

Pencarian Rute Terpendek Kendaraan Pengangkut Sampah dengan Menggunakan Algoritma Dijkstra di Kota Palangka Raya

Sendy Ariska Putra¹⁾, Widiatry²⁾, Abertun Sagit Sahay³⁾, Nova Noor Kamala Sari⁴⁾, Tomas Leonardo⁵⁾

¹⁾²⁾³⁾⁴⁾⁵⁾Jurusan Teknik Informatika, Fakultas Teknik, Universitas Palangka Raya
Jalan Hendrik Timang Kampus Tunjung Nyaho, Palangka Raya

¹⁾sendyariskaputra00@gmail.com

²⁾widiatry@it.upr.ac.id

³⁾abertun@it.upr.ac.id

⁴⁾novanoorks@it.upr.ac.id

⁵⁾tomasleonardo@it.upr.ac.id

Abstrak

Permasalahan sampah yang masih menjadi isu serius di kota-kota besar Indonesia menuntut solusi yang inovatif dan efektif dalam pengelolaannya. Saat ini, sistem pengelolaan sampah perkotaan masih terpusat pada Tempat Pembuangan Akhir (TPA), yang memerlukan peningkatan efisiensi waktu dan biaya. Transportasi sampah menjadi subsistem kritis dalam pengelolaan ini, yang membutuhkan optimasi untuk mempercepat, memudahkan, dan meminimalkan biaya pengangkutan. Penelitian ini mengusulkan pembuatan aplikasi berbasis website untuk mempercepat proses pengangkutan sampah di Kota Palangka Raya. Melalui penerapan algoritma Dijkstra, aplikasi ini bertujuan untuk mencari rute terpendek dalam pengambilan sampah dari pangkalan ke titik sampah pengguna atau Tempat Pembuangan Sementara. Diharapkan sistem ini dapat membantu penyedia jasa sampah dalam mengoptimalkan operasi pengangkutan sampah, sehingga meminimalkan penumpukan sampah yang berdampak negatif pada kesehatan masyarakat dan estetika kota. Metodologi penelitian pada pembuatan sistem ini adalah metode pendekatan waterfall, yang memiliki tahapan yaitu Requirements and Definition yang digambarkan menggunakan Flowchart, Analysis and Definition yang digambarkan dengan UML (Unified Modelling Language), Implementation and Unit Testing dengan menggunakan metode *black box*.

Kata kunci: Dijkstra, Waterfall, PHP

Abstract

The ongoing issue of waste management in major cities of Indonesia demands innovative and effective solutions. Currently, urban waste management systems are predominantly reliant on Landfills, necessitating improvements in time and cost efficiency. Waste transportation stands as a critical subsystem within this management framework, requiring optimization to expedite, streamline, and minimize transportation costs. This research proposes the development of a web-based application to accelerate the waste transportation process in Palangka Raya City. Through the implementation of the Dijkstra algorithm, this application aims to identify the shortest route for waste collection from depots to user waste points or Temporary Disposal Sites. It is anticipated that this system will assist waste service providers in optimizing waste transportation operations, thereby reducing waste accumulation that negatively impacts public health and urban aesthetics. The methodology employed in developing this system follows the waterfall approach, comprising stages such as Requirements and Definition delineated using Flowcharts, Analysis and Definition depicted using Unified Modeling Language (UML), and Implementation and Unit Testing utilizing black box.

Keywords: Dijkstra, Waterfall, PHP

1. PENDAHULUAN

1.1 Latar Belakang

Sampah adalah hasil sisa kegiatan harian manusia dan proses alam yang memiliki bentuk padat. Setiap individu pasti menghasilkan jumlah sampah yang beragam setiap hari. Pertumbuhan tumpukan sampah terus meningkat sejalan dengan pertumbuhan populasi di perkotaan. Peningkatan timbunan sampah ini merupakan akibat dari perubahan kualitas hidup dan perubahan gaya hidup masyarakat. Oleh karena itu, peningkatan volume sampah harus diimbangi dengan peningkatan kualitas pengelolaan sampah, agar dapat menghindari kemungkinan dampak negatif yang mungkin timbul dari keberadaan sampah[1].

