JURNAL TRIDI
Teknologi Informatika & Komputer
Vol. 2 No. 2 Juli 2025
Universitas Bani Saleh FTID
P-ISSN: - , E-ISSN: -

IMPLEMENTASI ALGORITMA DIJKSTRA UNTUK MENEMUKAN RUTE TERDEKAT PENGAMBILAN BARANG BEKAS BERBASIS WEB PADA LAPAK RONGSOK BROSOT, BEKASI

HS Sulistyowati[1] , Muhammad Nur [2] , Dwi Fajri[3]

[1]Universitas Bani Saleh, sulis@ubs.ac.id
[2]Universitas Bani Saleh, mnur@ubs.ac.id
[3]Universitas Bani Saleh, dwifajri000@gmail.com

ABSTRAK

Pengelolaan sampah dan limbah yang memiliki nilai sebagai bahan baku, khususnya barang bekas, menjadi tantangan dalam hal pengambilan dan rute yang efisien bagi agent pengumpul barang rongsokan. Penelitian ini mengimplementasikan *Algoritma Dijkstra* untuk menentukan rute terdekat bagi agen rongsokan dalam proses pengambilan barang bekas di Lapak Rongsok Brosot. Aplikasi dirancang agar berjalan pada smartphone dengan platform android yang digunakan mayoritas pada saat ini. Berdasarkan pengujian didapatkan pemilihan rute terpendek menggunakan algoritma dijkstra dinilai sangat efektif berdasarkan pencarian rute terpendek dari setiap perhitungan bobot jarak dengan node yang dijalurkan kearah tujuan. *Platform* berbasis web yang dikembangkan memanfaatkan algoritma ini mampu mengoptimalkan rute agen, sehingga dapat mengurangi waktu dan tenaga yang dibutuhkan serta meningkatkan efisiensi operasional. Sistem yang dibangun juga memudahkan masyarakat untuk menghubungi agen terdekat, mempercepat proses transaksi, dan meningkatkan jumlah barang yang berhasil dikumpulkan. Selain itu, penelitian ini memberikan peluang bagi pengembangan lebih lanjut melalui penerapan algoritma alternatif serta peningkatan fitur seperti pemantauan kinerja agent dan keamanan sistem.
Kata Kunci : *Algoritma Dijkstra*, rute terpendek, *platform web*, agen rongsokan, efisiensi operasional.

ABSTRAC

Waste management, which has value as raw materials, particularly used goods, presents a challenge in terms of efficient collection and routing for scrap collection agents. This study implements Dijkstra's Algorithm to determine the shortest route for scrap agents during the scrap collection process at Lapak Rongsok Brosot. The application is designed to run on smartphones with the Android platform, which is currently the majority in use. Testing revealed that selecting the shortest route using the Dijkstra algorithm is considered highly effective, based on finding the shortest route based on each distance weighted calculation with the nodes that are routed to the destination. The developed web-based platform utilizes this algorithm to optimize agent routes, thereby reducing time and effort and improving operational efficiency. The system also makes it easier for the public to contact the nearest agent, speeding up the transaction process, and increasing the number of items collected. Furthermore, this study provides opportunities for further development through the implementation of alternative algorithms and feature enhancements such as agent performance monitoring and system security.
Keywords: Dijkstra's Algorithm, shortest route, web platform, scrap agents, operational efficiency.

PENDAHULUAN

Sampah dan limbah mempunyai suatu pengertian yang berbeda, menurut undang undang RI No. 18 tahun 2008 dan Peraturan Pemerintah No. 18 Tahun 1999. Pengertian sampah adalah sisa kegiatan sehari – hari manusia atau proses alam yang berbentuk padat dan hampir semua sampah bisa di daur ulang baik untuk pupuk atau lainnya. Pengertian limbah adalah sisa suatu usaha atau kegiatan yang lebih rumit dari sampah. Sampah dan limbah memiliki nilai untuk dijadikan kembali sebagai bahan baku dari suatu barang lain. Hal ini merupakan peluang bagi pelaku usaha untuk mengumpulkan sampah dan limbah yang sering disebut lapak rongsok. Agen rongsokan berkeliling mengumpulkan sampah dan limbah dari masyarakat yang memiliki nilai untuk di jual.

Lapak Rongsok Brosot adalah salah satu pelaku usaha yang mengumpulkan sampah dan limbah dari masyarakat yang memiliki nilai sebagai bahan baku. Sampah dan limbah yang memiliki nilai digunakan sebagai barang baku disebut barang bekas. Lapak tersebut memiliki 3 agen rongsokan yang berkeliling setiap hari untuk mengambil sampah dan limbah yang telah dikumpulkan oleh masyarakat untuk dijual. Masyarakat menghubungi lapak rongsok untuk menjual barang bekas. Dalam hal ini setiap hari agen harus menghampiri kesetiap daftar lokasi masyarakat yang menjual barang bekas. Permasalahan yang dihadapi oleh Lapak adalah bagaimana urutan pengambilan barang ke agen-agen , sehingga diperoleh waktu yang efisien dan biaya transportasi yang lebih murah. Masalah tersebut perlu dipecahkan mengingat jumlah barang yang diambil volumenya cukup besar dan jarak agen yang cukup berjauhan antara agen yang satu dengan yang lain. Terdapat beberapa metode yang digunakan untuk memecahkan masalah pengambilan atau pengurutan rute agar diperoleh nilai yang minimum. Beberapa metode yang digunakan *Algoritma Dijkstra, Algortma Floyd Warshal, Algortime Semut (ant colony)* [1], [2]. *Algortima Dijkstra* adalah sebuah algoritme rakus (*greedy algorithm*) yang dipakai dalam memecahkan permasalahn jarak terpendek (*shortest path problem*) untuk sebuah graf berarah (*directed grap*) dengan bobot – bobot garis (*egde weights*) dengan yang bernilai *nonnegative*. Algoritme yang digunakan pada permasalah menentukan rute terpendek pengambilan barang bekas pada kasus ini adalah *Algortima Dijkstra*. *Algoritma Dijkstra* menghitung setiap jarak antara titik satu ke titik yang lain dan menemukan jarak terpendek. Adapun rata-rata waktu eksekusi *Algoritma Dijkstra* bila dibandingkan dengan *Algoritme Ant Colony* masih terukur lebih kecil, maka *Algoritma Dijkstra* banyak dimanfaatkan dalam proses mencari jalur optimum di dalam jaringan internet dibanding algoritma yang lainnya [3]. Sistem dapat mencari rute dengan baik menggunakan *Algoritma Dijkstra,* namun sebelum mencari rute tersebut dibutuhkan *algoritma geometri haversine* formula yang berfungsi untuk menghitung antar *latitude dan longitude* satu dengan lainnya [4].

