TPA) merupakan bagian penting dari tantangan ini[2]. Dinas Lingkungan Hidup sering kali dihadapkan dengan keterbatasan jumlah alat

transportasi yang tersedia untuk mengangkut sampah, sementara volume sampah yang harus diangkut terus meningkat seiring pertumbuhan populasi dan aktivitas masyarakat. Hal ini menuntut adanya solusi yang inovatif untuk mengatasi keterbatasan ini dan memastikan sampah dapat diangkut dan diproses tepat waktu[5].

Permasalahan dalam menentukan rute pengumpulan sampah sering kali disebabkan oleh berbagai faktor seperti kemacetan lalu lintas, kondisi jalan yang tidak menentu, dan distribusi titik pengumpulan sampah yang tidak merata[6]. Salah satu metode yang telah terbukti efektif dalam menentukan rute terpendek adalah Algoritma *Dijkstra*[7]. Algoritma ini membutuhkan parameter tempat asal dan tempat tujuan, dan hasil akhirnya adalah jarak terpendek dari tempat asal ke tempat tujuan beserta rutenya[8].

Beberapa penelitian terdahulu yang mengaplikasikan Algoritma *Dijkstra* dalam konteks transportasi dan logistik menunjukkan hasil yang efisien. Penelitian oleh Lucky Hapsari (2019) menunjukkan bahwa penggunaan algoritma *Dijkstra* dalam pencarian rute terpendek truk pengangkut sampah dan meningkatkan efisien secara signifikan[9]. Selain itu, penelitian oleh Maria Bunaen et al (2022) membuktikan bahwa integrasi Algoritma *Dijkstra* dengan sistem informasi geografis (GIS) dapat meningkatkan akurasi dan efisiensi dalam menentukan rute terpendek dari pusat kota Surabaya ke tempat bersejarah[10]. Penelitian lainnya oleh Widiyanto et al (2024) berhasil merekomendasikan lokasi wisata terdekat dan menentukan rute terpendek ke lokasi wisata terdekat[11]. Berdasarkan penelitian-penelitian terdahulu, penelitian ini memberikan inovasi dalam penerapan Algoritma *Dijkstra* untuk menentukan rute terpendek dari Tempat Pembuangan Akhir (TPA) ke Tempat Pembuangan Sementara (TPS) guna mengurangi waktu tempuh dan biaya operasional bagi petugas pengangkut sampah di Kabupaten Ponorogo.

## II. TINJAUAN PUSTAKA

### A. Algoritma *Dijkstra*

Algoritma *Dijkstra* adalah salah satu algoritma pencarian jalur terpendek yang paling terkenal dan banyak digunakan dalam pemrosesan graf. Dikembangkan oleh Edsger W. Dijkstra pada tahun 1956 dan dipublikasikan tiga tahun kemudian, algoritma ini dirancang untuk menemukan jalur terpendek dari satu simpul (node) sumber ke semua simpul lain dalam graf berarah berbobot yang tidak mengandung bobot negatif. Keunggulan utama dari algoritma *Dijkstra* adalah kesederhanaan dan efisiensinya dalam menyelesaikan masalah rute terpendek, yang menjadikannya sangat populer dalam berbagai aplikasi, termasuk jaringan komunikasi, transportasi, dan sistem navigasi. Algoritma *Dijkstra* beroperasi dengan prinsip *greedy*, di mana pada setiap langkahnya, algoritma memilih simpul dengan jarak terpendek yang belum diproses, kemudian memperbarui jarak ke simpul-simpul tetangganya jika ditemukan jalur yang lebih pendek melalui simpul tersebut. Proses ini dimulai dengan menginisialisasi jarak semua simpul ke nilai tak terhingga, kecuali simpul awal yang diinisialisasi ke nol. Simpul terdekat yang belum diproses kemudian dipilih, jarak-jaraknya diperbarui, dan simpul tersebut ditandai sebagai telah diproses. Proses ini berlanjut hingga semua simpul diproses atau hingga jarak terpendek ke simpul tujuan ditemukan[12].

Algoritma *Dijkstra* efektif digunakan dalam graf yang memiliki bobot non-negatif dan sering diterapkan dalam berbagai aplikasi praktis, seperti navigasi GPS, jaringan komputer, dan perencanaan rute logistik. Algoritma ini tidak hanya mampu menentukan jalur terpendek dengan efisien tetapi juga mengoptimalkan pengelolaan rute dalam berbagai konteks, termasuk pengurangan waktu tempuh dan biaya operasional. Misalkan  $G$  adalah graf berarah dengan titik-titik  $V(G) = \{v_1, v_2, \dots, v_n\}$  dan kita ingin mencari jalur terpendek dari  $v_1$  ke  $v_n$ . Algoritma *Dijkstra* dimulai dari titik awal  $v_1$ , kemudian mencari titik  $v_n$  dengan total bobot terkecil. Dalam algoritma ini,  $L(v)$  menyatakan label dari *vertex*  $v$ . Setiap *vertex* dalam himpunan memiliki label sementara (T) dan label tetap (label yang dilingkari). Proses dimulai dengan menginisialisasi jarak dari  $v_1$  ke dirinya sendiri sebagai 0 dan jarak ke semua titik lainnya sebagai tak terhingga. Kemudian, dari titik dengan jarak terkecil yang belum diproses, algoritma memperbarui jarak ke titik-titik tetangganya jika ditemukan jalur yang lebih pendek melalui titik tersebut. Proses ini berlanjut hingga semua titik telah diberi label tetap atau titik tujuan  $v_n$  ditemukan dengan jumlah bobot terkecil[4].

