EMITOR: Jurnal Teknik Elektro

Design and Development of a Mobile-Based Application Path Planning System for Autonomous Electric Vehicles with Ant Colony Optimization Algorithm

Melia Sari*, Desi Windi Sari, Puspa Kurniasari
Fakultas Teknik Jurusan Teknik Elektro – Universitas Sriwijaya
Palembang, Indonesia
*meliasari@ft.unsri.ac.id

Abstract *– The rapid advancement of technology has generated numerous innovations across various domains, including transportation. One notable development is the autonomous vehicle, a driverless system capable of navigating to a designated destination without human intervention. This study emphasizes two critical aspects: navigation and efficient path planning. The objective is to design and develop a mobile application for optimal path planning based on the Ant Colony Optimization (ACO) algorithm. The application was developed using Visual Studio Code as the integrated development environment (IDE) and implemented under the waterfall software development model. The ACO algorithm served as the core mechanism for path determination, supported by the Google Maps API to provide spatial data required for processing. Additionally, Firebase was employed for user authentication-such as registration and login-and for storing trip history. Testing results indicate that the developed mobile application successfully operates according to its intended functions. In particular, the system demonstrates the capability to determine the shortest path effectively through the implementation of the Ant Colony Optimization algorithm. These findings suggest that the proposed approach can support advancements in autonomous vehicle navigation systems by offering efficient and reliable path planning solutions.*

Keywords – *Mobile Application; Path Planning; Ant Colony Optimization;Autonomous Electric Vehicle; Visual Studio.*

I. Introduction

THE influence of technology cannot be denied, as it has brought numerous benefits to human life. It is difficult to imagine living without technology in the present era, since most daily human activities rely heavily on technological advancements in various aspects, including work, social life, and others [1]. One such example is the mobile phone, commonly referred to as a smartphone. Smartphones represent a form of technology that continues to advance rapidly in order to meet human needs. They are widely used to support daily activities, such as making phone calls, accessing the internet, sending messages, and utilizing a variety of other applications [2].

In addition, autonomous vehicles represent another technological breakthrough that will likely become increasingly common in the coming years [3]. Autonomous vehicles have already emerged as a major focus of research in the field of transportation, particularly in relation to commercial vehicles [4]. An autonomous vehicle is a driverless system capable of traveling to a predetermined destination without requiring human involvement [5]. To achieve this capability, the vehicle must be equipped with systems that can recognize environmental conditions and generate path planning based on its surroundings [6]. Efficient path planning requires the implementation of algorithms that enable autonomous vehicles to independently process and execute navigation tasks [7].

In previous studies, various methods have been proposed for path planning in autonomous vehicles. For instance, [8] applied ant colony optimization (ACO) combined with fuzzy logic for global path planning implemented in mobile robots. Furthermore, [9] employed a hybrid of the A* algorithm and ACO for 3D navigation in dense obstacle environments. Another study [10] adopted a PSO-ACO fusion algorithm for 3D path planning in unmanned underwater vehicles (UUVs). Similarly, [6] implemented path planning in autonomous underwater vehicles using an enhanced

The manuscript was received on September 27, 2025, revised on October 1, 2025, and published online on November 28, 2025. `Emitor` is a Journal of Electrical Engineering at Universitas Muhammadiyah Surakarta with ISSN (Print) 1411 – 8890 and ISSN (Online) 2541 – 4518, holding Sinta 3 accreditation. It is accessible at `https://journals2.ums.ac.id/index.php/emitor/index`.

ACO algorithm based on particle swarm optimization (PSO). Study [11] also utilized PSO hybridized with the modified frequency bat (MFB) algorithm for multi-objective path planning, while [12] applied a hybrid genetic algorithm (GA) and firefly algorithm (FA) for mobile robot path planning. In another work, [13] integrated GA with improved ACO for optimizing path planning in intelligent vehicles. Meanwhile, [14] designed a membrane evolutionary algorithm hybridized with artificial potential fields (APF) for mobile robot path planning. Study [15] employed an improved APF algorithm for local path planning in autonomous ground vehicles.

In addition, [16] used an APF hybridized with the A* algorithm for local path planning in autonomous vehicles while considering road constraints and various obstacles. Research [17] adopted a fusion method based on long short-term memory (LSTM) neural networks and reinforcement learning for local path planning in mobile robots. Furthermore, [18] implemented dynamic rolling windows for local path planning in mobile robots.