Tujuan dari penelitian ini adalah untuk menentukan rute terpendek pengambilan barang oleh agen dari tempat-tempat penjualan/seller barang bekas pada Lapak Rongsok Brosot di wilayah Bekasi. Penentuan rute terpendek menggunakan Algoritma Dijkstraa, dengan implementasinya berupa pemrograman berbasis web. Tiap- tiap seller merupakan titik- titik yang akan digunakan untuk menemukan rute terdekat pada daftar lokasi yang di berikan oleh lapak rongsok kepada agen rongsokan. Penggunaan *Algoritma Dijkstra* juga menggunakan 1 titik sumber yang sama dengan permasalahan

tersebut. Sarana atau platform yang dibuat berupa program berbasis *web* dengan menggunakan Algoritma Dijkstra untuk menentukan rute yang dilalui oleh agen rongsokan. Sarana atau platform berbesis *web* ini berguna untuk meningkatkan effisiensi biaya transportasi dan meningkatkan pendapatan dari Lapak Rongsok serta memudahkan penjual menemukan dan menghubungi lapak rongsok. Sarana atau platform yang dibuat juga meningkatkan jumlah barang yang yang di kumpulkan, karena akan mudah bagi penjual/masyarakat untuk menemukan rute yang akan dilewati oleh setiap agen/kurir. Demikian juga bagi kurir penentuan rute terpendek dapat mempersingkat waktu pengambilan barang.

TINJAUAN PUSTAKA

Berikut beberapa referensi yang berkaitan dengan obyek pembahasan pada penelitian ini, diantaranya sebagai berikut:

1. *Implemantasi algortima Djkstra Untuk Menentukan Rute Terpendek Meneju Pelayanan Kesehatan* (Ika Arthalia Yulandari dan Pristi Sukmasetyan, 2022 .) Dalam penelitian ini Algoritma Djikstra dapat menentukan jalur terpendek (shortes path) yaitu berdasarkan bobot terkecil dari suatu titik ke titik lainnya ,langkah – langkah yang dilakukan dalam algortima djiksta dari penentuan titik awal , kemudian pembobotan jarak dari node pertama ke node terdekat satu per satu sehingga diperoleh jarak minimum.
2. *Pencarian Rute Terpendek Pada Aplikasi Ojek Sampah Dengan Menggunakan Algoritama Djkstra,* (Luluk Suryani dan Ery Murniasih, 2022). Kesimpulan dari jurnal ini adalah Aplikasi android dengan Algoritma Djikstra dapat di terapkan dan berhasil berjalan sebagai metode penentuan jarak terpendek untuk proses pengangkutan sampah pada rumah warga berdasarkan inputan alamat awal dari supir menuju alamat tujuan penjemputan .
3. *Implementasi Algoritma Djkstra Untuk Menenetukan Rute Terpendek Wilayah Pasar Minggu Dan STMIK NusaMandiri Jakarta* (Supriadi Panggabean,Windu Gata, et all, 2022) .Kesimpulan dari jurnal ini adalah berdasarkan dari pembahasan diatas maka dapat disimpulkan bahwa untuk mencari jalur rute terpendek antara wilayah Pasar Minggu dengan STMIK Nusa Mandiri Kramat Jakarta sudah ditemukan rutenya, yaitu rute ketiga dengan jarak tempuh 14,8 km. Namun hasil ini tidak menjamin pengguna jalan cepat sampai ketempat tujuan melainkan peluangnya lebih besar. Maka diharapkan pengguna jalan dapat dengan bijak memilih jalur atau rute yang ditempuh untuk mencapai tempat tujuan yang diinginkan.
4. *Implementasi Algoritma Djkstra Pada Pencarian Rute Terpendek Ke Museum di Jakarta* (Aldy cantona,Fauziah,Winarsih, 2020) .Kesimulan dari bahwa algoritma yang digunakan merupakan solusi dalam mencari rute terpendek, dengan metode algoritma Dijkstra yang diterapkan dalam smartphone pengguna dapat mempersingkat efektifitas waktu untuk mencari museum di Jakarta. Rute yang dihasilkan pada algoritma ini sangat efektif bila dilalui dengan mengendarakan mobil dengan mengesampinkan kemacetan dan kondisi ganjil genap di Jakarta. Berdasarkan pengujian yang dilakukan, penerapan algoritma Dijkstra untuk mendapatkan rute terpendek dinilai efektif sebab dengan total 20 bobot hanya

memerlukan 7 bobot, yang mana hal tersebut menyatakan hanya perlu 35% bobot untuk mencapai Museum Nasional.

Algoritma Dijkstra, adalah sebuah algoritma rakus *(greedy algorithm)* yang dipakai dalam memecahkan permasalahan jarak terpendek *(shortest path problem*) untuk sebuah graf berarah (*directed graph*) dengan bobot-bobot garis (*edge weights*) yang bernilai nonnegatif, [0 , ∞) .Input algoritma ini adalah sebuah graf berarah yang berbobot G dan sebuah titik asal s dalam himpunan garis V. Misalnya, bila titik dari sebuah graf melambangkan kota-kota dan bobot garis melambangkan jarak antara kota-kota tersebut, Algoritma Dijkstra dapat digunakan untuk menemukan jarak terpendek antara dua kota pada graf tersebut . Biaya (*cost*) dari sebuah garis dapat dianggap sebagai jarak antara dua simpul, yaitu jumlah jarak semua garis dalam jalur tersebut. Untuk sepasang titik s dan t dalam garis V, algoritma ini menghitung jarak terpendek dari titik s ke titik t.