## B. Shortest Path

Teori jalur terpendek (*Shortest Path*) dalam graf bertujuan untuk menemukan rute dengan total bobot terkecil dari satu titik awal ke satu titik tujuan di dalam graf. Graf tersebut bisa berarah atau tak berarah, dengan bobot pada setiap sisi yang mewakili jarak atau biaya. Berbagai algoritma umum digunakan untuk menemukan jalur terpendek, termasuk algoritma *Dijkstra*, *Bellman-Ford*, dan *Floyd-Warshall*. Algoritma *Dijkstra*, misalnya, memulai dari titik awal, kemudian memilih titik dengan jarak terpendek yang belum diproses, dan memperbarui jarak ke titik-titik tetangganya jika ditemukan jalur yang lebih pendek melalui titik tersebut. Proses ini berlanjut hingga jalur terpendek ke semua titik atau ke titik tujuan ditemukan.

Algoritma *Bellman-Ford* digunakan untuk graf yang mungkin mengandung bobot negatif, meskipun lebih lambat dari *Dijkstra*. Algoritma ini memperbarui jarak dengan melakukan relaksasi pada setiap sisi, berulang kali untuk semua titik. *Floyd-Warshall* adalah algoritma lain yang digunakan untuk menemukan jalur terpendek antara semua pasangan titik dalam graf, melalui pendekatan pemrograman dinamis. Teori jalur terpendek memiliki banyak aplikasi praktis, seperti dalam sistem navigasi GPS untuk menemukan rute tercepat, perencanaan jaringan untuk meminimalkan biaya, dan optimasi rute dalam logistik untuk meningkatkan efisiensi pengiriman barang[13].

## C. Teori Graf

Teori graf dalam konteks rute terpendek adalah studi tentang cara menemukan jalur dengan bobot total terkecil dari satu simpul awal ke simpul tujuan dalam sebuah graf. Graf ini terdiri dari himpunan simpul (*vertex*) dan himpunan sisi (*edge*) yang menghubungkan pasangan simpul tersebut. Dalam graf berarah (*directed graph*), setiap sisi memiliki arah tertentu dari satu simpul ke simpul lainnya, sementara dalam graf tak berarah (*undirected graph*), sisi-sisi tidak memiliki arah. Graf dapat berbobot (*weighted graph*) jika setiap sisi memiliki nilai atau bobot yang mewakili jarak, biaya, atau waktu.

Dalam konteks rute terpendek, konsep penting termasuk derajat (*degree*) dari sebuah simpul, yaitu jumlah sisi yang terhubung ke simpul tersebut, serta jalur (*path*), yang merupakan urutan simpul-simpul yang dilalui oleh serangkaian sisi. Algoritma seperti *Dijkstra*, *Bellman-Ford*, dan *Floyd-Warshall* digunakan untuk mencari jalur terpendek. Algoritma *Dijkstra*, misalnya, dimulai dengan inisialisasi jarak dari simpul awal ke dirinya sendiri sebagai 0 dan jarak ke simpul lainnya sebagai tak terhingga. Algoritma ini kemudian memilih simpul dengan jarak terkecil yang belum diproses dan memperbarui jarak ke simpul-simpul tetangganya. Proses ini berlanjut hingga jalur terpendek ke semua simpul atau ke simpul tujuan ditemukan. Aplikasi teori graf dalam pencarian rute terpendek mencakup navigasi GPS, optimasi rute logistik, dan perencanaan jaringan[5].

## D. Blackbox Testing

*Black Box Testing* adalah salah satu metode pengujian perangkat lunak yang digunakan untuk menguji fungsionalitas suatu sistem tanpa mengetahui detail internal atau struktur kode program. Dalam pendekatan ini, penguji hanya fokus pada input dan output dari sistem untuk memastikan bahwa aplikasi berfungsi sesuai dengan spesifikasi yang telah ditentukan. Pengujian ini biasanya dilakukan dengan mengidentifikasi berbagai skenario penggunaan yang mungkin dihadapi oleh pengguna akhir dan kemudian menguji bagaimana sistem menangani skenario tersebut[14]. Akurasi pengujian *Black Box* mengacu pada kemampuan pengujian untuk mendeteksi dan memverifikasi bahwa sistem perangkat lunak bekerja sesuai dengan spesifikasi yang diharapkan. Dalam konteks ini, akurasi diukur dengan membandingkan hasil yang diharapkan dengan hasil yang sebenarnya diperoleh dari pengujian terhadap berbagai skenario. Pengujian *Black Box* berfokus pada input dan output, dan akurasinya menunjukkan sejauh mana sistem mampu memberikan output yang benar untuk setiap set input yang diberikan, tanpa memperhatikan bagaimana hasil tersebut dihasilkan di dalam kode[15].

Berikut rumus akurasi :

$$\text{Akurasi Pengujian} = \frac{\Sigma \text{Pengujian Valid}}{\Sigma \text{Total Valid}} \times 100\% \quad (1)$$

### E. Metode Waterfall

Metode *Waterfall* adalah salah satu model proses pengembangan perangkat lunak yang paling tradisional dan banyak digunakan. Metode ini bersifat linear dan sistematis, di mana setiap tahap harus diselesaikan sebelum tahap berikutnya dimulai. Model ini menyerupai air terjun, dengan tiap tahapan mengalir secara berurutan dari tahap satu ke tahap berikutnya [16] :

a) *Requirement Analysis (Analisis Kebutuhan)*

Pada tahap ini, kebutuhan pengguna dan sistem diidentifikasi dan didokumentasikan secara lengkap. Semua persyaratan fungsional dan non-fungsional dirinci untuk memastikan pengembangan yang tepat.

b) *Design (Desain)*

Berdasarkan hasil analisis kebutuhan, tahap ini berfokus pada perancangan arsitektur sistem dan desain yang mendetail, termasuk desain database, antarmuka pengguna, dan komponen perangkat lunak lainnya.