Sampai saat ini, permasalahan sampah masih menjadi isu serius di berbagai kota besar di Indonesia. Sistem pengelolaan sampah perkotaan saat ini masih sangat bergantung pada Tempat Pembuangan Akhir (TPA) sebagai pusat pengelolaan sampah, dan perlu peningkatan perhatian terutama dalam hal efisiensi waktu dan biaya. Transportasi sampah merupakan subsistem kritis yang bertujuan mengangkut sampah dari tempat pemindahan atau sumber sampah secara langsung ke Tempat Pembuangan Akhir (TPA). Dengan mengoptimalkan subsistem ini, diharapkan pengangkutan sampah dapat dilakukan dengan lebih mudah, cepat, dan biaya yang lebih terjangkau, dengan akhir tujuan mengurangi penumpukan sampah yang berdampak langsung pada kesehatan masyarakat dan estetika kota. Pengurangan jarak dan waktu tempuh dianggap sebagai solusi utama dalam perencanaan rute pengangkutan sampah, sehingga rute yang dibuat dapat menjadi yang paling optimal dalam hal efektivitas dan efisiensi [2]. Berdasarkan pemikiran tersebut maka pembuatan aplikasi berbasis website perlu dilakukan untuk mempercepat proses pengangkutan sampah di Kota Palangka Raya dan sampah bisa secara cepat diangkut oleh truk pengangkut sampah ke Tempat Pembuangan Akhir (TPA) penyedia jasa dapat memperoleh

Berdasarkan latar belakang tersebut, akan dirancang suatu sistem “Penerapan Algoritma Dijkstra Dalam Pencarian Rute Terpendek Pengambilan Sampah Di Kota Palangka Raya” yang nantinya diharapkan dapat mengimplementasikan algoritma Dijkstra dalam operasi pengambilan sampah dari pangkalan ke titik sampah user atau ke Tempat Pembuangan Sementara.

1.2 Tujuan Penelitian

Adapun Tujuan dari perancangan dan pembangunan sistem ini adalah menentukan rute terpendek berdasarkan graf yang ada, dan memfasilitasi truk pengangkut sampah untuk mengetahui rute terpendek antara Pangkalan ke TPS/titik sampah Customer.

2. TINJAUAN PUSTAKA

Tinjauan pustaka digunakan sebagai pembanding dan acuan untuk pengembangan sistem. Penelitian ini menggunakan beberapa sumber pustaka yang berhubungan dengan kasus yang akan diteliti. Penelitian oleh Lucky Indrayu Hapsari (2017) dari Universitas Jember dengan judul Penerapan Algoritma Dijkstra Dalam Pencarian Rute Terpendek Truk Pengangkut Sampah (Studi Kasus : Kabupaten Jember). Penelitian ini dilakukan dengan tujuan mengimplementasikan algoritma dijkstra untuk mencari rute terpendek truk pengangkut sampah di Kabupaten Jember. Metodologi penelitian yang digunakan adalah waterfall dengan tahapan analisis kebutuhan, pembuatan desain, penulisan kode program, dan pengujian sistem. Perancangan sistem menggunakan UML [3].

Penelitian oleh Raafiud Fauzi Nuuryagandhi (2016) dari Universitas Negeri Semarang dengan judul Implementasi Algoritma Dijkstra Untuk Menentukan Rute Terpendek Lokasi Objek Wisata Di Kabupaten Pati Berbasis Web. Tujuan dari penelitian ini adalah menerapkan Algoritma Dijkstra dalam menentukan pencarian rute terpendek. Metodologi yang digunakan pada penelitian ini adalah metode waterfall dengan tahapan analisis kebutuhan, pembuatan desain, penulisan kode program, dan pengujian sistem. Perancangan sistem menggunakan Data Flow Diagram (DFD) [4].

Penelitian oleh Aldy Cantona, Fauziah, Winarsih (2020) dari Universitas Nasional. Penelitian ini berjudul Implementasi Algoritma Dijkstra Pada Pencarian Rute Terpendek ke Museum di Jakarta. Persoalan jarak dan awamnya pencarian rute adalah masalah setiap orang ketika mengunjungi tempat-tempat tertentu. Pada penelitian ini, algoritma Dijkstra digunakan untuk mengkalkulasi jarak paling dekat dari satu titik ke museum yang dipilih menjadi

tujuan. Langkah-langkah dari metodologi penelitian ini adalah pembuatan flowchart dan perancangan aplikasi. Pembuatan aplikasi dilakukan dengan melakukan input coding system menggunakan bahasa Dart (Junanda, Kurniadi, & Huda, 2016). Sehingga setiap sistem akan menjalankan fungsinya masing-masing dengan output yang akan berjalan sesuai pada rancangan aplikasi yang telah [5].