Pengertian lain dari Algoritma Djikstra adalah algoritma yang digunakan untuk mencari jalur terpendek dari satu titik ke semua titik lain dalam sebuah graf berbobot. Bobot ini dapat berupa jarak, biaya atau parameter lain yang mempengaruhi perhitungan jalur terpendek. Algoritma Dijkstra menggunakan pendekatan *greedy*, yang berarti pada setiap langkahnya, algoritma ini akan memilih simpul yang memiliki jarak terpendek dari titik awal. Dalam prosesnya, algoritma ini secara bertahap memperluas jalur yang diketahui untuk mencapai simpul-simpul lain dalam graf, hingga mencapai titik tujuan atau seluruh simpul telah dikunjungi.[7]

METODOLOGI PENELITIAN

Penelitian ini mengimplementasikan *Algoritma Dijkstra* untuk menentukan rute terdekat bagi agen rongsokan dalam proses pengambilan barang bekas di Lapak Rongsok Brosot . Kemudian dibuat aplikasi berbasis web untuk menentukan rute terpendek, berdasarkan analisis data menggunakan teknik UML (*Unified Modeling Language)* , sedangkan pencarian rute terpendek menggunakan Algoritma Djikstra. Pada dasarnya Algoritma Djikstra termasuk dalam algoritma untuk masalah pencarian graf yang mampu menuntaskan masalah mencari lintasan terpendek dengan satu sumber pada sebuah graf yang tidak memiliki bobot sisi negatif, dan menghasilkan sebuah alur lintasan terpendek. Algoritma Dijkstra ini sering digunakan untung melakukan routing (Aprilianingsih, Primananda, & Suharsono, 2017).[1] Algoritma Dijkstra ini sendiri ditujukan untuk menemukan jalur terpendek berdasarkan bobot terkecil dari titik keberangkatan ke titik lainnya. Misalkan sebuah gedung dan tempat tertentu dijadikan titik dan jalanan dijadikan garis, maka Algoritma Dijkstra akan melakukan perhitungan terhadap semua garis dengan bobot terkecil dari setiap titik (Martin Nugroho Parapat, Deddy Kusbianto, 2017).[6]

Langkah – langkah dalam pengerjaan Algoritma Djikstra adalah sebagai berikut 1. Menentukan titik awal yang ingin dicari jalur terpendeknya.

2. Inisialisasi jarak awal untuk semua simpul dalam graf dengan nilai tak hingga, kecuali titik awal yang diatur jaraknya menjadi 0.
3. Tandai titik awal sebagai simpul saat ini.

4. Untuk setiap tetangga simpul saat ini, hitung jarak baru yang lebih pendek melalui simpul saat ini. Jika jarak baru lebih pendek dari jarak sebelumnya, perbarui jarak tersebut.
5. Menandai simpul saat ini sebagai simpul yang dikunjungi.
6. Jika titik tujuan telah dikunjungi atau tidak ada simpul yang dapat dikunjungi lagi, selesaikan algoritma. Jika tidak, pilih simpul dengan jarak terpendek yang belum dikunjungi sebagai simpul saat ini dan lanjutkan ke langkah 4.

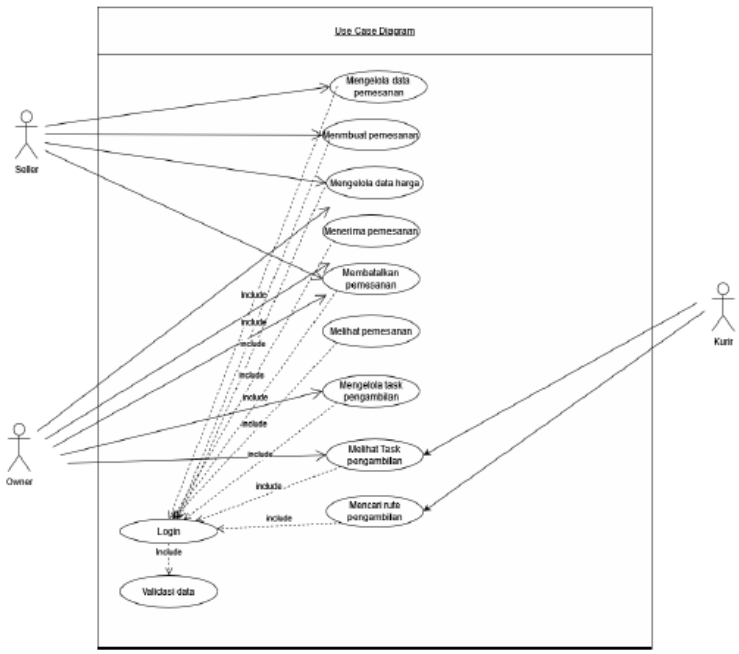
Perancangan Sistem

Pemodelan proses dan data pada analisa ini menggunakan teknik UML (*Unified Modeling Language*). Tahap-tahap pemodelan analisa tersebut terbagi menjadi 4 yaitu, identifikasi actor, *Use Case* diagram,*Activity* diagram, sequence diagram, dan class diagram.

1. *Use Case* Diagram. *Use Case* diagram menggambarkan siapa saja aktor yang melakukan prosedur dalam sistem serta fungsi – fungsi (proses) yang terlibat dalam transformasi pada sistem tersebut .Adapun *Use Case* diagram yang berjalan pada skripsi implementasi Algoritma Djikstra untuk menemukan rute terdekat pengambilan barang bekas oleh agen /kurir pada Lapak Rongsok Brosot ini sebagai berikut:

a. *Seller. Seller* adalah aktor yang berperan memesan pengambilan barang kelokasi yang dia inginkan dengan menginput harga,lokasi,jenis barang.
b. *Owner. Owner* adalah aktor yang berperan untuk mengambil pesanan yang sudah dibuat oleh *Seller* lalu memberikan task pengambilan kepada kurir/agen
c. Kurir/agen . Melakukan pengambilan barang sesuai task yang sudah di berikan oleh *Owner* dan menggunakan Algoritma Djikstra untuk menemukan rute terdekat untuk pengambilan task tersebut.

Usecase diagram dari hubungan aktor-aktor diatas dapat dilihat pada gambar 2.1

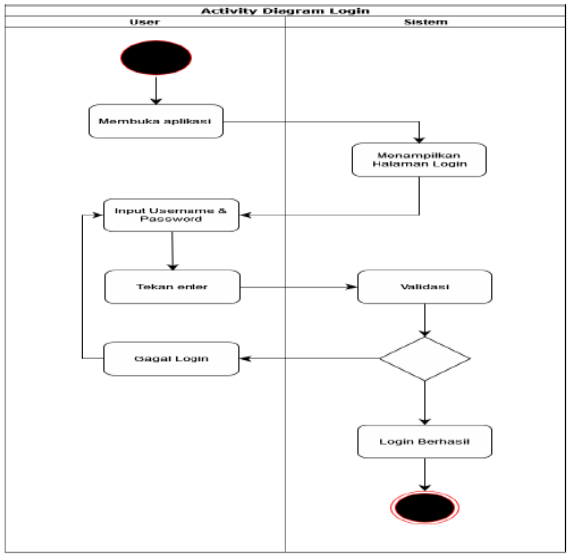


Gambar 2.1. Usecase Diagram

2. *Activity Diagram*.

