International Journal of Computer, Network Security and Information System (IJCONSIST)
Vol. 5 No. 2, Maret 2024

Implementation of A* Algorithm and Contraction Hierarchies for Delivery Route Optimization (Case Study: CV. Almaed.id)

Boy Erdyansyah Gunawan
Computer Science Faculty
UPN "Veteran" East Java
Surabaya, Indonesia
21081010021@student.upnjatin.ac.id

Fawwaz Ali Akbar
Computer Science Faculty
UPN "Veteran" East Java
Surabaya, Indonesia
Fawwaz_ali.fik@upnjatim.ac.id

Mohammad Idhom
Computer Science Faculty
UPN "Veteran" East Java
Surabaya, Indonesia
idhom@upnjatim.ac.id

Prismahardi Aji Riyantoko
Department of Information and Communication System
Okayama University
Okayama 700-8530, Japan
pnai2m3s@s.okayama-u.ac.jp

In the digital era, manufacturing companies like CV. Almaed.id are required to have an efficient distribution system to compete in the furniture industry. This study proposes the application of the A* algorithm and Contraction Hierarchies (CH) to optimize product delivery routes. This system utilizes road network data from OpenStreetMap and calculates geographic distances using the Haversine method. Implementation results show that the combination of A*, CH, and Haversine can accelerate route calculation and reduce operational costs compared to manual methods.

Keywords : A*, Contraction Hierarchies, Haversine, OpenStreetMap, logistics, route optimization.

I. INTRODUCTION

The advancement of information and communication technologies in the digital era has driven significant transformations in logistics and supply chain management. Amid the growing consumer demand for speed, precision, and service efficiency, distribution systems have become a critical factor in determining a company's competitiveness. This is especially relevant for businesses in the manufacturing and trade sectors, including the furniture industry, which heavily rely on product deliveries to various geographical locations as part of their daily operations.

CV. Almaed.id, a company engaged in the production and distribution of furniture, faces considerable challenges in managing product deliveries across regions with varying distances and geographic conditions. Currently, the route selection process is conducted manually or with the help of general-purpose navigation applications that are not integrated with the company's management system. This approach often results in suboptimal routing, increased fuel consumption, delivery delays, and reduced customer satisfaction due to inconsistent delivery times. These issues highlight the urgent need for an automated decision-making system that can efficiently and dynamically determine the best delivery routes.

One relevant approach to address this challenge is the implementation of shortest-path search algorithms. The A* algorithm is a heuristic method that has proven effective for finding optimal paths in weighted graphs, especially in navigation and geographic information systems (GIS). A* combines the strengths of Dijkstra's algorithm with a best-first search strategy, enabling faster route computation by considering heuristic distance estimates to the destination. However, in large and complex road networks, the efficiency of A* can be further improved using the Contraction Hierarchies (CH) technique. CH preprocesses the road graph by removing less significant nodes and replacing them with shortcuts, which reduces the computational load during pathfinding operations.

Additionally, to calculate distances between locations based on geographic coordinates, the Haversine formula is used. This method accounts for the curvature of the Earth, producing more accurate distance estimates compared to the Euclidean method. This accuracy is essential for real-time, coordinate-based routing systems.

The system developed in this study utilizes road network data from OpenStreetMap (OSM), an open-source mapping platform that provides comprehensive and regularly updated geospatial data. OSM data is converted into graph representations using Osmgraph, a tool that facilitates the integration of spatial data into routing algorithms. Built on web and mobile-based architecture, this system allows for real-time monitoring and management of delivery routes, adapting dynamically to field conditions and offering easy access for users.

By combining the A* algorithm, Contraction Hierarchies, and the Haversine method, this system aims to solve classical distribution issues related to inefficient route selection. It not only helps reduce operational costs but also improves distribution planning accuracy and customer service performance. This research focuses on a case study at CV. Almaed.id and is expected to serve as a model for route optimization systems in similar companies within the manufacturing and logistics industries.

II. METHODOLOGY

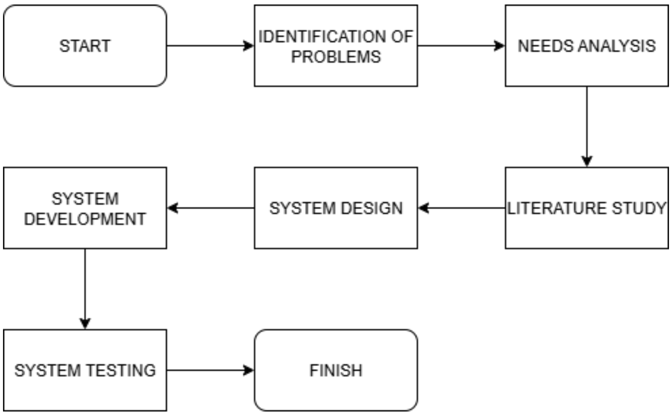

Fig. 1 Flow diagram

A. Problem Identification

In the initial stage, the main issue faced by CV. Almaed.id in its furniture distribution process was identified. The company struggles to optimize its product delivery routes due to continued reliance on manual methods and generic navigation applications that are not tailored to the specific needs of its business operations.