3. METODE PENELITIAN

Pendekatan yang digunakan dalam *System Development Life Cycle* (SDLC) adalah metode *waterfall* dari Ian Sommerville 2011. Adapun penjelasan dari tahapan-tahapan metode *waterfall* menurut Ian Sommerville tersebut sebagai berikut.

1. Requirement Analysis and Definition

Requirement Analysis and Definition adalah tahapan penetapan fitur, kendala dan tujuan sistem melalui konsultasi dengan pengguna sistem. Analisis yang dilakukan pada penelitian ini menggunakan analisis sistem lama atau sistem yang sedang berjalan sekarang dan analisis rekomendasi sistem baru yang akan dibuat.

2. System and Software Design

Pada Tahap *System and Software Design* ini akan dibentuk suatu arsitektur sistem berdasarkan persyaratan yang telah ditetapkan. Pada tahapan ini akan dibuat desain dari sistem yang akan dikembangkan. Desain sistem meliputi desain sistem dan desain user interface sistem. Pada tahapan desain sistem pemodelan yang digunakan adalah UML (*Unified Modeling Language*). Pada penelitian ini diagram yang akan dibuat adalah *Class Diagram*, *Use Case Diagram*, *Activity Diagram*, dan desain *user interface*.

3. Implementation and Unit Testing

Dalam tahapan *Implementation and Unit Testing* ini, hasil dari desain perangkat lunak akan direalisasikan sebagai satu set program atau unit program. Setiap unit akan diuji apakah sudah memenuhi spesifikasinya. Pada penelitian ini, website akan dikerjakan dan dikembangkan menggunakan Framework Codeigniter 3 dan untuk membuat sistem database digunakan MySQLi.

4. Integration and System Testing

Dalam tahap *Integration and System Testing* ini, setiap unit program akan diintegrasikan satu sama lain dan diuji sebagai satu sistem yang utuh untuk memastikan sistem sudah memenuhi persyaratan yang ada. Setelah itu sistem akan dikirim ke pengguna sistem. Dalam penelitian ini, tahapan pengujian sistem dilakukan dengan metode testing menggunakan metode Blackbox.

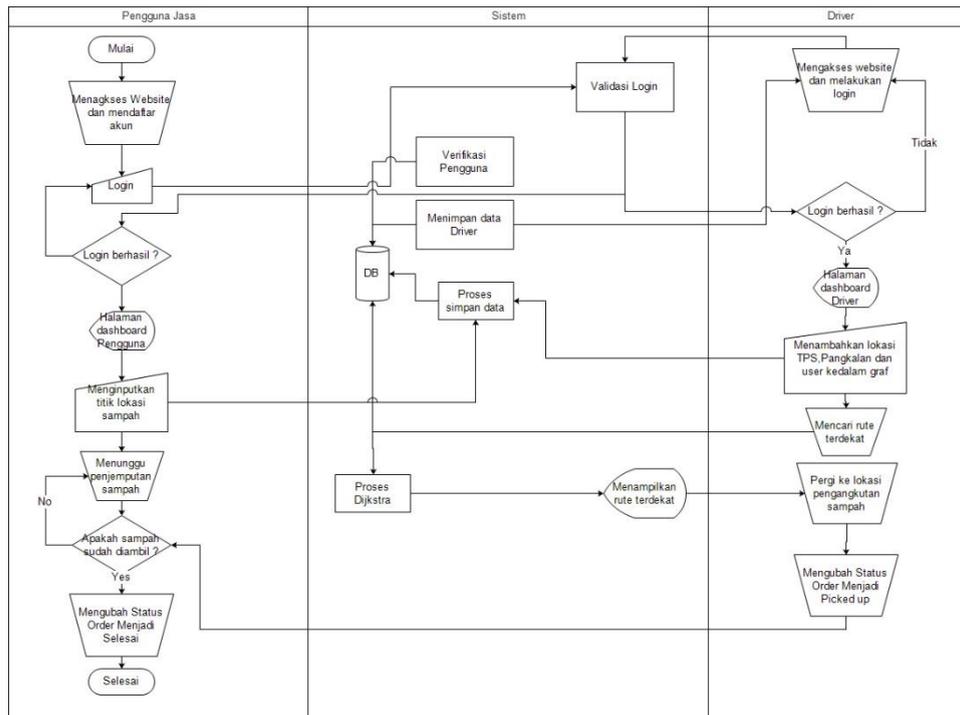
5. Operation and Maintenance

Dalam tahap *Operation and Maintenance* ini, sistem diinstal dan mulai digunakan. Selain itu juga memperbaiki error yang tidak ditemukan pada tahap pembuatan. Dalam tahap ini juga dilakukan pengembangan sistem seperti penambahan fitur dan fungsi baru.