Activity diagram menguraikan interaksi yang terjadi antara user dengan sistem pada masing – masing *Use Case*. Ada beberapa aktivitas user terhadap sistem sebagai berikut:

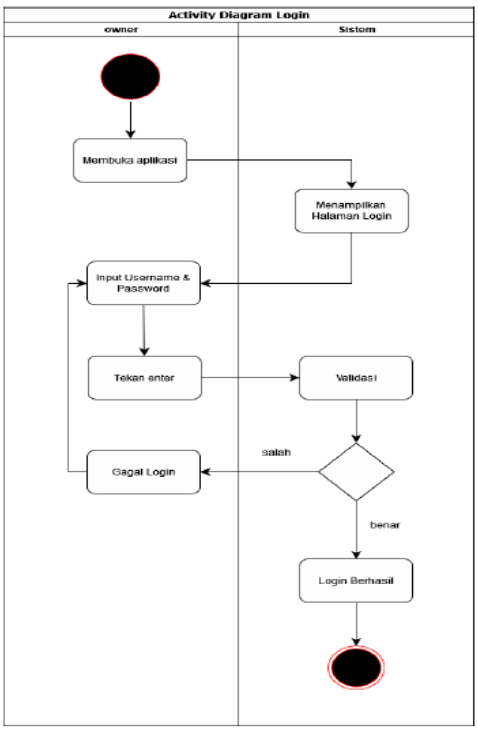
a. Activity diagram Login Seller

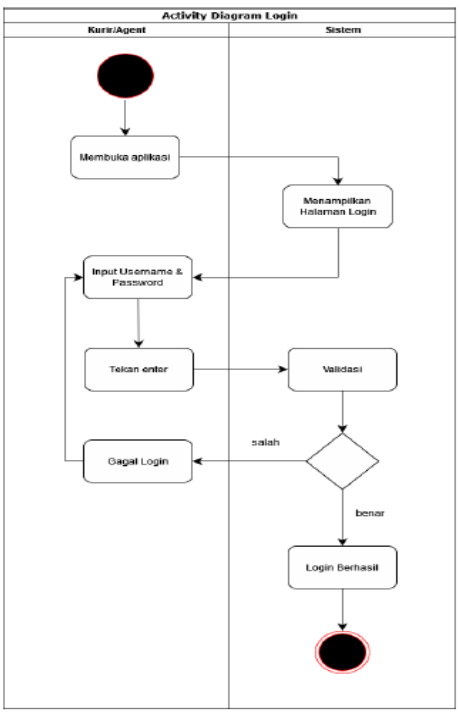


Gambar 2.2. *Activity Diagram Seller*

b. Activity diagram Login Owner dan *Activity diagram Login Kurir/Agen*

Berikut adalah gambar *Activity diagram Login Owner* dan*Activity diagram Login Kurir/Agen*

