ADAPTIVE PORTABLE INTELLIGENT TRAFFIC SIGNAL CONTROLLER (APILL): A DUAL-PURPOSE TOOL FOR REAL-TIME TRAFFIC MANAGEMENT AND COMPETENCY-BASED LEARNING

Putu Eka Suartawan[1]*, Riz Rifai Oktavianus Sasue[2], Desak Made Leony Krisna Rossanthi1
[1]Manajemen Transportasi Jalan, Politeknik Transportasi Darat Bali, Jl. Cempaka Putih, Samsam, Tabanan, Bali 82111, Indonesia
[2]Teknologi Otomotif, Politeknik Transportasi Darat Bali, Jl. Cempaka Putih, Samsam, Tabanan, Bali 82111, Indonesia
*Putu.eka@poltradabali.ac.id

ABSTRACT
The Portable Traffic Signal Controller (APILL) system was developed as a wireless traffic management tool, leveraging Bluetooth technology and controlled by an Arduino Uno R3 microcontroller. It features an application-based platform for calculating intersection cycle times, designed to enhance educational competencies in road equipment management. The research followed a structured methodology, including a theoretical review, conceptual development, field data collection, analysis, validation, and conclusions. The results demonstrated that cycle time calculations performed using both Excel and the Python-based application, in accordance with the Indonesian Highway Capacity Manual (PKJI) 2023 guidelines, were consistently synchronized and accurately visualized through the APILL signal lamp operations. However, minor discrepancies were identified during real-world testing, with the green light duration for the northern approach observed at 7.1 seconds compared to 6 seconds in Excel and 7 seconds in the application. These discrepancies were primarily attributed to rounding errors in calculations and manual inaccuracies during stopwatch-based observations. This research underscores the reliability and practicality of the Portable APILL system as both a traffic management tool and an effective educational medium. Furthermore, it highlights the importance of precision in calculation and observation to optimize its performance. The system holds significant potential for advancing competencies in traffic signal management and the operation of road equipment.

Keywords: APILL; cycle time calculation; PKJI 2023; portable traffic signal controller; traffic management education; wireless control

INTRODUCTION
Traffic signal controllers play a critical role in enhancing road safety and mobility by managing traffic flow at intersections. However, current manual operations often require on-site interventions, leading to delays and inefficiencies in traffic management (Adeleke et al., 2023). This reliance on manual control poses significant challenges, especially in urban areas where high traffic volumes demand timely responses to prevent congestion and accidents (Baz et al., 2020; Saeed, 2024). Addressing these issues requires innovative solutions that optimize traffic signal operations. To overcome these limitations, there is an increasing need for portable traffic signal systems that incorporate wireless and automated technologies. Such systems allow remote monitoring and control, enabling real-time adjustments based on traffic conditions without the necessity for physical presence at the site (Jin & Ma, 2019; Huang et al., 2018). Moreover, advancements in data analytics and connected vehicle technologies provide opportunities for more dynamic and responsive traffic management, improving overall traffic flow and reducing delays (Saldivar-Carranza et al., 2023; Islam et al., 2021). Automated systems not only enhance operational efficiency but also promote safer road environments by minimizing human error in traffic control (Dinh & Tang, 2017;(Neetu Gupta et al., 2023).

Beyond addressing traffic management inefficiencies, portable and automated systems hold significant potential as educational tools in traffic engineering. By integrating modern technologies such as wireless communication and Python-based automation, these systems can provide hands-on learning experiences for students and practitioners. This dual-purpose approach ensures that future traffic engineers are equipped with the skills to implement intelligent transportation systems effectively while addressing real-world traffic challenges. This study explores the development of a Python-based Portable Adaptive Traffic Signal Controller (APILL) system, designed not only to optimize traffic flow but also to serve as a practical educational medium for enhancing competencies in traffic signal management.

The development of a Python-based Adaptive Portable Intelligent Traffic Signal Controller (APILL) system, controllable via Android, addresses a significant gap in traffic management education. Current educational tools often lack real-time adaptability and integration with mobile technologies, which are essential for effective training in modern traffic management practices. The integration of wireless communication and automated systems is crucial for enhancing the functionality and accessibility of traffic signal control systems (Kamasetty et al., 2022;Gade, 2019). Existing research highlights the potential of adaptive traffic signal control systems that utilize real-time data to optimize traffic flow. For instance, studies have shown that adaptive systems can significantly reduce congestion and improve traffic efficiency by dynamically adjusting signal timings based on current traffic conditions (Jamil & Nower, 2021; Gu et al., 2020; Kim et al., 2023). Furthermore, the application of reinforcement learning techniques in traffic signal control has demonstrated promising results in optimizing signal operations across multiple intersections (Yan, 2024; Chen, 2020). By leveraging these advanced methodologies, a Python-based APILL system could provide a hands-on learning experience for students, allowing them to understand the complexities of traffic signal management in a controlled environment. This approach not only enhances educational outcomes but also prepares future traffic engineers to implement intelligent transportation systems effectively (Neetu et al., 2023; Ge et al., 2019).

METHOD

System Design and Development

The proposed APILL system will utilize various hardware components, including sensors, microcontrollers, and wireless modules. Sensors such as infrared or ultrasonic detectors will be employed to monitor vehicle presence and traffic flow at intersections. Microcontrollers, such as the Arduino or Raspberry Pi, will serve as the central processing units, managing sensor data and controlling signal outputs. Wireless modules, such as Wi-Fi or Bluetooth, will facilitate communication between the microcontroller and the Android application, enabling real-time data transmission and remote control capabilities.