4. PEMBAHASAN DAN IMPLEMENTASI

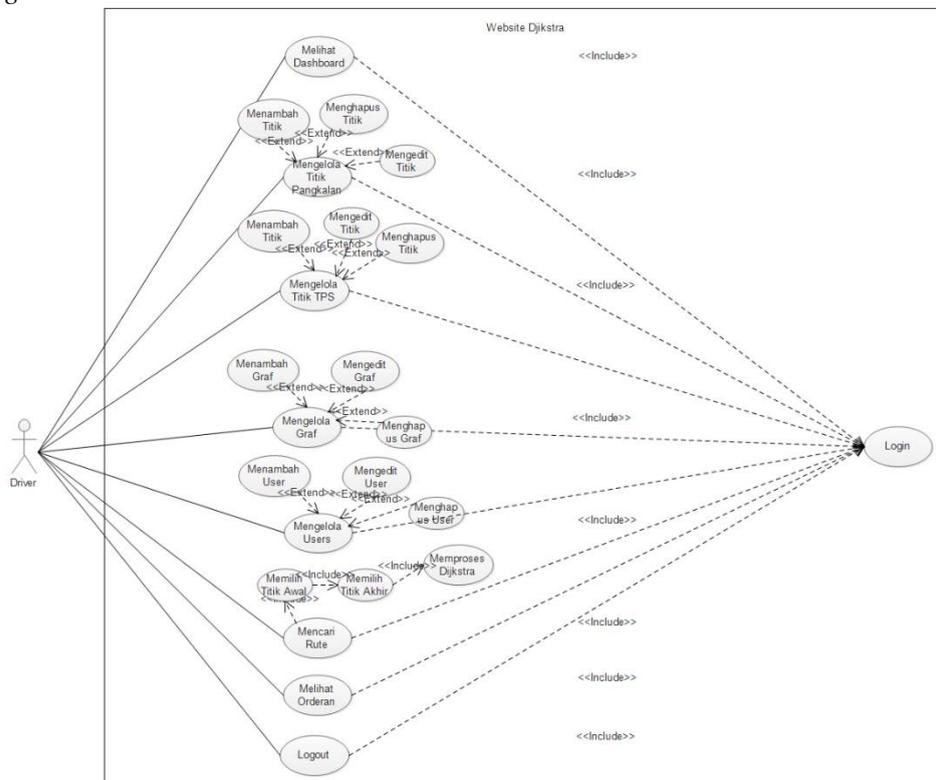
4.1. Analisis Sistem

Tahap analisis dan definisi terbagi menjadi analisis sistem, *Use Case Diagram*, *Activity Diagram*, dan *Class Diagram*. Analisis yang sudah dilakukan pada sistem yang akan dikembangkan dapat digambarkan pada *flowchart* dibawah ini



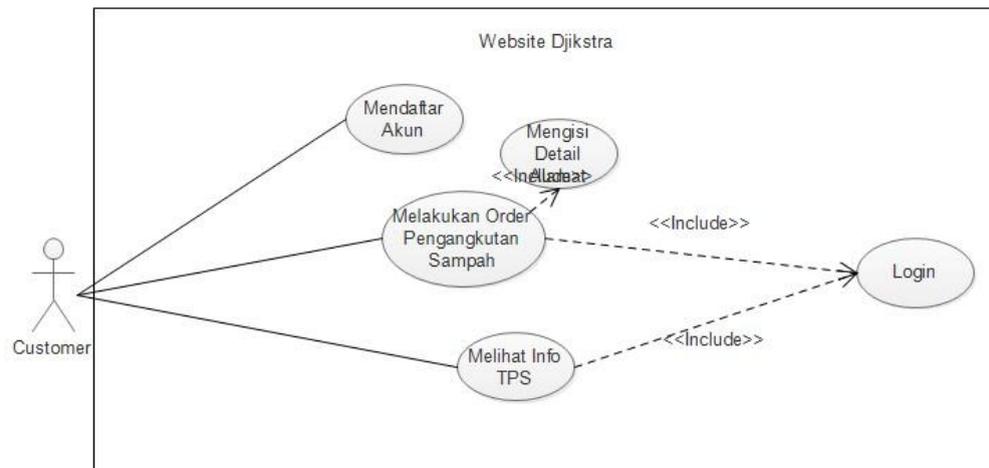
Gambar 1. Flowchart Sistem Baru

Pada sistem baru terdapat 2 aktor yaitu *Driver* dan *Customer*. Berikut merupakan gambar dari use case diagram *Driver*.