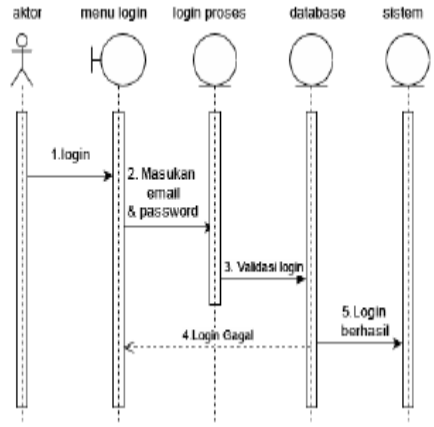


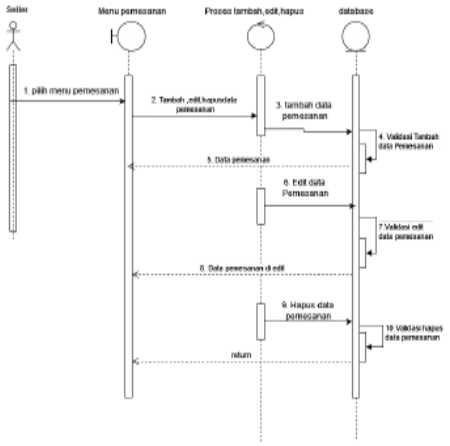


Gambar 2.3. *Activity Diagram Owner* Gambar 2.4. *Activity Diagram Kurir*

3. Sequence Diagram

a. Sequence Diagram Login dan *Sequence Diagram pada Seller*

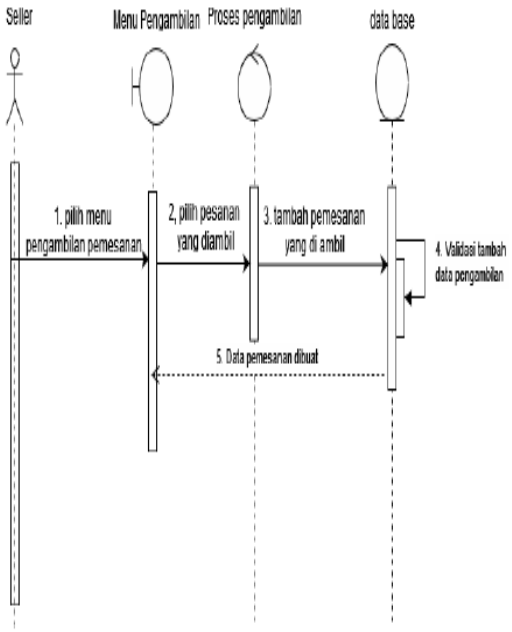


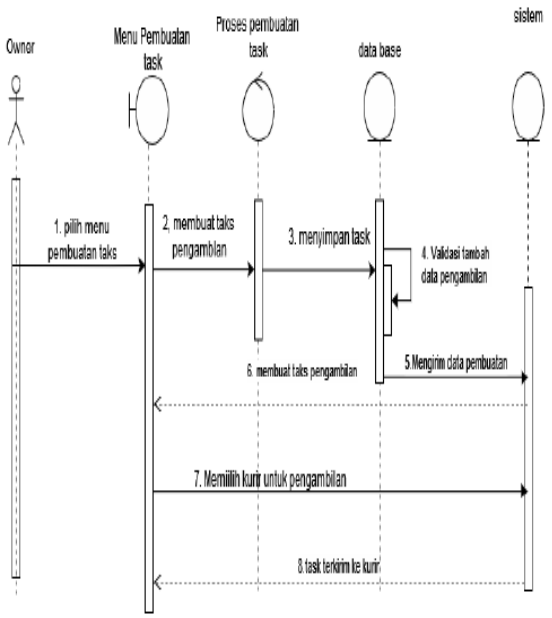


Gambar 2.5. *Sequence Diagram Login* Gambar 2.5. *Sequence Diagram Seller*

b. Squence Diagram owner menerima pesanan dan *Squence Diagram owner mengirim task* ke kurir.

Squence diagram owner untuk menerima pesanan dari penjual /seller dan sequence diagram owner mengirim task ke kurir, dapat dilihat pada gambar 2.6 dan 2.7.

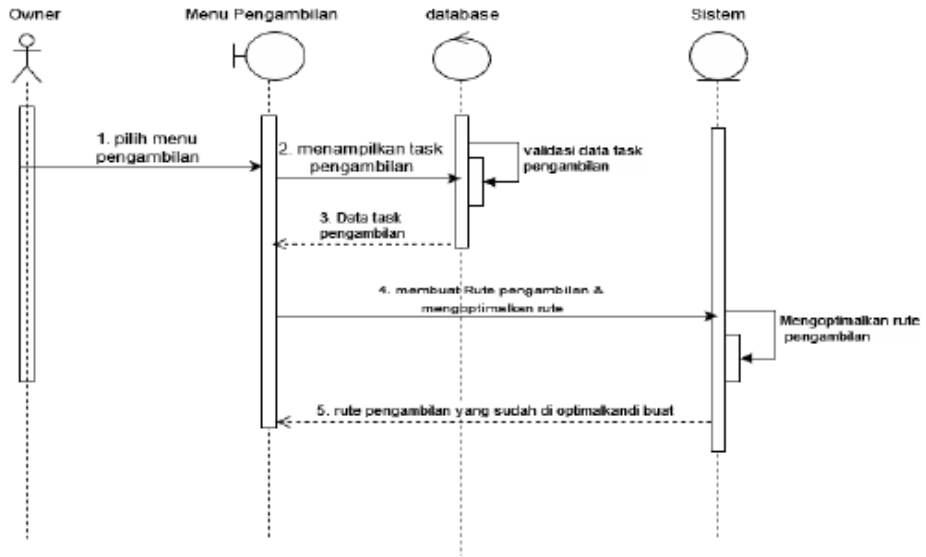




Gambar 2.6. *Squence Diagram owner* menerima pesanan

Gambar 2.7. *Squence Diagram owner* mengirim task ke kurir

c. Squence Diagram Kurir/agen



Gambar 2.7. *Squence Diagram Kurir/Agen*

Teknik Analisis

1. Persiapan Data

Persiapan data dilakukan untuk menentukan data yang akan digunakan. Kriteria yang dipakai untuk persiapan data ini adalah jarak lokasi yang sudah di ambil dari google maps. Tabel jarak dari setiap node lokasi yang sudah di tentukan dibuat berdasarkan jarak yang di dapat dari google maps dan mendapatkan data sebagai berikut berdasarkan kilometer (km).

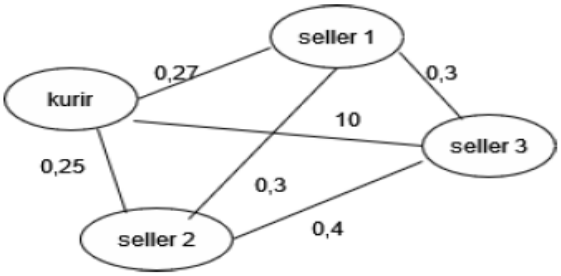
Tabel 2.1. Tabel Jarak setiap node lokasi

	Kurir	Seller 1	Seller 2	Seller 3
Kurir	0	0.27	0.25	0.35
Seller 1	0.27	0	0.35	0.27
Seller 2	0.25	0.35	0	0.4
Seller 3	0.35	0.27	0.4	0

Keterangan :

Kurir berperan sebagai agen dalam pengambilan barang bekas

Seller berperan sebagai penjual barang bekas



Gambar 2.8. Graf Lokasi

2. Analisis Algorita Dijkstra

Langkah – langkah analisis penggunaan Algoritma Dijkstra adalah sebagai berikut

Langkah 1 : Inisiasi jarak awal semua simpul dalam graf dengan nilai tak hingga ,kecuali simpul awal (kurir) yang diatur jaraknya menjadi 0, dan mendapatkan jarak awal Kurir (0), Seller 1(∞), Seller 2(∞), Seller 3(∞).

Langkah 2 : Pilih simpul awal , Yaitu Kurir

Langkah 3 : Hitung Jarak baru melalui A ke tetangga - tetangganya yaitu seller 1 (0,27) dan seller 2 (0,25)

Langkah 4 : Tandai simpul kurir sebagai di kunjungi

Langkah 5 :Pilih simpul dengan jarak terpedek yang belum di kunjungi yaitu seller 2

Langkah 6 : Hitung jarak baru melalui seller 2 ketetanggaan yaitu seller 1 (0,35) dan seller 3(0,4)

Langkah 7 : Tandai seller 2 sebagai di kunjungi

Langkah 8 : Pilih simpul jarak terpendek yang belum di kunjungi yaitu seller 1

Langkah 9 : Tandai seller 2 sebagai dikunjungi

Langkah 10 : pilih simpul jarak terpendek yang belum di kunjungi seller 3

Langkah 11 : Tandai seller 3 sebagai dikunjungi

Langkah 12 : Selesaikan algoritma ,karena telah mencapai seller yaitu terakhir seller 3

Dalam bentuk tabel hasil Perhitungan dengan menggunakan Algoritma Dikstra dapat dilitah dalam tabel berikut:

Tabel 2.2. Langkah Perhitungan Algoritma Dijkstra

Iteration	*Unvisited* (Q)	*Visited* (s)	Current	Node:Min= (dist[node],prev[node])iteraton			
				V1	V2	V3	V4
	Initalization {v1,v2,v3, v4}	{-}		(0,-)0	(∞,-)0	(∞,-)0	(∞,-)0
1	{v2,v1,v3}	{V1}	V1		(0,25, v1)1	(0,27, v1)1	(∞,v1)1
2	{v3,v4	{v1,v2}	V2			(0,55, v2)2	(0,75, v2)2
3	{v4}	{v1,v2, v3}	V3				(0,85, v3)3

Dengan demikian jarak terpendek yang di dapatkan adalah 0,85 km dengan jalur kurir (V1) →seller 2(V2) →seller 1 (V3) →seller 3(V4)

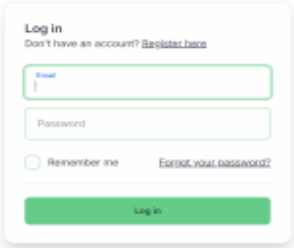
HASIL DAN PEMBAHASAN

Implementasi

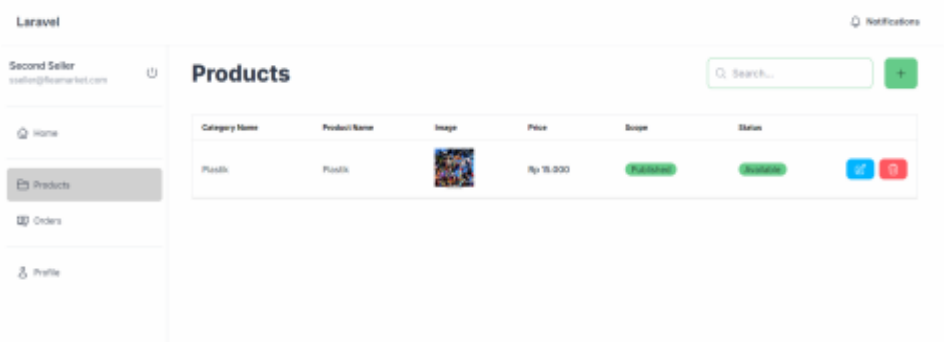
Bab ini menjelaskan implementasi dan pengujian dari aplikasi. Tahapan ini dilakukan setelah analisis dan desain serta diimplementasikan ke dalam bahasa pemrograman. Setelah implementasi selesai, maka dilakukannya pengujian aplikasi. Aplikasi yang di uji bertujuan untuk memastikan tidak adanya kesalahan pada aplikasi dan berjalan sesuai dengan tujuan. Tujuan dari implementasi ini adalah untuk menerapkan rancangan yang telah dilakukan terhadap aplikasi, sehingga pemakai dapat memberikan masukan untuk pengembangan dan perbaikan aplikasi dari yang dibuat..

1. Implementasi *User Interface*.
   a. Tampilan untuk Seller

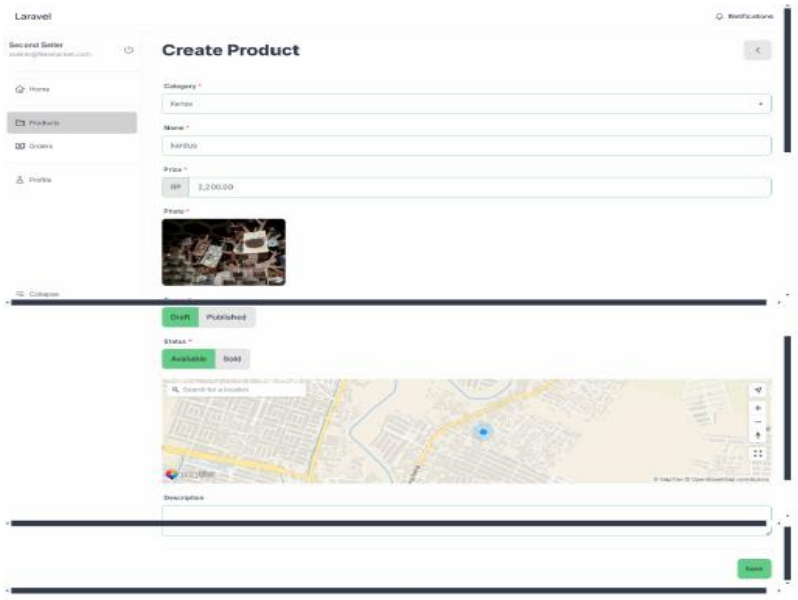
      Tampilan untuk seller terdiri dari: tampilan interface login seller, tampilan menu product, tampilan menu pemesanan pengambilan barang, tampilan pengambilan barang yang sudah diambil dan tampilan menu order .