c) *Implementation (Implementasi)*

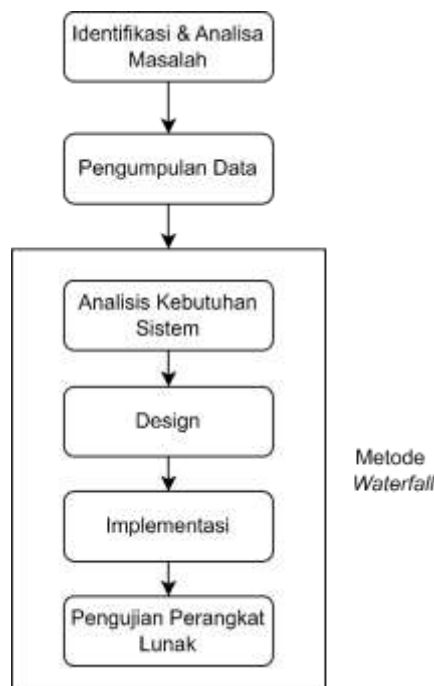
Pada tahap ini, desain yang telah disusun diubah menjadi kode program. Setiap komponen sistem dikembangkan sesuai dengan spesifikasi yang telah ditentukan pada tahap desain.

d) *Testing (Pengujian Perangkat Lunak)*

Setelah implementasi, semua komponen perangkat lunak yang telah dikembangkan diuji secara menyeluruh. Pengujian dilakukan untuk memastikan bahwa sistem berfungsi sesuai dengan kebutuhan yang telah ditetapkan dan bebas dari kesalahan.

### III. METODE PENELITIAN

Dalam implementasi Algoritma *Dijkstra* dalam menentukan rute terpendek pada TPA Mrican, digunakan metode *Waterfall* sebagai pendekatan pengembangan sistem. Metode *Waterfall* memungkinkan pengembangan aplikasi secara terstruktur dan bertahap[17]. Berikut *Flowchart* tahapan penelitian :



Gambar 1. Tahapan Penelitian

#### A. Identifikasi & Analisa Masalah

Pada tahap ini, dilakukan identifikasi dan analisis terhadap masalah yang dihadapi dalam penentuan rute pengangkutan sampah di TPA Mrican. Ini termasuk mengidentifikasi kendala-kendala seperti kemacetan lalu lintas, kondisi jalan yang berubah-ubah, dan distribusi lokasi pengumpulan sampah yang tidak merata. Tujuan dari tahap ini adalah memahami secara mendalam permasalahan yang ada agar dapat dirumuskan solusi yang tepat.

## B. Pengumpulan Data

Tahap ini melibatkan pengumpulan data yang diperlukan untuk pengembangan sistem, meliputi peta kota yang rinci, lokasi tempat pembuangan sampah (TPS), jadwal pengangkutan, serta data jalan, termasuk panjang jalan, kondisi jalan, dan lalu lintas. Berdasarkan wawancara dengan Pak Angga, terdapat 9 TPS yang tersebar di seluruh kota, dengan setiap kecamatan memiliki 1 TPS. Adapun jadwal pengangkutan mencakup waktu, frekuensi, dan durasi pengangkutan, yaitu rata-rata 1 jam untuk lokasi dekat dan 2-3 jam untuk lokasi jauh. Kesembilan TPS tersebut adalah TPS 3R Kertosari Sehat, TPS 3R KSM "PEKA" Paju, TPST Stasiun Banyudono, PDU Purbosuman, TPS 3R SAHANG, TPS 3R KUTUWETAN, TPS 3R GONTOR, TPST Tonatan, dan TPS 3R SUMUR WOLU. Informasi tentang jumlah armada pengangkut sampah, yakni 3 dam truk dan 10 kontainer, dengan setiap dam truk memiliki 1-2 unit kontainer, juga sangat penting untuk mengatur jadwal dan rute pengangkutan secara efisien. Data ini dapat dikumpulkan melalui survei lapangan, wawancara dengan dinas kebersihan, dan penggunaan sumber daya GIS, sehingga dengan data yang lengkap dan akurat, sistem akan mampu menentukan rute pengangkutan sampah yang paling efisien.

## C. Analisis Kebutuhan Sistem

Pada tahap ini, dilakukan analisis terhadap kebutuhan sistem yang akan dibangun. Semua kebutuhan fungsional dan non-fungsional diidentifikasi dan didokumentasikan. Kebutuhan fungsional untuk admin mencakup kemampuan untuk *login*, mengelola pengguna, mengelola data TPS, melihat informasi geografis, dan *logout*. Admin harus dapat menambahkan, mengedit, atau menghapus data pengguna dan TPS, serta memperbarui informasi terkait. Kebutuhan non-fungsional mencakup keamanan sistem melalui autentikasi dan otorisasi, kinerja yang responsif, ketersediaan sistem yang tinggi, dan antarmuka pengguna yang intuitif. Dokumentasi yang memadai juga diperlukan untuk membantu admin dalam mengoperasikan sistem. Dengan mengidentifikasi kebutuhan ini, diharapkan sistem dapat memenuhi kebutuhan pengelolaan TPA Mrican secara efisien.

## D. Design

Tahap desain melibatkan pembuatan arsitektur sistem dan perancangan komponen-komponen perangkat lunak, termasuk desain *database* untuk menyimpan data rute, algoritma *Dijkstra* untuk menghitung rute terpendek, dan antarmuka pengguna. Desain ini harus memenuhi semua kebutuhan yang telah diidentifikasi pada tahap sebelumnya. Dalam desain ini, *Flowchart* login & registrasi menggambarkan alur registrasi dan login pada sistem penentuan rute terpendek. *Flowchart* algoritma *Dijkstra* menjelaskan langkah-langkah detail dalam menghitung rute terpendek, termasuk inisialisasi, pembaruan jarak, dan penentuan simpul berikutnya hingga rute terpendek ditemukan. *Use case diagram* menunjukkan interaksi antara admin dan sistem dalam mengelola data TPS, menghitung rute, dan melihat informasi geografis. Desain *user interface* mencakup tampilan antarmuka yang intuitif untuk login, mengelola pengguna dan data TPS, serta menampilkan peta geografis dengan rute terpendek yang dihitung. Seluruh elemen desain ini diintegrasikan untuk memastikan sistem berfungsi secara efisien dan memenuhi kebutuhan pengguna.