Gambar 2. Use Case Diagram Admin

Berikut merupakan use case diagram *Customer*

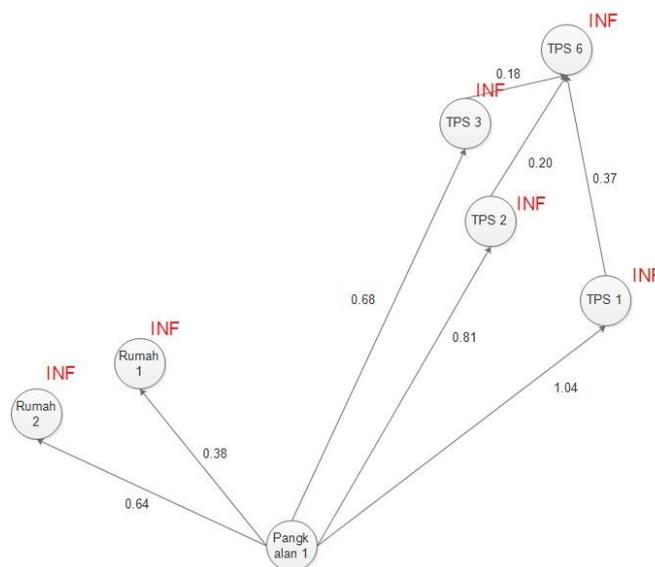


Gambar 3. Use Case Diagram Customer

Tabel atau tempat penyimpanan (storage) yang digunakan pada pembuatan website ini terdapat sebanyak 4 tabel yang akan digunakan. Tabel yang digunakan adalah Tbl_graph, Tbl_node, Tbl_order, Tbl_user.

4.2. Implementasi Algoritma Dijkstra

Pada perhitungan manual ini akan dibuat sebuah graf yang terdiri dari 7 node / vertex, masing-masing vertex ini akan terhubung ke vertex yang lainnya, dan memiliki bobot. Berdasarkan graf tersebut akan dicari lokasi terdekat berdasarkan titik awal dan titik akhir. Berikut gambar 4.16 merupakan graf yang akan digunakan untuk menghitung jarak terdekat menggunakan algoritma Dijkstra..



Gambar 4. Graf

Prosedur algoritma Dijkstra dapat dilakukan dengan langkah-langkah berikut:

1. Pilih simpul awal yang akan berfungsi sebagai node awal, kemudian tambahkan secara berurutan bobot jarak dari simpul awal ke simpul terdekat. Algoritma Dijkstra akan melakukan ekspansi pencarian dari satu simpul ke simpul lainnya, serta ke simpul berikutnya, langkah demi langkah

2. Atributkan bobot (jarak) untuk setiap pasangan simpul, kemudian tetapkan nilai 0 pada simpul awal dan nilai tak terhingga pada simpul lainnya yang belum mendapatkan nilai (belum terisi).
3. Set semua node yang belum dilalui dan set node awal sebagai "Node keberangkatan".
4. Dari node keberangkatan, pertimbangkan node tetangga yang belum dilalui dan hitung jaraknya dari titik keberangkatan. Jika jarak ini lebih kecil dari jarak sebelumnya (yang telah terekam sebelumnya) hapus data lama, simpan ulang data jarak dengan jarak yang baru.
5. Saat kita selesai mempertimbangkan setiap jarak terhadap node tetangga, tandai node yang telah dilalui sebagai "Node dilewati". Node yang dilewati tidak akan pernah di cek kembali, jarak yang disimpan adalah jarak terakhir dan yang paling minimal bobotnya.
6. Set "Node belum dilewati" dengan jarak terkecil (dari node keberangkatan) sebagai "Node Keberangkatan" selanjutnya dan ulangi langkah 5.

Untuk mempermudah perhitungan maka dibuat tabel seperti pada tabel 4.1.