Gambar 2.9. Tampilan menu login untuk seller

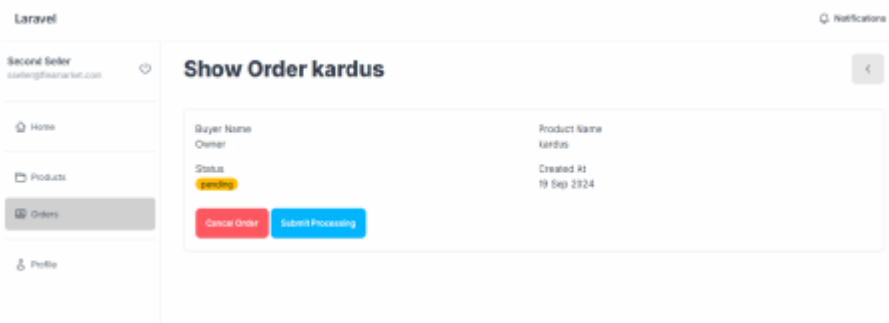


Gambar 2.10. Tampilan menu produk



Gambar 2.11.Menu pemesanan pengambilan barang

Gambar 2.12. Tampilan pengambilan barang yang sudah dibuat

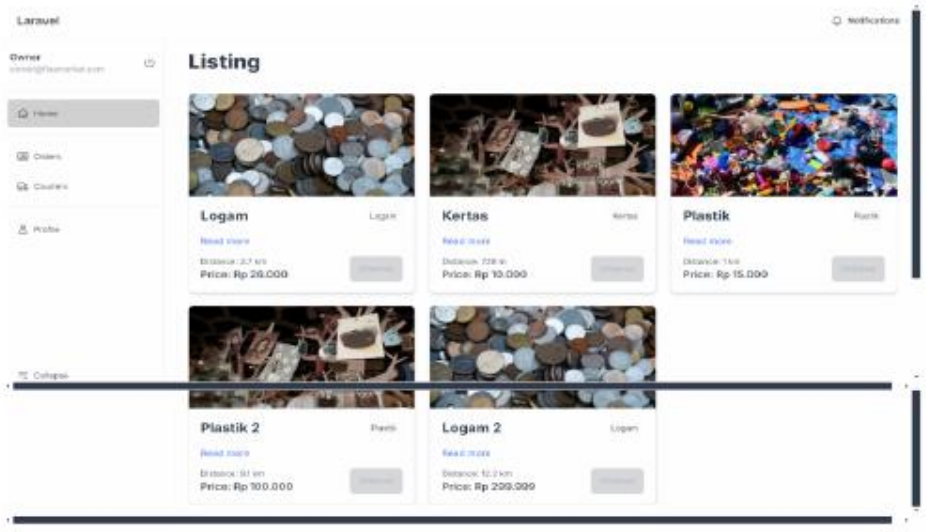


Gambar 2.13. Tampilan order oleh seller.

b. Tampilan untuk owner

Tampilan untuk owner terdiri dari:

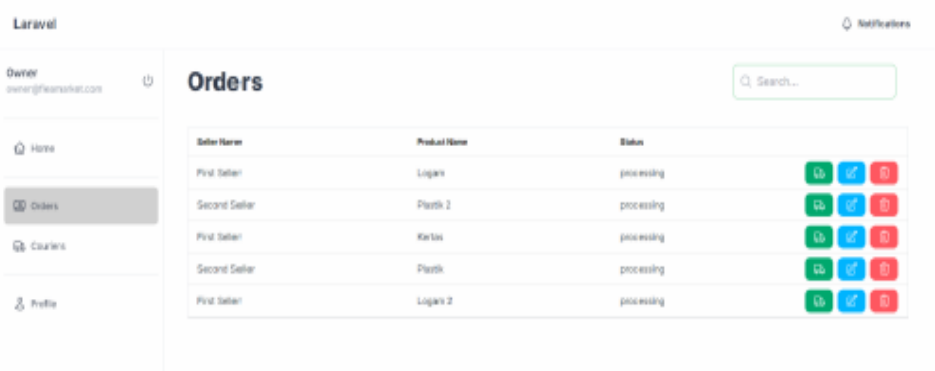
- Tampilan login untuk owner, yang dilanjutkan dengan tampilan home pada owner yang menampilkan pemesanan pengambilan barang yang sudah masuk di owner.



Gambar 2.14. Tampilan menu home pada owner

Pengambilan pemesanan produk yang sudah dibuat oleh seller. Pada tampilan ini menampilkan pemesanan pengambilan yang sudah diambil oleh owner, sehingga tampilan warna akan berubah.

- Tampilan order
  Pada menu order task pengambilan barang yang sudah diambilakan ditampilan dan dikirim ke kurir/agen. Pada tampilan ini owner memilih kurir untuk diberikan task pengambilan dan memerlukan assign oleh owner, sehingga tampilan daftar seller/penjual barang akan ditindak lanjuti oleh kurir sesuai dengan rute terdekat yang dapat dijangkau.



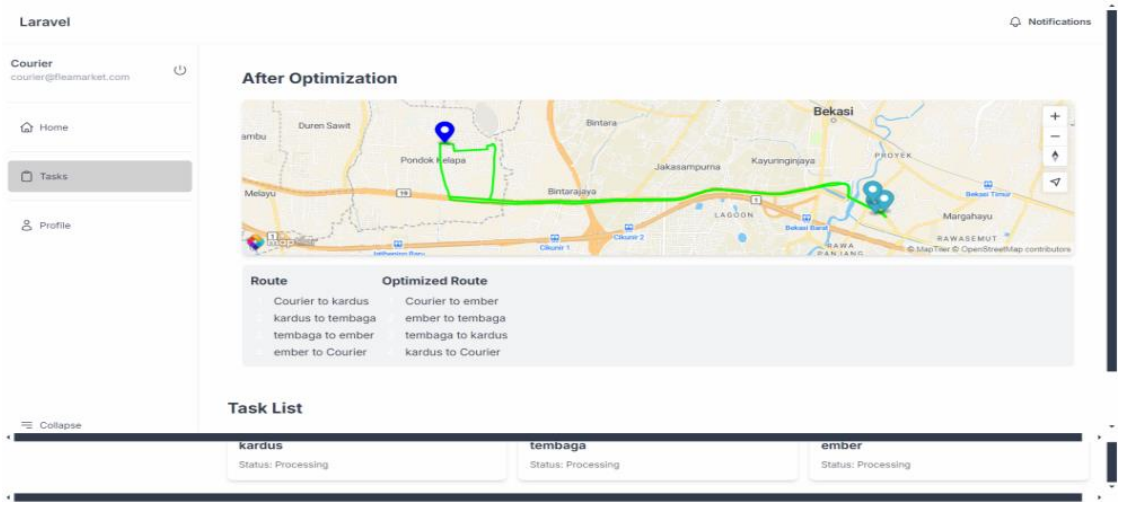
Gambar 2.15. Gambar tampilan order oleh owner

Gambar 2.16 Tampilan task yang sudah dikirim ke kurir

Gambar 2.17. Tampilan pembuatan akun untuk kurir

c. Tampilan untuk akun kurir

Pada menu order ini ditampilkan rute yang sudah dioptimasi untuk pengambilan barang:



Gambar 2.18 Tampilan rute pengambilan barang untuk kurir.