## E. Implementasi

Pada tahap implementasi, desain yang telah dibuat diterjemahkan ke dalam kode program. Setiap komponen perangkat lunak dikembangkan sesuai dengan spesifikasi desain, melibatkan pengkodean Algoritma *Dijkstra* dalam bahasa pemrograman yang dipilih, integrasi dengan database untuk menyimpan data rute jalan, lokasi TPS, dan informasi pengangkutan sampah, serta pengembangan antarmuka pengguna menggunakan teknologi *front-end*. Pengujian fungsional dilakukan untuk memastikan semua fitur bekerja sesuai dengan kebutuhan, dan pengujian integrasi dilakukan untuk memastikan semua komponen sistem berfungsi dengan baik secara bersama-sama.

## F. Pengujian Perangkat Lunak

Tahap pengujian perangkat lunak dilakukan untuk memastikan bahwa sistem yang dibangun berfungsi sesuai dengan kebutuhan yang telah ditetapkan. Pengujian ini mencakup pengujian fungsional untuk memeriksa setiap fitur sistem, serta pengujian kinerja untuk memastikan sistem dapat menangani beban yang diharapkan. Salah satu metode yang digunakan adalah *black box testing*, yang berfokus pada pengujian fungsionalitas sistem tanpa melihat ke dalam struktur kode internal. *Black box testing* memastikan bahwa input yang diberikan menghasilkan output yang diharapkan sesuai dengan

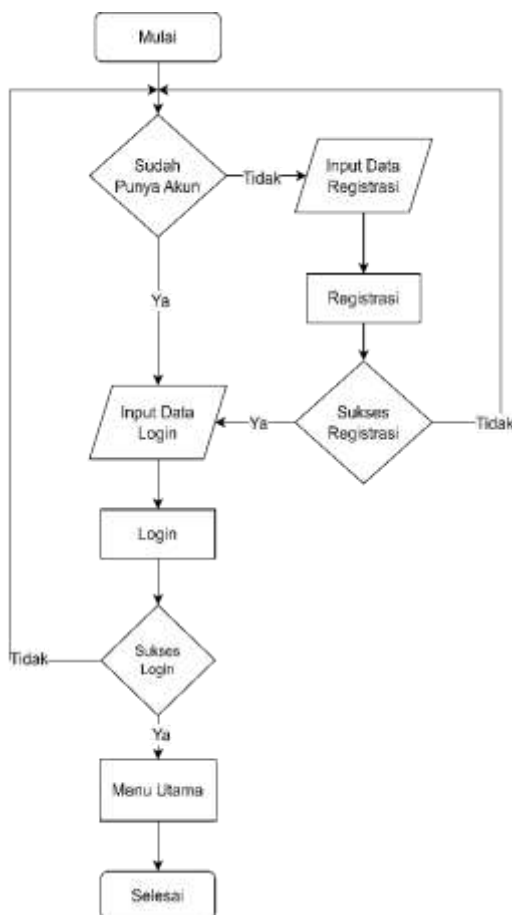
spesifikasi yang telah ditetapkan. Setiap kesalahan atau kekurangan yang ditemukan selama pengujian diperbaiki hingga sistem berfungsi dengan optimal. Pengujian ini membantu memastikan bahwa sistem mampu memenuhi semua persyaratan fungsional dan non-fungsional yang telah diidentifikasi sebelumnya, serta memberikan pengalaman pengguna yang andal dan efisien[18].

#### IV. HASIL DAN PEMBAHASAN

##### A. Hasil Perancangan Sistem

Berikut hasil perancangan sistem pada sistem penentuan Rute Terpendek pada TPA Mrican :