Tabel 1. Perhitungan Dijkstra

Vertex	Pangkalan 1	Rumah 1	Rumah 2	TPS 1	TPS 2	TPS 3	TPS 6
Pangkalan 1	0	∞	∞	∞	∞	∞	∞
Rumah 1	0	0,38	0,64	1,04	0,81	0,68	∞
Rumah 2	0	0,38	0,64	1,04	0,81	0,68	∞
TPS 3	0	0,38	0,64	1,04	0,81	0,68	∞
TPS 2	0	0,38	0,64	1,04	0,81	0,68	0,86
TPS 6	0	0,38	0,64	1,04	0,81	0,68	0,86
TPS 1	0	0,38	0,64	1,04	0,81	0,68	0,86

Pada tabel diatas nilai yang pertama diketahui adalah nilai vertex Pangkalan 1 ,karena merupakan titik awal maka diberikan nilai 0 dan nilai Infinity pada vertex lainnya.langkah selanjutnya adalah mencari nilai terkecil maka nilai yang didapatkan adalah 0,lalu jadikan vertex Pangkalan 1 sebagai current vertex ,langkah selanjutnya adalah menghitung bobot dari vertex yang terhubung langsung dengan vertex Pangkalan 1,yaitu vertex Rumah 1,vertex Rumah 2,vertex TPS 1,vertex TPS 2 dan vertex TPS 3.Setelah nilainya didapatkan cari nilai bobot terkecilnya dan jadikan sebagai current vertex maka vertex yang dipilih adalah vertex Rumah 1,selanjutnya cari vertex yang terhubung langsung pada vertex Rumah 1,karena tidak ada vertex yang terhubung ke Rumah 1 ,maka langkah selanjutnya adalah mencari nilai terkecil dan didapatkan vertex Rumah 2, setelah vertex Rumah 2 dikunjungi dan tidak ada vertex yang terhubung ke vertex Rumah 2 ,maka dicari lagi nilai minimum dan didapatkan vertex TPS 3,langkah selanjutnya adalah mencari vertex yang terhubung ke TPS 3,TPS 3 terhubung dengan TPS 6,maka update nilai TPS 6 dari Infinity ke bobot yang lebih kecil yaitu 0,86.Selanjutnya didapatkan nilai minimum yang belum dikunjungi yaitu TPS 2,lalu hitung bobot vertex yang terhubung ke TPS 2 yaitu TPS 6, namun karena nilainya lebih besar dari nilai sebelumnya maka nilai dari TPS 6 tidak diupdate.Selanjutnya adalah mencari nilai minimum pada vertex yang belum dikunjungi yaitu TPS 6,namun karena TPS 6 tidak memiliki neighbor maka perhitungan selanjutnya adalah mencari nilai minimum pada vertex yang belum dikunjungi yaitu TPS 1,TPS 6 terhubung ke TPS 6 namun karena bobotnya lebih besar dari yang sudah ada,maka nilainya tidak diupdate.

Karena semua vertex sudah dikunjungi maka perhitungan selesai dan didapatkan path terpendek dari Pangkalan 1 ke TPS 6 adalah Pangkalan 1 - TPS 3 - TPS 6 dengan total bobot 0,86.

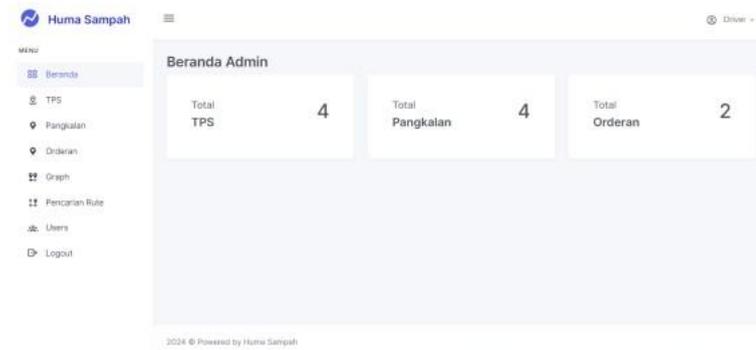
4.3. Implementasi Sistem

Implementasi aplikasi dilakukan sesuai dengan perancangan yang telah dilakukan pada tahap perancangan dan desain Gambar dibawah ini merupakan gambar dari halaman pertama saat website diakses.