B. Requirements Analysis

An analysis was conducted to determine the system requirements for an effective route optimization solution for CV. Almaed.id. This stage aimed to identify the appropriate algorithms and technologies that can help optimize delivery routes based on various relevant factors such as road characteristics, travel distances, and existing infrastructure.

C. Literature Review

A literature review was carried out to gather references related to technologies and algorithms that could be used for route optimization. Several algorithms were identified, including A*, Contraction Hierarchies (CH), and the Haversine method. The system also requires the use of geospatial data from platforms like OpenStreetMap and Google Maps to build an efficient road network model.

D. System Design

A route optimization system was designed that integrates the A* algorithm, Contraction Hierarchies, and the Haversine method. The system also utilizes road network data sourced from OpenStreetMap and Google Maps, processed using Osmgraph to construct a road graph. This design ensures that the calculated routes are the most efficient in terms of time and cost, while also considering various geospatial factors.

E. System Development

The system was implemented according to the proposed design. It was built using pathfinding algorithms such as A*, Contraction Hierarchies, and Haversine to determine the most efficient delivery routes. The system is also capable of receiving real-time location data to dynamically manage product deliveries.

F. System Testing

After development, the next stage was to conduct system testing to ensure that the algorithms and all system components functioned as expected. This testing aimed to evaluate the efficiency and accuracy of the generated routes. Once testing was completed and the system performed as intended, the research phase was considered complete. The outcomes of this study are expected to provide a practical solution for CV. Almaed.id to optimize their delivery process and enhance their competitiveness in the increasingly demanding furniture industry.

Fig. 2 System Architecture

This system consists of two main interfaces: a web admin interface built with Laravel and a mobile application built with React Native. Laravel is connected to a MySQL Database and utilizes OpenStreetMap to display maps on the admin dashboard. Meanwhile, the mobile app uses GPS to obtain the user's location and displays the map using Google Maps. FastAPI serves as the API bridge that connects the mobile application to the database. This architecture clearly separates the admin and user functions and leverages different map services according to each platform.

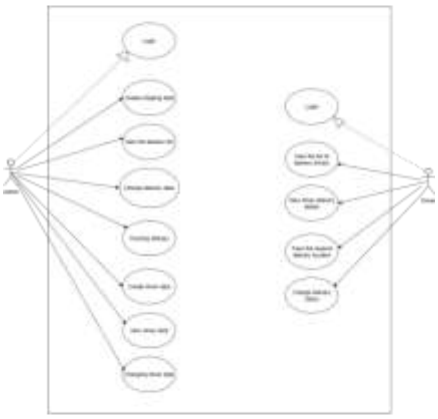

Fig. 3 Use case Diagram

The Use Case Diagram illustrates how two main actors Admin and Driver interact with the system to manage and monitor the delivery process. On the Admin side, the main role is to manage delivery data and ensure smooth operations. The Admin logs into the system to access key features such as creating delivery records (including destination, date, and item details), viewing and updating delivery lists, and modifying delivery information when necessary (e.g., changes in address or schedule). Admins are also responsible for tracking ongoing deliveries, managing driver data, and updating driver details as needed.

On the Driver side, access is limited to tasks related to delivery execution. Drivers must log in to view assigned deliveries, access detailed information about each delivery, and locate the nearest delivery points using GPS to determine the best routes. Drivers also update delivery statuses to reflect progress, such as marking deliveries as completed or in progress.

Both roles work collaboratively: the Admin manages data and oversees operations via the system, while the Driver performs deliveries in the field. This interaction ensures an efficient and timely delivery process. Table 3.1 outlines the login interaction flow between the Admin and the system.

III. RESULT AND DISCUSSION

Fig 4. Login page

The login interface serves as the initial entry point used to authenticate users before accessing the system. Users are required to input a registered email and password. Upon successful login, the system redirects users based on their roles for instance, administrators are directed to the web dashboard, while drivers are redirected to the mobile application. This feature is essential to ensure system security and to restrict access solely to authorized users.

Fig.5 Track location page

This interface represents the Track Location page, which displays the real-time position of the user or driver on a map. It includes a "Refresh Route" button to update delivery routes and a "Back" button to return to the previous page. This feature enables administrators to monitor delivery movements live, ensuring better coordination and tracking of the shipment process.