##### a) Flowchart Login & Registrasi

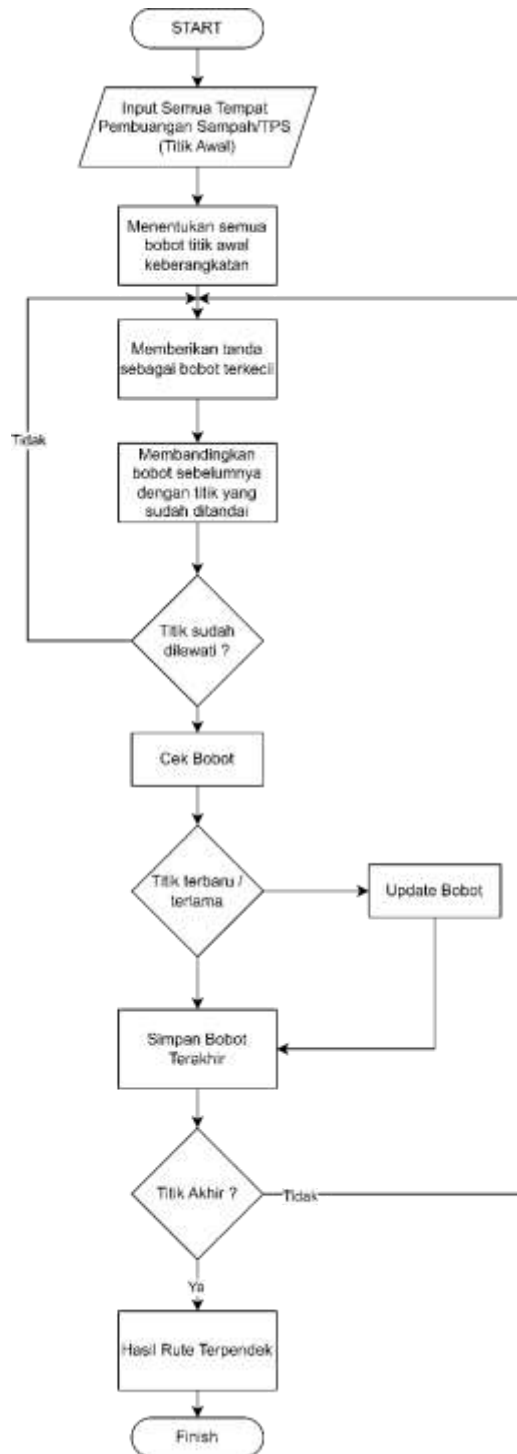


Gambar 2. Flowchart Login & Registrasi

Gambar 2 menunjukkan *flowchart* proses login dan registrasi untuk sistem. Proses dimulai dengan pengecekan apakah pengguna sudah memiliki akun atau belum. Jika pengguna belum memiliki akun, mereka harus menginput data registrasi dan melakukan registrasi. Setelah registrasi berhasil, pengguna dapat melanjutkan ke langkah login. Jika sudah memiliki akun, pengguna langsung menginput data login. Sistem kemudian memverifikasi keberhasilan login. Jika login sukses, pengguna diarahkan ke menu utama. Jika tidak, pengguna harus menginput ulang data login. *Flowchart* ini memastikan bahwa hanya pengguna terdaftar yang dapat mengakses sistem dengan memastikan proses registrasi dan login yang valid.

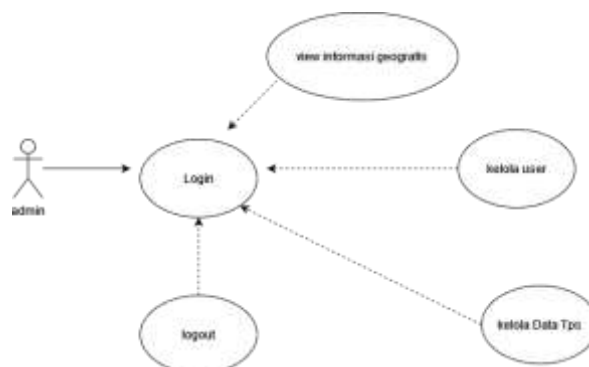
##### b) Flowchart Algoritma Dijkstra

Gambar 3 menunjukkan *flowchart* Algoritma Dijkstra yang digunakan untuk menentukan rute terpendek. Proses dimulai dengan menginput semua tempat pembuangan sampah (TPS) sebagai titik awal. Selanjutnya, algoritma menentukan bobot awal dari setiap titik dan menandai titik dengan bobot terkecil. Kemudian, dilakukan perbandingan bobot antara titik sebelumnya dengan titik yang sudah ditandai. Jika titik tersebut belum dilewati, bobot akan diperbarui. Setelah memeriksa dan memperbarui bobot, algoritma menyimpan bobot terakhir dan mengecek apakah titik akhir telah tercapai. Jika belum, proses berlanjut; jika sudah, algoritma menghasilkan rute terpendek. *Flowchart* ini menggambarkan langkah-langkah iteratif dalam Algoritma Dijkstra untuk menemukan rute terpendek secara sistematis.



Gambar 3. Flowchart Algoritma Dijkstra

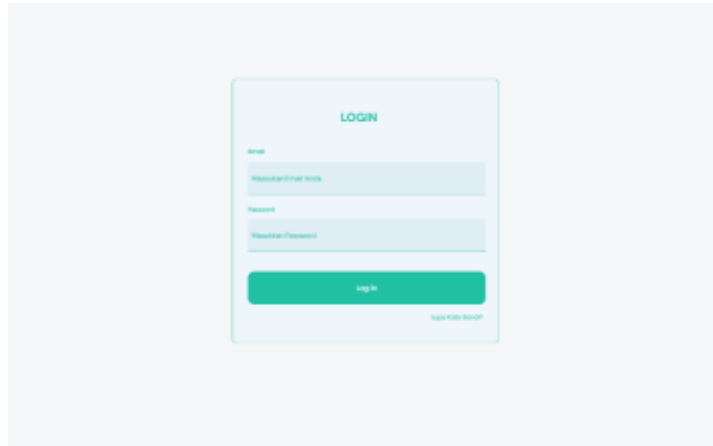
c) Use Case Diagram



Gambar 4. Use Case Diagram

Gambar 4 menunjukkan *use case diagram* untuk sistem penentuan rute terpendek di TPA Mrican. Diagram ini menggambarkan interaksi antara admin dan sistem. Admin dapat melakukan beberapa fungsi utama, yaitu login ke dalam sistem, mengelola pengguna (menambah, mengedit, menghapus), mengelola data TPS (menambah, mengedit, menghapus data TPS), melihat informasi geografis yang menunjukkan lokasi TPS dan rute pengangkutan, serta logout dari sistem. Use case diagram ini membantu memvisualisasikan fungsionalitas utama yang disediakan oleh sistem dan bagaimana admin dapat berinteraksi dengan masing-masing fungsi tersebut.

d) *User Interface*



Gambar 5. Tampilan Halaman Login

Pada Gambar 5, terdapat tampilan *login* dimana user harus memasukkan *email* dan *password* yang benar untuk masuk ke halaman sistem.



Gambar 6. Tampilan Halaman Input Lokasi

Gambar 6 menampilkan antarmuka pengguna untuk form pengisian data tempat pembuangan sampah. *Form* ini berisi beberapa input *field* yang perlu diisi oleh pengguna, termasuk *latitude*, *longitude*, titik awal, dan titik akhir. Selain itu, terdapat opsi *checkbox* untuk menandai apakah pengangkutan bersifat mendesak (*urgent*). Setelah mengisi semua data yang diperlukan, pengguna dapat menyimpan data dengan menekan tombol "Simpan" atau membatalkan pengisian dengan menekan tombol "Batal". Antarmuka ini dirancang untuk memudahkan pengguna dalam memasukkan data terkait lokasi TPS secara efisien.