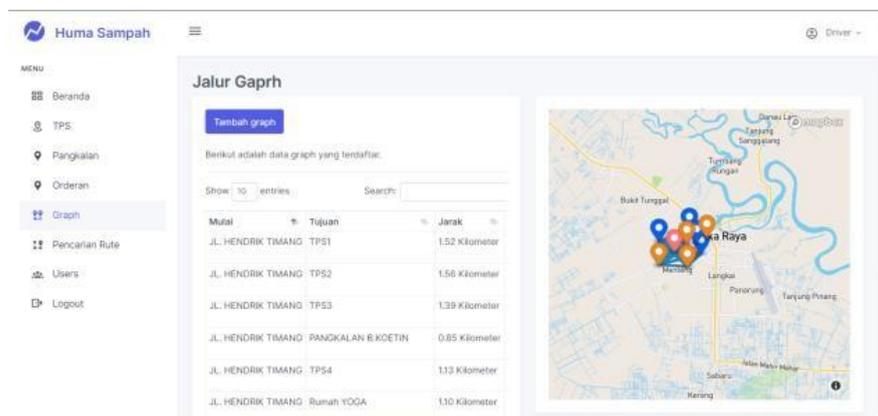


Gambar 5. Tampilan Halaman Utama Website

Selanjutnya pada halaman Driver terdapat beberapa fitur seperti yang terlihat pada gambar menu dibawah ini.

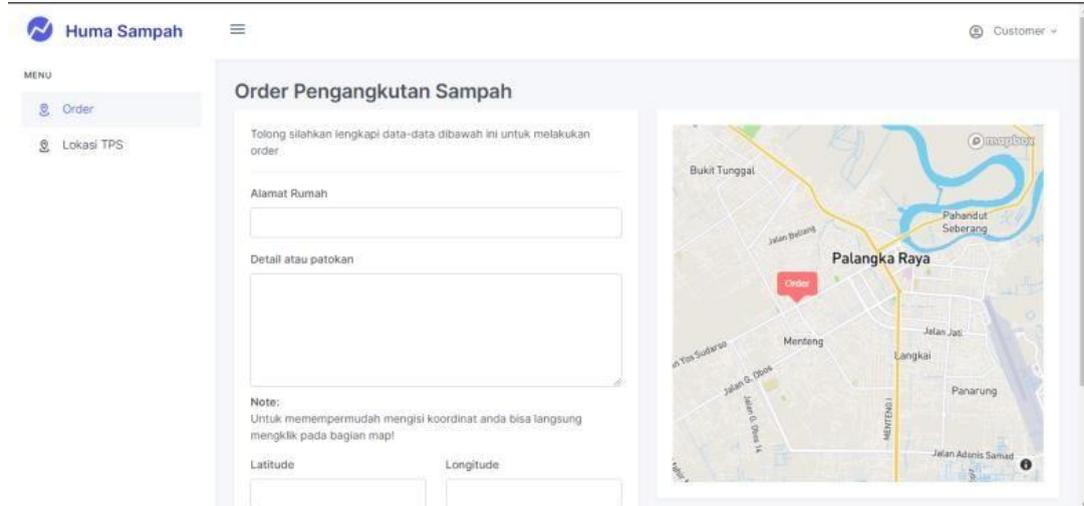


Gambar 6. Tampilan Menu pada Halaman Admin



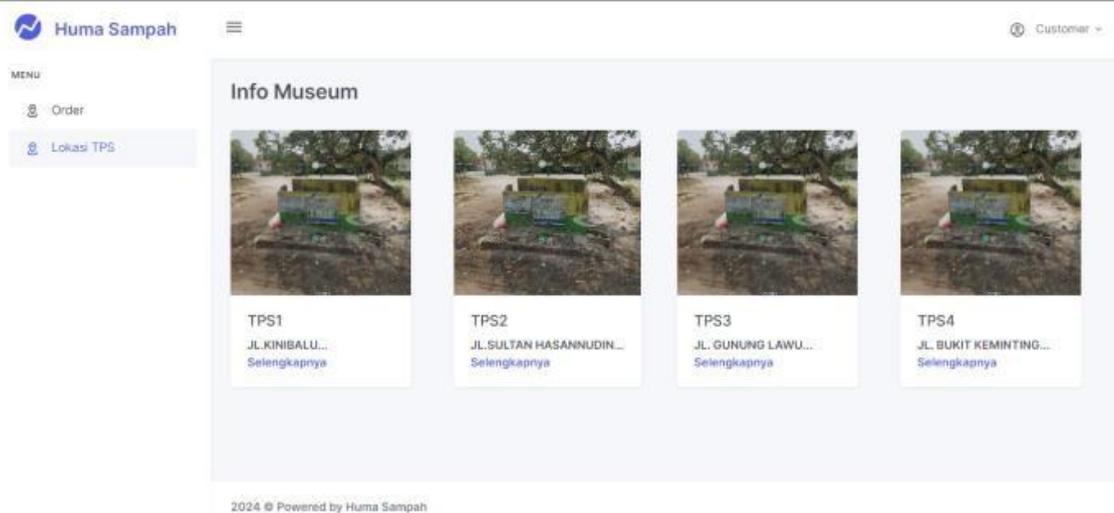
Gambar 7. Tampilan Menu pada Halaman Admin

Selanjutnya pada halaman *Customer* terdapat beberapa fitur seperti yang terlihat pada gambar menu dibawah ini.