Fig 6. Delivery Location page

The Delivery Location page appears after activating the "Enable Location" and "Search Location" features. The system displays the driver's real-time position using a vehicle icon and illustrates the delivery route with a blue line connecting the starting point to the destination. Red markers indicate key points such as pickup and drop-off locations. This feature facilitates drivers in following the optimal route and assists the system in guiding deliveries efficiently and accurately.

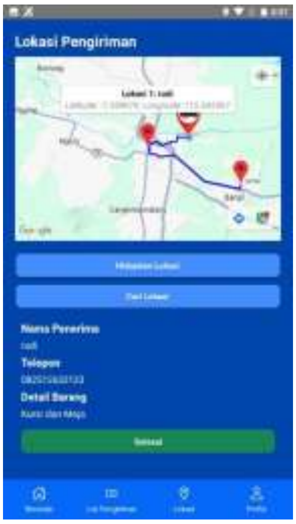

Fig. 7. Delivery page display

The Delivery Location page displays a complete delivery route along with detailed shipment information. At the top of the map, a location marker is shown with latitude and longitude coordinates, while a blue line illustrates the optimal route to the destination. Below the map, delivery details such as recipient name, phone number, and item being delivered are presented. A "Complete" button is provided to indicate that the delivery has been successfully completed. This feature helps drivers confirm delivery while also offering integrated location tracking and recipient data.

Fig. 8 part of the location

This feature is part of the location implementation on the mobile driver application, integrating the Google Maps API to display the driver's position and delivery route in real time. On this screen, the map shows a truck icon as the marker for the driver's current position, along with two red markers indicating the pickup and delivery locations. The delivery route is visualized with a blue polyline, generated using the *A and Contraction Hierarchies algorithms**, calculated over road network data extracted from OpenStreetMap (OSM).

Fig. 9. Detail Locatoon point

This display represents the "Detail Location Point" page, featuring an interactive map built with OpenStreetMap and the Leaflet.js library. On the map, a marker labeled "Business Location" indicates the coordinates of the main operational center or the starting point of delivery. Additionally, an orange-bordered area outlines the administrative boundaries of the service coverage, which includes the regions of Sidoarjo, Pasuruan, and Pasuruan City.

Users can adjust the delivery location by dragging the marker; however, the system will validate whether the selected point falls within the defined boundary. If the location is outside the allowed coverage, the system will trigger a warning and reset the marker to its previous position. This feature ensures that all delivery points remain within the designated operational area, maintaining service accuracy and preventing input errors.

Fig. 10. Track location page

The Track Location page is a component of the delivery monitoring system that features an interactive map based on OpenStreetMap. The main function of this page is the real-time tracking of the driver's position, assisting administrators in monitoring delivery movement from origin to destination. At the top of the page, there are two essential buttons: a yellow "Back" button to return to the previous page, and a blue "Refresh Route" button that allows the system to reload and update the delivery route automatically.

The map displays the driver’s current location and the pre-calculated delivery route. This visualization provides a clear overview of the path being taken and the current status of the shipment. With this tracking feature, the system enhances distribution efficiency, minimizes delays, and improves delivery security through direct supervision of the travel route.

Fig. 11. Delivery process page

During the delivery process, the React Native application on the driver's device continuously utilizes GPS to obtain real-time location data. This location data is then transmitted back to the system through FastAPI and stored in the database.

Fig. 12. Track Location page

The administrator can monitor the delivery status and the driver's location in real-time through the Laravel Web Admin. The integrated map preview, utilizing OpenStreetMap or Google Maps, allows the administrator to visualize the driver's position directly on the map.

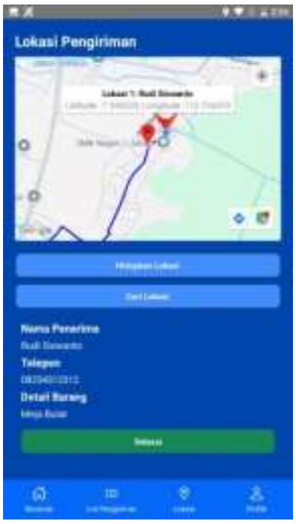

Fig. 13. Delivery location page

The driver can update the delivery status through the React Native application. This update is sent to the system via FastAPI and recorded in the Database, allowing the administrator to view the changes in real-time through the Laravel Web Admin.

Fig. 14. Real-time delivery page

This describes the map representation used in the route determination system as a graph structure. In this graph, key locations such as the starting point, destination points, and road intersections are represented as nodes, while the edges are the connecting paths between these nodes, corresponding to actual road segments. Each edge is assigned a weight, which represents the travel cost calculated in this study using the Haversine formula based on distance. The blue path visualized on the map indicates the shortest route computed by the A* algorithm, which considers both efficiency and directional accuracy within the graph's structure. This approach allows the system to perform route navigation more optimally and realistically, reflecting real-world road conditions.