Gambar 7. Tampilan Halaman Output Lokasi

Gambar 7 menunjukkan rute pembuangan sampah di Ponorogo, Jawa Timur, dari TPS 3R Kertosari Sehat ke TPA Mrican. Rute dimulai dari TPS 3R Kertosari Sehat di Jalan Mayjend Sutoyo, lalu belok kanan ke Jalan Brigjen Katamso, kemudian belok kiri ke Jalan Halim Perdana Kusuma, dan berakhir di TPA Mrica di Jalan Mrica No.1. Rute ini diperkirakan memakan waktu sekitar 13 menit dengan jarak tempuh antara 6,6 km hingga 8,3 km. Peta dan informasi lokasi disediakan untuk memudahkan navigasi dan memastikan pembuangan sampah yang efisien.

e) *Data Lokasi TPS dan TPA di Ponorogo*

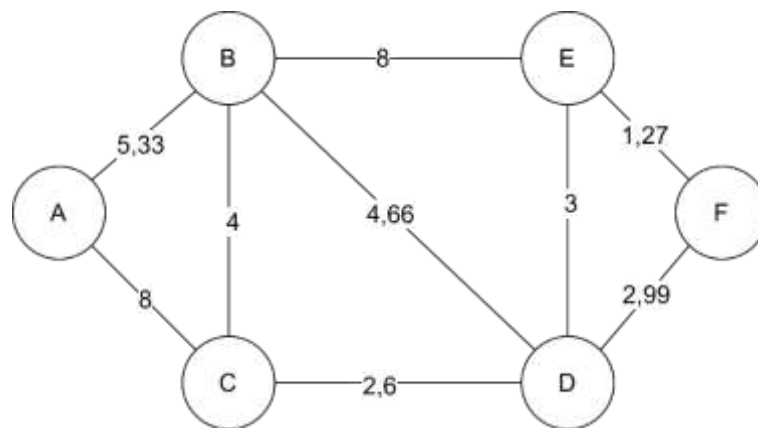
Berikut adalah data sampel untuk lokasi TPA, TPS dan PDU yang diperoleh dari hasil wawancara bapak angga. Data ini mencakup informasi terkait nama tempat dan alamat, yang akan digunakan untuk menganalisis dan mengoptimalkan proses pengambilan sampah dalam penelitian ini.

Tabel 1. Data Lokasi TPS dan TPA di Ponorogo

Nama Tempat	Latitude	Longitude
TPA Mrican	-7.86164203626107	111.52683135145323
TPS 3R Kertosari Sehat	-7.84776292444014	111.48052717788936
TPS 3R KSM "PEKA" Paju	-7.882953391966663	111.45748311580199
TPST Stasiun Banyudono	-7.862247672404003	111.46851759429813
TPST Tonatan	-7.871813654213037	111.49247240541894
PDU Purbosuman	-7.881911997527865	111.48709110778996

f) *Implementasi Algoritma Dijkstra Dengan Sampel*

Dalam system nantinya akan melakukan perhitungan dan menentukan jalur terpendek yang harus dilalui dengan mengimplementasikan Algoritma Dijkstra untuk contoh simulasi penggunaannya akan menggunakan sampel data pengambilan sampah dimulai dari Kode A yaitu TPA Mrican sebagai titik awal. Selanjutnya, kode B merujuk pada TPS 3R Kertosari Sehat, sebuah situs pengelolaan sampah di Kertosari yang berfokus pada prinsip "3R" (*Reduce, Reuse, Recycle*) dengan penekanan pada pengolahan sampah yang bersih dan sehat. Kode C adalah TPS 3R KSM "PEKA" Paju, sebuah fasilitas pengelolaan sampah di Paju yang dikelola oleh KSM (Kelompok Swadaya Masyarakat) "PEKA" dengan prinsip 3R. Kode D adalah TPST (Tempat Pengelolaan Sampah Terpadu) Stasiun Banyudono, yang terletak di dekat stasiun Banyudono dan berfungsi sebagai fasilitas pengelolaan sampah terpusat. Kode E merujuk pada TPST Tonatan, sebuah situs pengelolaan sampah terintegrasi di area Tonatan, tempat sampah kemungkinan disortir, diproses, atau didaur ulang. Terakhir, Kode F adalah PDU (Pengolahan Daur Ulang) Purbosuman, sebuah fasilitas daur ulang yang terletak di Purbosuman dan berfokus pada pengolahan serta daur ulang bahan-bahan sampah. Algoritma akan mengkalkulasi rute terpendek yang menghubungkan lokasi-lokasi ini berdasarkan jarak antar titik untuk menentukan urutan optimal pengambilan sampah.