However, these algorithms cannot independently achieve optimal solutions in complex dynamic environments. Such methods often become inefficient when the target is located at a considerable distance or in irregular environments; for example, APF is prone to being trapped in local minima [19]. Fuzzy logic offers the ability to emulate human expert knowledge, but it incurs high computational costs when the number of inputs increases [20]. Genetic algorithms, as evolutionary approaches, are capable of solving optimization problems; however, they completely update good individuals without leveraging the structural characteristics of the solution space [21]. Particle swarm optimization (PSO) is suitable for optimization problems, but it tends to produce suboptimal results and may become trapped in local minima [22].

Previous studies have shown increasing interest in metaheuristic algorithms based on ant colony optimization in mobile robotic systems. The fundamental idea of ACO is to discover the optimal path from a nest to a food source within a graph, inspired by the foraging behavior of ants. The advantages of ACO include environmental adaptability, strong robustness, and ease of integration with other algorithms.

In addition, an application system is required to support the path search process that will be utilized by autonomous vehicles, particularly for end-users who will operate these vehicles [23]. The design of such an application aims to provide convenience for users in operating autonomous vehicles.

Based on this background, this research proposes the design of a mobile application system for path planning, with the Ant Colony Optimization (ACO) algorithm applied as the core path planning method. It is expected that this ACO-based path planning application will enable users to perform path searches more easily while simultaneously supporting the operation of autonomous vehicles [24].

II. Research Methods

This study employed several research methodologies, as outlined below:

i. Literature Review

The research began with a literature review. The researcher studied and analyzed previous works related to this topic. The literature review was conducted using various sources, including books, journals, learning modules, and undergraduate theses. The purpose of this stage was to gain a comprehensive understanding of the theories, concepts, and processes relevant to the research topic.

ii. Path Data Collection

Data collection was conducted by selecting 32 location points around the research area. The data consisted of longitude and latitude values obtained from the GPS feature in the Google Maps browser.

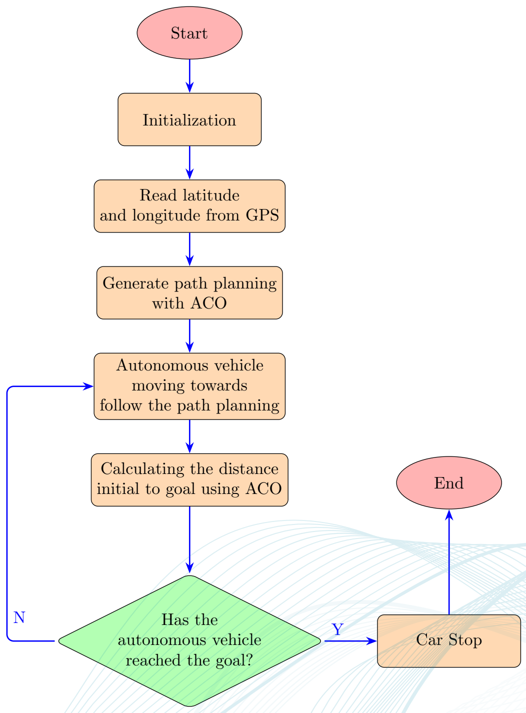

Start
Initialization
Read latitude and longitude from GPS
Generate path planning with ACO
Autonomous vehicle moving towards follow the path planning
Calculating the distance initial to goal using ACO
Has the autonomous vehicle reached the goal?
N
Y
Car Stop
End

Figure 1: Flowchart Path Planning System

iii. Path Planning Design

Path Planning design is illustrated in Fig. 1, showing the workflow of path determination for autonomous vehicles. The ACO algorithm is applied to calculate the optimal path.

The ACO algorithm simulates virtual ants that deposit pheromone trails along the paths they traverse. The pheromone intensity increases when ants successfully return to the nest with food, making highly traversed paths more attractive. Conversely, pheromones on less frequently visited paths gradually evaporate, preventing the algorithm from favoring suboptimal routes. Through this mechanism, virtual ants are guided toward discovering the optimal path to the food source [25].

The ACO algorithm operates as a simulation-based optimization method driven by positive feedback. In the path-searching process, two main factors are considered: the pheromone concentration heuristic and the distance heuristic. At time τ, the transition of ants between nodes is determined probabilistically, as formulated in the following equation [26]:

$$P_{ij}^{k}(t) = \begin{cases} \dfrac{\tau_{ij}(t)^{\alpha}\ \eta_{ij}(t)^{\beta}}{\sum\limits_{s \in allowed^{k}} \tau_{is}(t)^{\alpha}\ \eta_{is}(t)^{\beta}}, & j \in allowed^{k} \\ 0, & j \notin allowed^{k} \end{cases} \tag{1}$$

where α and β represent the weighting parameters of the two heuristic functions. The set of feasible next points is denoted as $allowed^{k}$. The pheromone concentration heuristic function is expressed as $\tau_{ij}(t)$, while the distance heuristic function is represented as $\eta_{ij}(t)$:

$$\eta_{ij} = \frac{1}{d_{ij}} \tag{2}$$

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{3}$$

Upon completion of the path planning process by all ants, the deposited pheromones undergo evaporation, a process in which their intensity gradually diminishes over time. The evaporation rate, denoted as ρ ($0 < \rho < 1$), reduces the amount of pheromone retained on the paths. The pheromone update mechanism can be formulated as follows:

$$\tau_{ij}(t+1) = (1-\rho)\ \tau_{ij}(t) + \Delta\tau_{ij}(t) \tag{4}$$

iv. Mobile Application Design

In designing the mobile application, the software waterfall method was adopted. Waterfall provides a systematic and structured approach to software development, ensuring that the mobile application is developed in a controlled and organized manner. This method was chosen because it is suitable for generic systems in which all requirements can be identified from the outset [27], offers predictable timelines, thorough documentation, and a well-defined scope [28]. Another reason is that waterfall is best suited for projects where the requirements are clearly specified and the outcomes can be anticipated [29].

v. Implementation

The coding process was carried out according to the design specifications. The user interface (UI) design was implemented, followed by the integration of the ACO algorithm for shortest-path searching within the application. Additionally, the required APIs were integrated into the developed application.

vi. Testing and Analysis

At this stage, the performance of the ACO algorithm within the application was tested to evaluate its ability to determine the shortest path. A comparison was then conducted between the results generated by the application and those provided by Google Maps. Finally, the live tracking feature of the application was also tested. The results of these tests were analyzed to assess the effectiveness and accuracy of the system.

III. Results and Discussion

i. Design Implementation

At this step, the researcher carried out the implementation based on the design that had been defined in the Software Development Life Cycle (SDLC) using the Waterfall model. The following section presents the results of the implementation.

As shown in Fig. 2(a), a simple login interface has been developed. Once the user has signed in, they can directly proceed to the path search interface, as illustrated in Fig. 2(b). In the path search interface, several operations are available, such as searching for the destination location using the *Search Address* button and confirming the selected destination.

After the location is confirmed, the user can search for the shortest path by pressing the *Generate Path* button, as shown in Fig. 3(a). Once the shortest path is generated, as illustrated in Fig. 3(b), the user may directly activate the live tracking feature by pressing the *Start Trip* button or cancel the generated path by selecting the *Cancel Path* option.

ii. Algorithm Testing with ACO

After the implementation phase, in order to ensure that the application can operate properly in performing the