Gambar 8. Tampilan Menu pada Halaman *Custom*

Selanjutnya pada halaman *Mahasiswa* terdapat beberapa fitur seperti yang terlihat pada gambar menu dibawah ini.



Gambar 9. Tampilan Halaman *Customer*

4.4. Pengujian

Setelah melakukan implementasi, dilakukan pengujian terhadap sistem dengan tujuan untuk melihat semua kesalahan dan kekurangan yang ada pada sistem. Pengujian sistem pada website ini dilakukan dengan menggunakan metode *blackbox*. Hasil pengujian *blackbox* menunjukkan bahwa setiap fitur yang ada pada website dapat digunakan dengan baik.

5. KESIMPULAN

Kesimpulan yang didapatkan dari penelitian “Pencarian Rute Terpendek Kendaraan Pengangkut Sampah Dengan Menggunakan Algoritma Dijkstra di Kota Palangka Raya” adalah

perancangan website ini menggunakan metodologi waterfall yang memiliki beberapa tahapan yaitu requirement and definition yang digambarkan menggunakan flowchart. System and software design yang digambarkan menggunakan unified modelling language (UML) yang terdiri dari use case diagram, activity diagram dan class diagram sehingga penggambaran struktur sistem dapat terlihat dengan jelas. Tahapan pengkodean dengan menggunakan beberapa bahasa pemrograman yaitu, PHP, Javascript, dan MySQL. Pengujian yang digunakan pada pembuatan website ini adalah metode blackbox testing. Untuk implementasikan pencarian rute terdekat dengan algoritma Dijkstra yang terhubung dengan peta, diperlukan map API dapat menggambarkan jalur terpendek yang akan dihasilkan algoritma namun tidak memperhitungkan kemacetan atau perubahan elevasi ketinggian sehingga rute yang diberikan bisa jadi rute terpendek.

Dari hasil rancang bangun dan hasil pengujian black box testing, ini dapat dihasilkan aplikasi berbasis website yang dapat memfasilitasi pengguna dalam mencari jasa jalur terpendek dengan algoritma Dijkstra.

DAFTAR PUSTAKA

- [1] Andayani, S., & Perwitasari, E. W. (2014). Penentuan Rute Terpendek Pengambilan Sampah di Kota Merauke Menggunakan Algoritma Dijkstra.
- [2] Maison, Pasaribu, F. T., Syarkowi, A., Evita, Novferma, Asiani, R. W., . . . Rahayu, M. A. (2017). Peran Sains, Teknologi dan Pendidikan MIPA dalam Menopang Sains Park, Teknopark, Serta Geopark Berbasis Argoindustri dan Lingkungan. Jambi: Fakultas Keguruan dan Ilmu Pendidikan (FKIP).
- [3] Hapsari, L. I. (2017). Retrieved from repository.unej.ac.id: <https://repository.unej.ac.id/bitstream/handle/123456789/82715/Lucky%20Indrayu%20Hapsari.pdf?sequence=1&isAllowed=y>
- [4] Galih Nalendro (2019). *Sistem Informasi Pengaduan Layanan Universitas*. <https://dspace.uin.ac.id/bitstream/handle/123456789/13526/Laporan%20TA-14523047.pdf?sequence=1&isAllowed=y>
- [5] Nuryagandhi, R. F. (2016). IMPLEMENTASI ALGORITMA DIJKSTRA UNTUK MENENTUKAN RUTE TERPENDEK LOKASI OBJEK WISATA DI KABUPATEN PATI BERBASIS WEB. Retrieved from lib.unnes: <https://lib.unnes.ac.id/27930/1/5302411025.pdf>
- [6] Cantona, A., Fauziah, & Winarsih. (2020). Implementasi Algoritma Dijkstra Pada Pencarian Rute Terpendek ke Museum di Jakarta. *Jurnal Teknologi dan Manajemen Informatika*, 27-34.