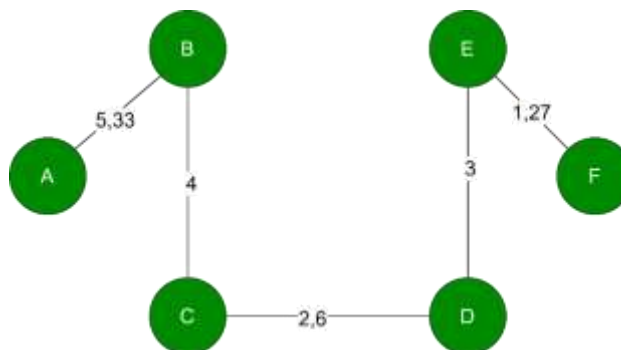
Gambar 8. Graf

Berikut adalah keterangan lengkap untuk setiap kode :

- A = TPA Mrican
- B = TPS 3R Kertosari Sehat
- C = TPS 3R KSM "PEKA" Paju
- D = TPST Stasiun Banyudono
- E = TPST Tonatan
- F = PDU Purbosuman

Tabel 2. Perhitungan manual pengujian algoritma dijkstra

Kode	A	B	C	D	E	F
A	0	5,33 KM	8 KM	$\infty$	$\infty$	$\infty$
B	5,33 KM	0	4 KM	4,66 KM	8 KM	$\infty$
C	8 KM	4 KM	0	2,6 KM	$\infty$	$\infty$
D	$\infty$	4,66 KM	2,6 KM	0	3 KM	2,99 KM
E	$\infty$	8 KM	$\infty$	3	0	1,27 KM
F	$\infty$	$\infty$	$\infty$	2,99 KM	1,27 KM	0



Gambar 9. Graf Terpendek

Rute Terpendek A-B-C-D-E-F dengan total jarak 16.2 km

g) Pengujian Blackbox

Tabel 3. Hasil Pengujian Blackbox

No.	Skenario	Hasil yang diharapkan	Hasil yang diperoleh	Status
1	Login dengan username dan password yang Berhasil	Pengguna berhasil masuk ke halaman utama sistem.	Pengguna berhasil masuk ke halaman utama sistem.	Valid
2	Login dengan username dan password yang salah atau Gagal	Muncul pesan error yang menyatakan bahwa username atau password salah/gagal.	Muncul pesan error yang menyatakan bahwa username atau password salah/gagal.	Valid

No.	Skenario	Hasil yang diharapkan	Hasil yang diperoleh	Status
3	Menekan tombol dengan mengisi titik awal dalam Menentukan Rute Terpendek pada TPS ke TPA Mrican.	Muncul informasi Rute Terpendek pada TPS ke TPA Mrican pada bagian output.	Muncul informasi Rute Terpendek pada TPS ke TPA Mrican pada bagian output.	Valid
4	Menekan tombol Bersihkan saat pada textbox penentuan rute terpendek masih terisi titik awal dan titik akhir.	Informasi pengiriman terhapus.	Informasi pengiriman terhapus.	Tidak Valid
5	Menekan tombol Tutup	Kembali ke tampilan form utama.	Langsung keluar dari sistem.	Valid

Hasil pengujian menggunakan metode *Black Box* pada sistem penentuan rute terpendek menunjukkan bahwa dari 5 skenario pengujian yang dilakukan, 4 skenario dinyatakan valid dan 1 skenario dinyatakan tidak valid. Dengan demikian, tingkat akurasi pengujian yang diperoleh adalah :

$$\text{Akurasi Pengujian} = \frac{4}{5} \times 100\% = 80\%$$

Hasil ini menunjukkan bahwa sistem memiliki tingkat akurasi 80% berdasarkan pengujian skenario yang telah dilakukan, sebagaimana terlihat pada tabel 3 di atas.

## V. KESIMPULAN DAN SARAN

Berdasarkan hasil pengujian yang telah dilakukan pada Tabel 2, dapat disimpulkan bahwa algoritma Dijkstra secara efektif menentukan rute terpendek untuk pengangkutan sampah dari TPA Mrican menuju berbagai TPS lainnya. Dalam pengujian ini, perhitungan manual menunjukkan bahwa jalur terpendek dimulai dari TPA Mrican (A), melalui TPS 3R Kertosari Sehat (B), TPS 3R KSM 'PEKA' Paju (C), TPST Stasiun Banyudono (D), dan TPST Tonatan (E), sebelum akhirnya mencapai PDU Purbosuman (F). Rute A-B-C-D-E-F, dengan total jarak 16,2 km, dipilih karena secara signifikan mengurangi waktu perjalanan dan biaya operasional. Hasil ini menunjukkan bahwa algoritma Dijkstra tidak hanya mampu menentukan rute terpendek, tetapi juga mendukung optimalisasi dalam pengelolaan rute pengangkutan sampah. Visualisasi yang ditampilkan pada Gambar 7 dan 9 memperkuat temuan ini, menunjukkan bahwa algoritma Dijkstra mampu memberikan solusi rute yang optimal, baik dari segi jarak maupun waktu tempuh. Dengan demikian, algoritma ini terbukti efektif dalam mendukung pengambilan keputusan operasional yang lebih efisien dalam konteks pengelolaan sampah.

Untuk penelitian selanjutnya, disarankan untuk membandingkan kinerja algoritma Dijkstra dengan algoritma A\* dalam menentukan rute terpendek. Algoritma Dijkstra dikenal dengan pendekatannya yang komprehensif dalam menjelajahi semua kemungkinan jalur, sedangkan algoritma A\* menggunakan heuristik tambahan untuk mempercepat proses pencarian dengan fokus pada jalur yang paling mungkin. Perbandingan ini dapat mengevaluasi efisiensi waktu dan akurasi hasil dalam berbagai skenario, seperti pengangkutan sampah atau aplikasi lain yang memerlukan optimasi rute.

## DAFTAR PUSTAKA

- [1] Z. Pratama, D. Hartama, M. Ridwan Lubis, S. Retno Andani, and I. Okta Kirana, "Penerapan Metode Dijkstra untuk Menentukan Jalur Lintasan Terpendek Kota Kisaran Menuju Objek Wisata Simalungun," *Rekayasa Tek. Inform. dan Inf.*, vol. 1, no. 2, pp. 68–77, 2020, [Online]. Available: <https://djournals.com/resolusi>
- [2] L. Suryani and E. Murniyasih, "Pencarian Rute Terpendek Pada Aplikasi Ojek Sampah Dengan Menggunakan Algoritma Dijkstra," *J. Tek. Inf. dan Komput.*, vol. 5, no. 2, p. 385, 2022, doi: 10.37600/tekinkom.v5i2.586.
- [3] A. Cantona, F. Fauziah, and W. Winarsih, "Implementasi Algoritma Dijkstra Pada Pencarian Rute Terpendek ke Museum di Jakarta," *J. Teknol. dan Manaj. Inform.*, vol. 6, no. 1, pp. 27–34, 2020, doi: 10.26905/jtmi.v6i1.3837.
- [4] D. Ariyanto, "Penerapan Algoritma Dijkstra Pada Manajemen Order Pemasangan Jaringan

Internet Wilayah Telkom Madiun,” Universitas Muhammadiyah Ponorogo, 2023.