Tabel 2.3. Tabel perbandingan sebelum dan sesudah optimasi

Sebelum Optimasi	Sesudah Optimasi
Kurir ke kertas	Kurir ke kardus
Kertas ke logam	Kardus ke kertas
Logam ke kardus	Kertas ke logam
Kardus ke kurir /owner	Logam ke kurir/owner

KESIMPULAN

Hasil berikut dapat diambil kesimpulan dari analisis aplikasi yang sudah dibuat dengan implementasi algoritma djkstra ini adalah

1. Perhitungan Algoritma Djikstra mendapatkan hasil 0,85 dari kurir menuju seller 3 ,dimana seller 3 adalah titik terjauh maka didapatkan rute terdekat pengambilan barang bekas oleh agen adalah kurir →seller 2 → seller 1 → seller 3 sebagai titik pengambilan terakhir
2. Sistem yang dikembangkan mempermudah Agen dalam menemukan lokasi yang harus dikunjungi terlebih dahulu, sehingga dapat meningkatkan efisiensi operasional serta mengurangi waktu yang diperlukan untuk mengumpulkan barang bekas dari masyarakat.
3. Bagi pelaku usaha seperti Lapak Rongsok Brosot, penggunaan sistem ini mampu meningkatkan jumlah barang yang dikumpulkan dan memberikan dampak positif pada peningkatan pendapatan, mengingat proses pengambilan barang yang lebih cepat.
4. Aplikasi berbasis web ini juga memberikan kemudahan bagi masyarakat dalam menemukan dan menghubungi Agen rongsokan terdekat, sehingga memperlancar proses transaksi.

SARAN

Rekomendasi berikut harus di berikan berdasarkan analisis dan pengaturan aplikasi yang sudah di buat dengan mengimplementasikan algoritma djkstra :

1. Pengembangan Fitur Aplikasi.
   Aplikasi dapat dikembangkan lebih lanjut dengan menambahkan fitur fitur baru seperti integrasi dengan sistem pembayaran digital atau penjadwalan pengambilan otomatis, sehingga Agen rongsokan dapat bekerja lebih efisien dan terstruktur.
2. Penggunaan Algoritma Lain.
   Selain Algoritma Dijkstra, disarankan untuk mencoba algoritma lain seperti Algoritma Floyd-Warshall atau Algoritma Semut (Ant Colony) untuk melihat apakah ada peningkatan efisiensi dalam beberapa kasus tertentu, terutama pada kondisi jaringan yang lebih kompleks
3. Perluasan Ruang Lingkup Lokasi.
   Sistem ini dapat diperluas cakupannya dengan menambahkan lebih banyak data lokasi atau memungkinkan aplikasi digunakan di area yang lebih luas. Hal ini dapat mendukung keberlanjutan bisnis Lapak Rongsok Brosot di skala yang lebih besar.

4. Uji Pengguna dan Pengembangan Antarmuka.
   Melakukan uji coba kepada pengguna (user testing) secara lebih luas agar bisa mendapatkan umpan balik yang lebih bervariasi. Dengan demikian, tampilan antarmuka aplikasi (user interface) dan pengalaman pengguna (user experience) dapat terus disempurnakan.
5. Pengoptimalan Keamanan Sistem.
   Mengingat aplikasi berbasis web ini melibatkan transaksi dan data lokasi, disarankan untuk meningkatkan pengujian dan penerapan keamanan sistem, agar terhindar dari ancaman siber yang dapat merusak kredibilitas dan keamanan data pengguna
6. Penambahan Fitur Pemantauan Kinerja Agen.
   Aplikasi dapat dikembangkan untuk menambahkan fitur pemantauan kinerja Agen rongsokan, seperti statistik harian engenai jarak yang ditempuh atau waktu pengambilan barang bekas, untuk memberikan evaluasi kinerja yang lebih terukur.

DAFTAR PUSTAKA

[1] Aprilianingsih, E. P., Primananda, R., & Suharsono, A. (2017). Analisis Fail Path Pada Arsitektur Software Defined Network Menggunakan Dijkstra Algorithm. Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer (J-PTIIK) Universitas Brawijaya, 1(3), 174–183. Retrieved from http://j-ptiik.ub.ac.id/index.php/jptiik/article/view/59

[2] B. Folaiman, R. Rosihan, and A. Mubarak, “Implementasi Algoritma Dijkstra Untuk Penentuan Jalur Terpendek Pada Aplikasi Evakuasi Bencana Untuk Penyandang Disabilitas,” JIKO (Jurnal Inform. dan Komputer), vol. 1, no. 2, pp. 61–69, 2018, doi: 10.33387/jiko.v1i2.770.

[3] Cantona, A., Fauziah, F., & Winarsih, W. (2020). Implementasi algoritma dijkstra pada pencarian rute terpendek ke museum di Jakarta. *Jurnal Teknologi dan Manajemen Informatika*, *6*(1), 27-34.

[4] K. Hermanto and T. D. Ermayanti, “Analisa Optimasi Rute Transportasi Antar Jemput Siswa Menggunakan Model CGVRP dan Algoritma Dijkstra di SDIT Darus Sunnah,” Unisda J. Math. Comput. Sci., vol. 5, no. 2, pp. 19–28, 2019, doi: 10.52166/ujmc.v5i2.1653.

[5] M. A. F. Nugroho, Y. W. Syaifudin, and D. Puspitasari, “Penentuan Jarak Terpendek Menggunakan Metode Dijkstra Pada Data Spasial Openstreetmap (Studi Kasus : Pada Perusahaan Pengantaran Barang Wahana Logistik Kota Malang),” Smatika J., vol. 9, no. 01, pp. 45–50, 2019, doi: 10.32664/smatika.v9i01.265.

[6] Martin Nugroho Parapat, Deddy Kusbianto, C. R. (2017). Rancang Bangun Aplikasi Pencarian Rute Terpendek Jasa Kiriman Barang Berbasis Mobile Dengan Metode Algoritma Dijkstra. Informatika Polinema, 1(2), 15–19.

[7] N. Dewi, D. A. W. P, and M. Purwaningsih, "Transjakarta Terdekat Dengan Algoritme Dijkstra Berbasis Application of Routing and Time Determination To the Nearest Transjakarta Bus Stop With Dijkstra Algorithm Based on Location Base System," vol. 7, no. 4, pp. 653–660, 2020, doi: 10.25126/jtiik.202071680

[8] Oktoviana, Mohamad, Yasindan Lucky, T., & Sholichin, R. (2017). Implementasi Algoritma Dijkstra Dalam Pencarian Lintasan Terpendek Lokasi Rumah Sakit, Hotel Dan Terminal Kota Malang Berbasis Web. Jurnal Informatika, 4(3), 3993–4000.

[9] Wulandari, I. A., & Sukmasetya, P. (2022). Implementasi Algoritma Dijkstra untuk Menentukan Rute Terpendek Menuju Pelayanan Kesehatan. *Jurnal Ilmiah Sistem Informasi (JISI)*, *1*(1), 30-37.