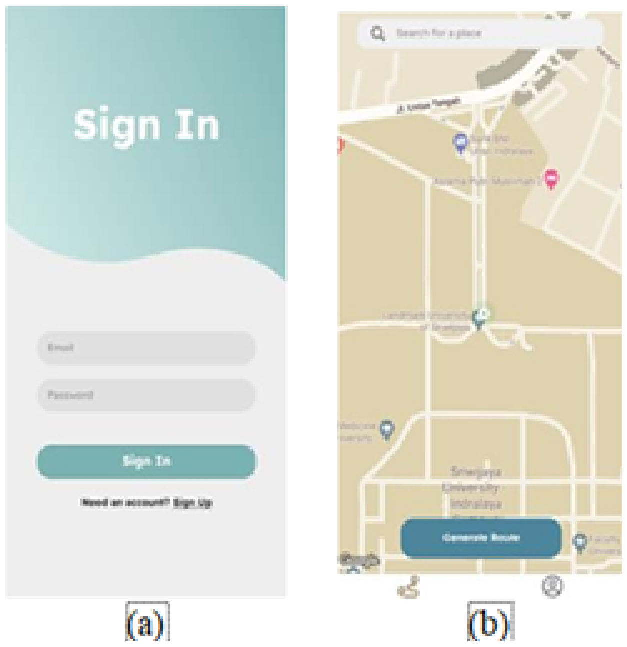


Figure 2: User Interface Design of the Mobile Application

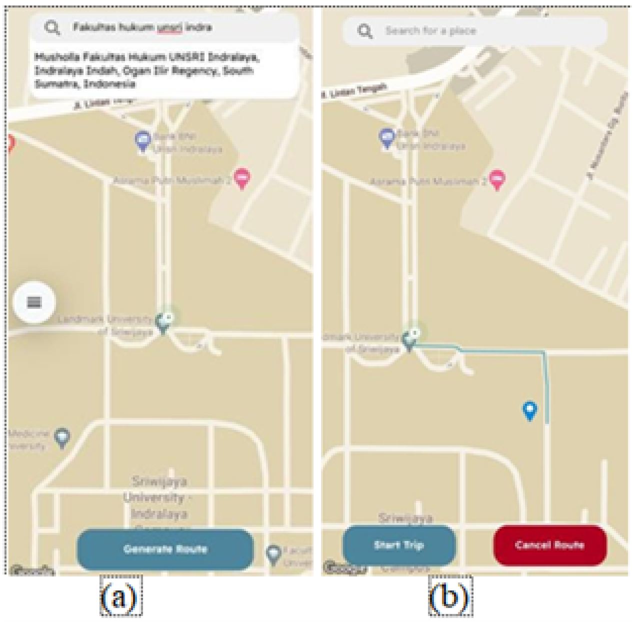

Fakultas hukum unsri indra
Musholla Fakultas Hukum UNSRI Indralaya, Indralaya Indah, Ogan Ilir Regency, South Sumatra, Indonesia
Search for a place
Generate Route
Start Trip
Cancel Route
(a)
(b)

Figure 3: Testing Scenario for Path Planning and Live Tracking

shortest path search, testing of the Ant Colony Optimization (ACO) algorithm was conducted using the distance data and path generation time obtained. The results of the testing are presented as follows. The testing results of the Ant Colony Optimization (ACO) algorithm implementation are summarized in Table 1
.

Based on Table 1, the total generation time required to process 10 test paths was 21.711 seconds, while the total distance covered in the 10 test paths was 6,282 meters. Furthermore, the average computation of the application's processing time and the average travel

Table 1: Testing Result

No	Path	Generate Time (s)	Path Length (m)
1	FT – FKM	1.418	515
2	FKM – Landmark	1.672	472
3	Landmark – FH	2.167	377
4	FH – FASILKOM	2.878	582
5	FASILKOM – FMIPA	1.989	584
6	FMIPA – Library	1.874	588
7	Library – FP	2.102	515
8	FP – FT	1.481	398
9	FH – Al Ghazali Mosque	2.672	757
10	Al Ghazali Mosque – ECO Machine Building	3.458	1,494
Total		21.711	6,282

distance during the path search were obtained. The results indicate that the average path generation time of the application is 2.1711 seconds, and the average travel distance obtained during the path search is 628.2 meters.

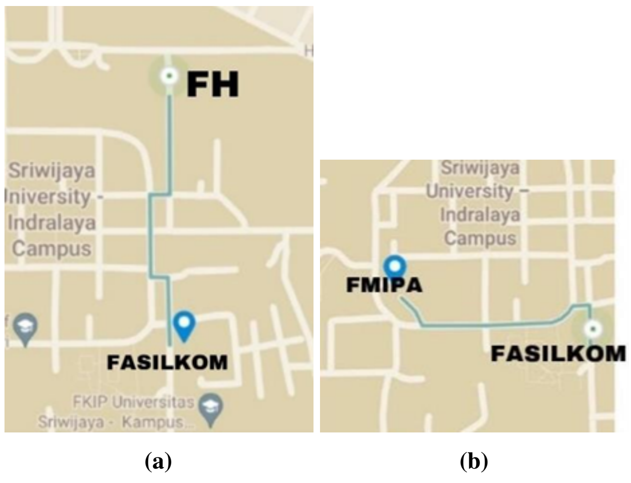

FH
Sriwijaya University Indralaya Campus
FASILKOM
FKIP Universitas Sriwijaya - Kampus
FMIPA
FASILKOM
(a)
(b)

Figure 4: (a) FH–FASILKOM Path, (b) FASILKOM–FMIPA Path

From this testing, it can be concluded that the ACO algorithm performs effectively with efficient processing time. Although two paths produced identical distances, they resulted in different generation times. These paths are FH – FASILKOM and FASILKOM – FMIPA. As presented in Table 1, the respective distances obtained were 582 meters and 584 meters; however, the generation times were 2.878 seconds and 1.989 seconds, respectively. This indicates a difference of nearly one second despite the similar distances. Such variation is attributed to the differing levels of path complexity encountered along the paths, as illustrated in Fig. 4.

At this stage, a comparison was conducted between the shortest path patterns generated by the application and those provided by Google Maps.