- [5] N. H. Adi, M. Giatman, W. Simatupang, A. Afrina, and R. Watrianthos, “Penerapan Metode Dijkstra Pada Jalur Distribusi LPG Untuk Penentuan Jarak Terpendek,” *Build. Informatics, Technol. Sci.*, vol. 3, no. 3, pp. 235–243, 2021, doi: 10.47065/bits.v3i3.1052.
- [6] Talenta Arta Deva Victoria and Hermansyah, “Penerapan Algoritma Dijkstra dalam Pemetaan UMKM Berbasis Android,” *Bull. Comput. Sci. Res.*, vol. 3, no. 6, pp. 420–426, 2023, doi: 10.47065/bulletincsr.v3i6.276.
- [7] H. R Saputra, “Penerapan Algoritma Dijkstra untuk Menentukan Rute Terpendek dari Kampus A UIN Raden Fatah ke Tempat Bersejarah di Palembang,” *E-Jurnal Mat.*, vol. 10, no. 3, pp. 173–178, 2021, doi: 10.33369/diophantine.v2i1.28321.
- [8] N. Arif Sudibyo, P. Eka Setyawan, and Y. Putra Surya Rahmad Hidayat, “Implementasi Algoritma Dijkstra Dalam Pencarian Rute Terpendek Tempat Wisata Di Kabupaten Klaten,” *Riemann Res. Math. Math. Educ.*, vol. 2, no. 1, pp. 1–9, 2020.
- [9] L. I. Hapsari, “Penerapan Algoritma Dijkstra Dalam Pencarian Rute Terpendek Truk Pengangkut Sampah (Studi Kasus : Kabupaten Jember),” Universitas Jember, 2019. [Online]. Available: <http://repository.unej.ac.id/>
- [10] M. C. Bunaen, H. Pratiwi, and Y. F. Riti, “Penerapan Algoritma Dijkstra Untuk Menentukan Rute Terpendek Dari Pusat Kota Surabaya Ke Tempat Bersejarah,” *J. Teknol. Dan Sist. Inf. Bisnis*, vol. 4, no. 1, pp. 213–223, 2022, [Online]. Available: <https://media.neliti.com/media/publications/441390-application-of-the-dijkstra-algorithm-to-f1576853.pdf>
- [11] A. Widiyanto, R. Sahara, and C. R. Hassolthine, “Implementasi Dijkstra Pada Aplikasi Pemesanan Tiket Wisata Untuk Rekomendasi Lokasi Wisata Terdekat,” *J. Manaj. Inform. dan Sist. Inf.*, vol. 7, no. 1, pp. 52–61, 2024.
- [12] I. Syarifudin and K. Ramanda, “Penerapan Metode Dijkstra Pada Sistem Informasi Pencarian Jarak Terpendek Menuju Rumah Sakit di Wilayah Jakarta Barat,” *J. Sains Komput. Inform. (J-SAKTI)*, vol. 5, no. 2, pp. 540–550, 2021.
- [13] G. N. Ramadhan and R. K. A. A. S. Bachrun, “Penerapan algoritma Dijkstra Untuk menentukan Rute Terpendek Tempat Tinggal Ke Kampus 2 Uin Sunan Ampel Surabaya,” *Indones. J. Bus. Intell.*, vol. 7, no. 1, pp. 1–6, 2024.
- [14] M. Syarif and E. B. Pratama, “Analisis Metode Pengujian Perangkat Lunak Blackbox Testing Dan Pemodelan Diagram Uml Pada Aplikasi Veterinary Services Yang Dikembangkan Dengan Model Waterfall,” *J. Tek. Inform. Kaputama*, vol. 5, no. 2, pp. 253–258, 2021.
- [15] A. P. Kusuma, M. F. Rahmat, and A. A. Rofiq, “Analisis Pengujian Sistem Pengiriman Barang Menggunakan Black Box Testing,” *J-Intech*, vol. 11, no. 2, pp. 287–293, 2023, doi: 10.32664/j-intech.v11i2.999.
- [16] T. Ardiansah and D. Hidayatullah, “Penerapan Metode Waterfall Pada Aplikasi Reservasi Lapangan Futsal Berbasis Web,” *J. Inf. Technol. Softw. Eng. Comput. Sci.*, vol. 1, no. 1, pp. 6–13, 2023, doi: 10.58602/itsecs.v1i1.8.
- [17] R. R. Az-zahra, T. A. Ramadhani, and R. A. Nuryadin, “Perancangan Sistem Informasi Geografis Pemetaan Layanan Kesehatan Kabupaten Ponorogo Berbasis Website ‘E-HEALTHY,’” *J. Ris. dan Apl. Mhs. Inform.*, vol. 04, no. 04, pp. 767–774, 2023, doi: <https://doi.org/10.30998/jrami.v4i04.8670>.
- [18] A. P. Putra, F. Andriyanto, K. Karisman, T. D. M. Harti, and W. P. Sari, “Pengujian Aplikasi Point of Sale Menggunakan Blackbox Testing,” *J. Bina Komput.*, vol. 2, no. 1, pp. 74–78, 2020, doi: 10.33557/binakomputer.v2i1.757.