One of the delivery destination points is Location 1: Rudi Siswanto, with latitude -7.554529 and longitude 112.726473. This point is represented as a node within the road network graph used by the system to calculate the delivery route. The coordinate information serves as a crucial input for the system to identify the nearest corresponding node, which is then used as a start or destination point in the shortest path search using the A* algorithm. The blue path leading to this point indicates the route that has been automatically calculated by the system.

Fig. 15. Point delivery page

In the bidirectional search process, the system calculates the route simultaneously from the starting point (vehicle icon) to the destination location, and vice versa from the destination back to the starting point. This approach accelerates the pathfinding process because both searches meet in the middle, resulting in more efficient computation time compared to a full one-directional search. The blue path on the map illustrates the final result of the shortest route found between these two points based on the road graph structure.

Bidirectional search in the A* algorithm operates by running two concurrent searches: a forward search from the start node to the goal node and a backward search from the goal node to the start node. The process terminates when both searches meet at a midpoint in the graph. This significantly reduces the search space compared to a full one-way search, leading to faster computation time, especially on large graphs such as city road networks.

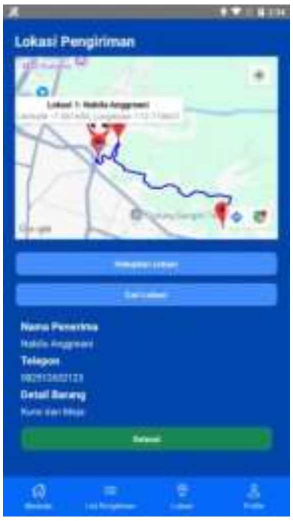

Fig. 16. Route result to Location page

The delivery route result to Location 1: Nabila Anggreani was generated by the system using the A* algorithm accelerated with Contraction Hierarchies (CH). The route is displayed as a blue line connecting the starting point of the delivery (marked with a vehicle icon) to the destination point (marked with a red marker and coordinate information).

The route generated automatically by the system follows main roads, minimizes the number of intersections, and successfully avoids narrow or dead-end roads that are commonly encountered when using manual route planning. Additionally, there are no indications that the system selected disconnected or inefficient paths, demonstrating the accuracy of the computed route.

This visual output also supports the claim that the structure and direction of the system's path closely resemble the results generated by Google Maps, making it reliable and practical for drivers to follow without needing further manual adjustments.

IV. Conclusion

Based on the system design and testing results, this research concludes that the integration of the A* algorithm, Contraction Hierarchies (CH), and the Haversine formula provides an effective solution for optimizing furniture delivery routes. A* ensures optimal routing using a heuristic approach, CH significantly accelerates the search through preprocessing by removing non-essential nodes and adding shortcuts, while the Haversine method accurately calculates geodetic distances by accounting for Earth's curvature. Together, these techniques enable the system to compute efficient routes in terms of distance and travel time, while reducing computational load especially for large and complex road networks.

The implemented system enables CV. Almaed.id to automate and optimize product deliveries using accurate, real-time geographic data. It replaces inefficient manual and generic navigation tools previously used, leading to better fuel efficiency, reduced delivery times, and fewer delays. This translates into lower logistics costs and improved delivery reliability for customers.

Built on OpenStreetMap road data processed via Osmgraph, the system supports both web and mobile platforms, allowing flexible access and real-time monitoring. Its geospatial features adapt to road conditions and terrain characteristics, offering dynamic routing decisions. The use of open-source data also ensures continuous updates and community support. Overall, the system improves routing decisions and enhances delivery service quality through accurate, adaptive path planning.

References

[1] S. Strasser and G. Zeitz, *Optimizing Logistics in the Digital Age*, Berlin: Springer, 2022.

[2] S. Funke, "Shortest path algorithms in transportation networks," *Algorithms*, vol. 13, no. 5, pp. 123-138, 2020.

[3] M. Buchhold and D. Wagner, "Engineering Contraction Hierarchies for Faster Shortest-Path Queries," in *Proc. of the ACM SIGSPATIAL*, 2021.

[4] D. Proissl, *Efficient Pathfinding in Large-Scale Road Networks*, Munich: Technical Reports in Informatics, 2022.

[5] A. Sumaryo, F. Hadi, and B. Setiawan, "Comparative Analysis of Haversine and Euclidean in Route Calculation," *Jurnal Geomatika*, vol. 26, no. 3, pp. 89-97, 2020.

[6] [16] M. B. Shami, G. Kiss, T. A. Haakonsen, and F. Lindseth, "Geo-locating Road Objects using Inverse Haversine Formula with NVIDIA Driveworks," *arXiv preprint arXiv:2401.07582*, Jan. 2024.

[7] A. Jain, "Analyzing OpenStreetMap Road Network Attributes with NetworkX, PyG and Graph Neural Networks," *Medium*, Jan. 5, 2022.