For the path Library – Faculty of Agriculture, a difference was observed in the generated paths. Similar

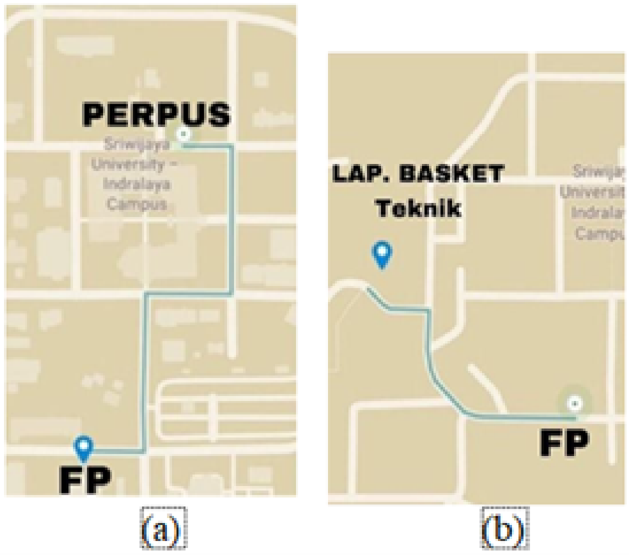


Figure 5: (a) Library–FP Path, (b) FP–FT Path

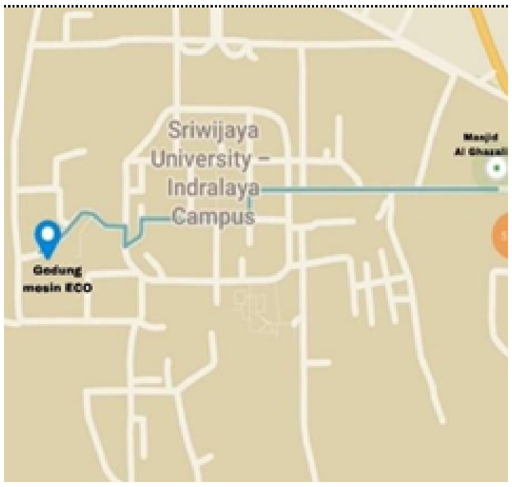


Figure 6: Al Ghazali Mosque–ECO Mechanic Building Path

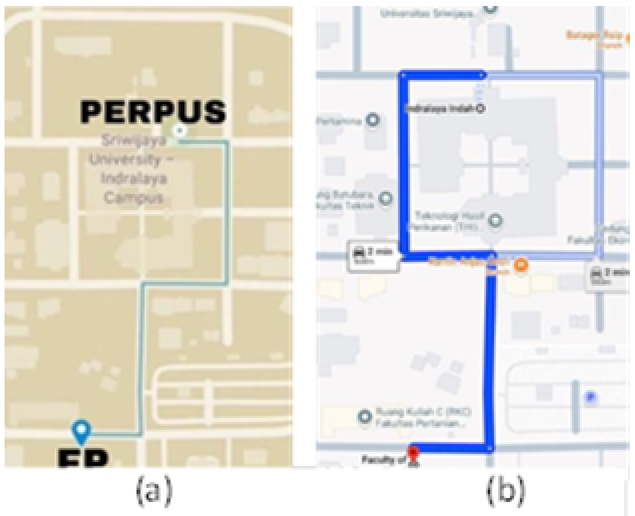


Figure 7: (a) Application Path, (b) Google Maps Path

to the FMIPA – Library path, Google Maps produced an alternative path with a distance of 550 meters, while the main path generated by Google Maps covered 500 meters. In this case, Google Maps demonstrated a shorter path of 500 meters, whereas the application produced a path of 515 meters. Thus, there is a distance discrepancy of 15 meters between the application and Google Maps.

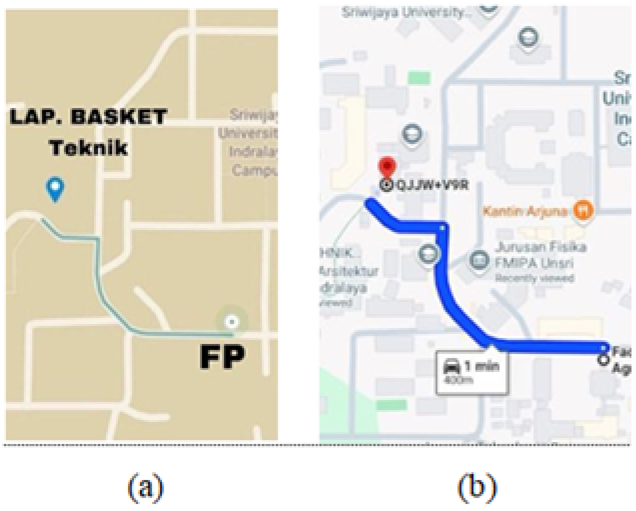


Figure 8: (a) Application Path, (b) Google Maps Path for Faculty of Agriculture – Engineering Basketball Court

For the path Faculty of Agriculture – Engineering Basketball Court, the generated path patterns were identical, with the difference observed only in the distance. The application produced a distance of 398 meters, whereas Google Maps generated 400 meters. Thus, there is a distance discrepancy of 2 meters between the application and Google Maps.

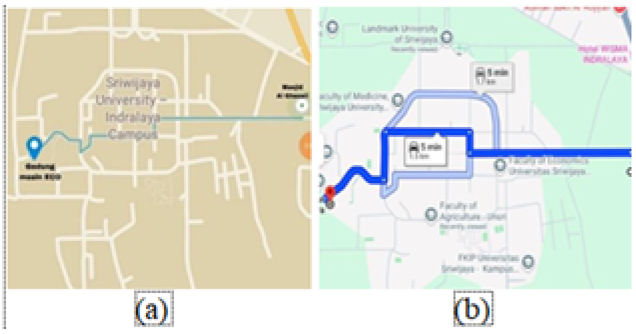


Figure 9: (a) Application Path, (b) Google Maps Path for Al Ghazali Mosque – ECO Mechanical Building

Finally, for the path Al Ghazali Mosque – ECO Mechanical Building, the generated path patterns differed. Google Maps produced three paths, namely a main path of 1,500 meters, the first alternative path of 1,700 meters, and the second alternative path of 1,500 meters. In contrast, the application generated a path with a distance of 1,494 meters. This indicates that the application outperformed Google Maps in terms of the shortest distance produced.

Following the comparative discussion of the path patterns generated by the application and Google Maps, the next section presents a comparison of the distances generated by both in tabular form.

Table 2: Comparison of Application and Google Maps Path Length

No	Path	Application Path (m)	Google Maps Path (m)
1	FT – FKM	515	500
2	FKM – Landmark	472	450
3	Landmark – FH	377	400
4	FH – FASILKOM	582	600
5	FASILKOM – FMIPA	584	600
6	FMIPA – Library	588	600
7	Library – FP	515	500
8	FP – Basketball Court FT	398	400
9	FH – Al Ghazali Mosque	757	750
10	Al Ghazali Mosque – ECO Mechanical Building	1,494	1,500
Total		6,300	6,297

Based on Table 2, it can be observed that there are differences in the path distances generated by the application and Google Maps. These discrepancies arise from two factors: first, the difference in distance calculation methods employed by Google Maps and the designed application. In the application, the Haversine formula is utilized to calculate the distance between two location points on the surface of a sphere.

This finding demonstrates that the error margin in the application is relatively minimal, at approximately 1.8 meters. Nevertheless, the performance of the shortest path search using the ACO-based path planning application can be considered highly satisfactory.

iii. Live Tracking Testing

At this stage, live tracking testing was conducted after the shortest path had been generated and the journey commenced. Once the journey begins, the application automatically performs a zoom-in, initiates live tracking, and provides a dynamic panning display in accordance with the GPS signal on the smartphone. When the GPS indicates arrival at the destination, a notification appears to inform the user that the journey has been completed. This occurs because, during the coding process, the system was programmed to trigger a completion notification once the GPS position of the application is within 20 meters of the destination.

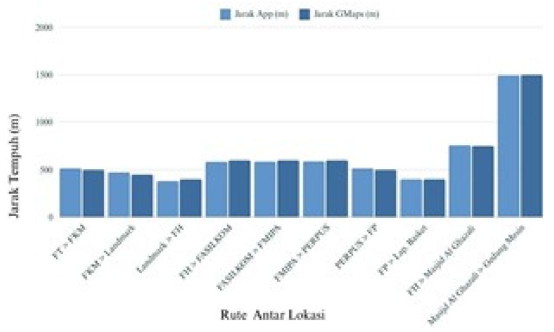
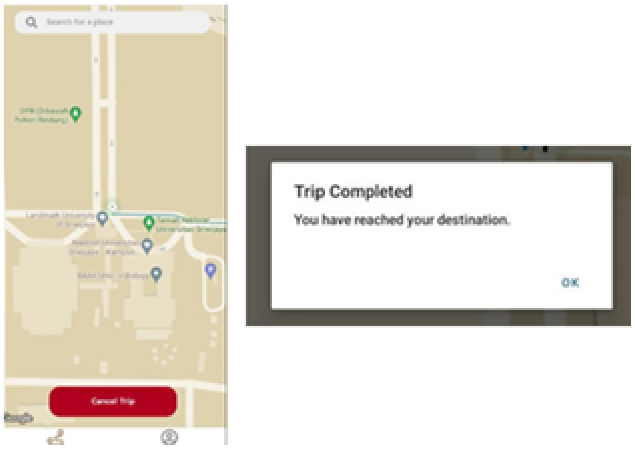


Figure 10: Graphical Comparison of Distances Generated by the Application and Google Maps



Trip Completed

You have reached your destination.

OK

Figure 11: (a) Live Tracking Interface, (b) Live Tracking Notification Interface

When live tracking begins, the application interface automatically zooms in, as illustrated in Fig. 11(a). As the smartphone's GPS updates during the journey, the application display automatically follows the GPS movement through panning. Once the user's smartphone GPS is within approximately 20 meters of the destination, the system generates a notification indicating that the user has reached the intended location, as shown in Fig. 11(b). The appearance of this completion notification signifies the end of live tracking, after which the application returns to its initial state on the finder screen.

IV. Conclusion

Based on the results of the tests presented in the previous chapter, the following conclusions can be drawn: The mobile path planning application based on the Ant Colony Optimization (ACO) algorithm for the UNSRI Indralaya campus was successfully developed and operates in accordance with the intended features. Comparative testing with Google Maps revealed distance discrepancies ranging from 2 to 23 meters, which were mainly caused by differences in calculation methods and GPS accuracy. Furthermore, the variation in path search time was influenced by the configuration of ACO parameters, route complexity, and total distance, indicating that more complex and longer routes required greater computation time.

Acknowledgment

The research and publication of this article were funded by Universitas Sriwijaya in 2025, in accordance with the Rector's Decree Number 0027/UN9/LPPM.PT/2025, dated September 17, 2025.

References

[1] G. Chen and J. Liu, "Mobile robot path planning using ant colony algorithm and improved potential field method," *Computational Intelligence and Neuroscience*, vol. 2019, 2019. [Online]. Available: https://doi.org/10.1155/2019/1932812

[2] J. Ni, Y. Chen, Y. Chen, J. Zhu, D. Ali, and W. Cao, "A survey on theories and applications for self-driving cars based on deep learning methods," *Applied Sciences (Switzerland)*, 2020.

[3] D. W. Sari, S. Dwijayanti, and B. Y. Suprapto, "Development of autonomous vehicle navigation in unstructured environments: The impact of implementing a path planning algorithm on autonomous vehicles," vol. 3, pp. 16–24, 2025. [Online]. Available: https://doi.org/10.15587/1729-4061.2025.323746

[4] H. J. Chae, "Uav path planning for local defense systems," in *Lecture Notes in Mechanical Engineering*, 2020, pp. 199–211. [Online]. Available: https://doi.org/10.1007/978-981-13-8323-6_17

[5] D. W. Sari, S. Dwijayanti, and B. Y. Suprapto, "Path planning for an autonomous vehicle based on the ant colony algorithm: A review," in *IWAIIP 2023 - International Workshop on Artificial Intelligence and Image Processing*, 2023, pp. 304–308. [Online]. Available: https://ieeexplore.ieee.org/document/10462744

[6] G. Che, L. Liu, and Z. Yu, "An improved ant colony optimization algorithm based on particle swarm optimization algorithm for path planning of autonomous underwater vehicle," *Journal of Ambient Intelligence and Humanized Computing*, 2019. [Online]. Available: https://doi.org/10.1007/s12652-019-01531-8

[7] J. Ou and M. Wang, "Path planning for omnidirectional wheeled mobile robot by improved ant colony optimization," in *Chinese Control Conference (CCC)*, vol. 2019-July, 2019, pp. 2668–2673. [Online]. Available: https://doi.org/10.23919/ChiCC.2019.8866228

[8] Y. Tao *et al.*, "A mobile service robot global path planning method based on ant colony optimization and fuzzy control," *Applied Sciences*, vol. 11, no. 8, 2021. [Online]. Available: https://doi.org/10.3390/app11083605

[9] X. Yu, W. Chen, T. Gu, and H. Yuan, "Aco - a*: Ant colony optimization plus a* for 3d traveling in environments with dense obstacles," *IEEE Transactions on Evolutionary Computation*, vol. PP, no. c, p. 1, 2018. [Online]. Available: https://doi.org/10.1109/TEVC.2018.2878221

[10] Z. Cai, J. Liu, L. Xu, and J. Wang, "Uuv 3d path planning based on pso-aco fusion algorithm," *Robotics and Autonomous Systems*, vol. 179, p. 104748, 2024. [Online]. Available: https://doi.org/10.1016/j.robot.2024.104748

[11] R. Paper, B. Tang, Z. Zhu, and J. Luo, "Hybridizing particle swarm optimization and differential evolution for the mobile robot global path planning," 2016. [Online]. Available: https://doi.org/10.5772/63812

[12] T. Wei, Z. Guang, H. Xu, X. Sheng, and Z. Tao, "A new hybrid algorithm for path planning of mobile robot," *Journal of Supercomputing*, 2021. [Online]. Available: https://doi.org/10.1007/s11227-021-04031-9

[13] K. Shi *et al.*, "Path planning optimization of intelligent vehicle based on improved genetic and ant colony hybrid algorithm," *Frontiers in Bioengineering and Biotechnology*, vol. 10, pp. 1–17, 2022. [Online]. Available: https://doi.org/10.3389/fbioe.2022.905983

[14] U. Orozco-Rosas, O. Montiel, and R. Sepúlveda, "Mobile robot path planning using membrane evolutionary artificial potential field," *Applied Soft Computing Journal*, vol. 77, pp. 236–251, 2019. [Online]. Available: https://doi.org/10.1016/j.asoc.2019.01.036

[15] Z. Zhang, "Local path planning of unmanned underwater vehicle based on improved apf and rolling window method," in *Proceedings of the International Conference on Cyber-Physical Social Intelligence (ICCSI)*, 2022, pp. 542–549. [Online]. Available: https://doi.org/10.1109/ICCSI55536.2022.9970658

[16] C. Park and S. C. Kee, "Online local path planning on the campus environment for autonomous driving considering road constraints and multiple obstacles," *Applied Sciences*, vol. 11, no. 9, 2021. [Online]. Available: https://doi.org/10.3390/app11093909

[17] N. Guo, "Local path planning of mobile robot based on long short-term memory neural network," *Automation and Remote Control*, vol. 55, no. 1, pp. 53–65, 2021. [Online]. Available: https://doi.org/10.3103/S014641162101003X

[18] J. H. Zhang, "Local path planning of mobile robot based on self-adaptive dynamic window approach," in *Journal of Physics: Conference Series*, vol. 1905, no. 1, 2021. [Online]. Available: https://doi.org/10.1088/1742-6596/1905/1/012019

[19] A. Vagale, R. T. Bye, R. Oucheikh, O. L. Osen, and T. I. Fossen, "Path planning and collision avoidance for autonomous surface vehicles ii: a comparative study of algorithms," *Journal of Marine Science and Technology*, vol. 26, no. 4, pp. 1307–1323, 2021. [Online]. Available: https://doi.org/10.1007/s00773-020-00790-x

[20] B. Tutuko, S. Nurmaini, and G. Ogi, "Fuzzy logic-ant colony optimization for explorer-follower robot with global optimal path planning," *Computer Engineering and Applications Journal*, vol. 7, no. 1, pp. 61–74, 2018. [Online]. Available: https://doi.org/10.18495/comengapp.v7i1.241

[21] A. H. Karami and M. Hasanzadeh, "An adaptive genetic algorithm for robot motion planning in 2d complex environments," *Computers and Electrical Engineering*, 2015. [Online]. Available: https://doi.org/10.1016/j.compeleceng.2014.12.014

[22] H. Qin, S. Shao, T. Wang, X. Yu, Y. Jiang, and Z. Cao, "Review of autonomous path planning algorithms for mobile robots," 2023.

[23] R. Yang and L. Cheng, "Path planning of restaurant service robot based on a-star algorithms with updated weights," in *Proceedings of the 12th International Symposium on Computational Intelligence and Design (ISCID)*, vol. 1, 2019, pp. 292–295. [Online]. Available: https://doi.org/10.1109/ISCID.2019.00074

[24] L. Wu, X. Huang, J. Cui, C. Liu, and W. Xiao, “Modified adaptive ant colony optimization algorithm and its application for solving path planning of mobile robot,” *Expert Systems with Applications*, vol. 215, p. 119410, 2023. [Online]. Available: https://doi.org/10.1016/j.eswa.2022.119410

[25] D. W. Sari, S. Dwijayanti, and B. Y. Suprapto, “Integration of regression-based guidance ant for enhanced exploration and convergence in ant colony optimization (aco),” *IEEE Access*, vol. 13, pp. 107 621–107 630, 2025. [Online]. Available: https://doi.org/10.1109/ACCESS.2025.3581425

[26] W. Sari, “Ant colony optimization-based path planning for autonomous vehicle navigation systems,” 2024, pp. 135–140.

[27] N. Rachma and I. Muhlas, “Comparison of waterfall and prototyping models in research and development (r&d) methods for android-based learning application design,” *Jurnal Inovasi Teknologi Informasi dan Inform*, vol. 5, no. 1, p. 36, 2022. [Online]. Available: https://doi.org/10.32832/inova-tif.v5i1.7927

[28] B.-A. Andrei, “A study on using waterfall and agile methods in software project management,” *Journal of Information Systems and Operations Management*, pp. 125–127, 2019.

[29] F. Ji and T. Sedano, “Comparing extreme programming and waterfall project results,” in *24th IEEE-CS Conference on Software Engineering Education and Training (CSEE&T)*, 2011, pp. 482–486. [Online]. Available: https://doi.org/10.1109/CSEET.2011.5876129