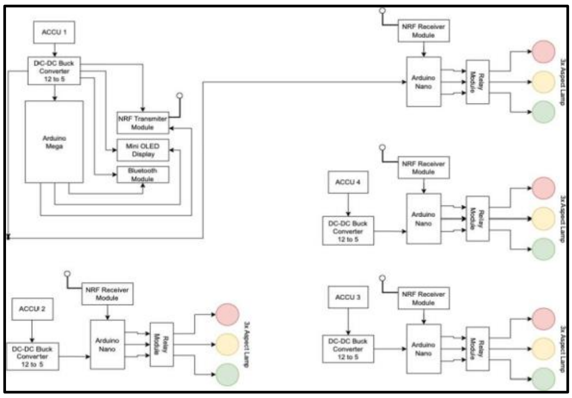

Figure 1. Electronic System Design

The software architecture will be primarily Python-based, leveraging libraries such as Flask for backend development. The Android application will be developed using Android Studio, incorporating Java or Kotlin for the user interface. The app will allow users to monitor traffic conditions, adjust signal timings, and visualize real-time data from the traffic management system. The integration of Python scripts with the Android app will enable seamless communication and control of the traffic signal controller. The integration between Python automation scripts and the Android interface will follow a structured workflow. Sensor data will be collected and processed by the microcontroller, which will execute Python scripts to determine optimal signal timings based on real-time traffic conditions. This information will be transmitted to the Android application, where users can view traffic data and make adjustments as necessary. The Android app will also send commands back to the microcontroller to modify signal operations, creating a bidirectional communication loop that enhances system responsiveness and adaptability.

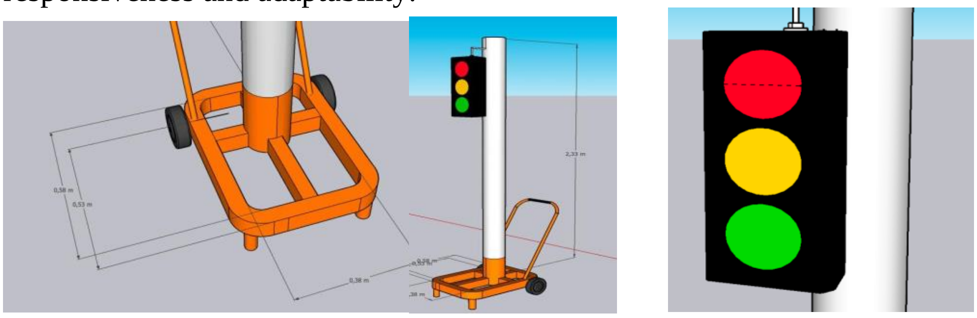

Figure 2. Portable APILL Design Planning

Validation Process

The validation process will begin with the calculation of traffic signal cycle times using the PKJI 2023 guidelines. This will be accomplished through both Excel and a Python application. In Excel, users will input traffic volume data and apply the PKJI formulas to derive optimal cycle times. Simultaneously, a Python script will be developed to automate this calculation, ensuring consistency and reducing human error. The Python application will utilize libraries such as Pandas for data manipulation and NumPy for numerical calculations, allowing for efficient processing of traffic data. To ensure accuracy and reliability, synchronization of cycle times between the Excel calculations, the Python application, and the APILL system will be implemented. This will involve exporting the calculated cycle times from Excel and importing them into the Python application. The Python application will then communicate with the APILL system to update the traffic signal timings in real-time. A logging mechanism will be established to track any discrepancies between the three systems, facilitating troubleshooting and ensuring that all components operate with the same cycle time data.

Data Collection

Following the observation phase, the collected data will be analyzed to compare the calculated cycle times (derived from the PKJI 2023 guidelines and processed through both Excel and Python applications) with the actual cycle times recorded during the observations. This comparison will help assess the accuracy of the cycle time calculations and the effectiveness of the APILL system in implementing optimal traffic signal control.

RESULT AND DISCUSSION

System Design Outcomes

The Python-based portable Adaptive Portable Intelligent Traffic Signal Controller (APILL) system was designed to enhance traffic management through several key features. It incorporates real-time traffic monitoring using various sensors, which provide data on vehicle presence and flow. The system is capable of dynamically adjusting traffic signal timings based on this data, optimizing traffic flow and reducing congestion. Additionally, the system includes a user-friendly Android application that allows operators to monitor traffic conditions, adjust signal timings, and receive alerts regarding system performance. The integration of cloud computing enables remote access and data storage, facilitating efficient management of traffic signals across multiple locations.

The APILL system leverages Python scripts to automate various functions, including data collection, cycle time calculations, and signal timing adjustments. The integration with the Android application allows for seamless control capabilities, enabling users to modify signal operations remotely. This bidirectional communication between the Python backend and the Android interface enhances the responsiveness of the traffic management system, allowing for timely adjustments based on current traffic conditions. The schematic wiring diagram of the system includes a master controller comprising an Arduino, Bluetooth module, breakout board, and an OLED display. This master controller functions as the primary interface for operating the Portable APILL system, enabling it to send commands to each slave unit at the intersection approaches. Each slave unit consists of an electronic system designed to receive data from the master controller via Bluetooth. The received data is processed by an Arduino within the slave unit, which then activates the respective traffic light aspects of the APILL using power supplied by a battery or accumulator. The activation of each traffic light aspect is visualized through the APILL lamps, controlled by relays assigned to each light. The installed devices at each intersection leg are depicted in Figure 3, showcasing the complete setup and integration of the system components.

Figure 3. Electronic Installation Results

Validation Result

This study focuses on validating the Adaptive Portable Intelligent Traffic Signal Controller (APILL) system implemented at the Batu Bolong – Babakan Canggu intersection, a four-legged intersection with protected approaches and a median, predominantly surrounded by commercial land use. The traffic analysis was conducted during the morning peak hours (07:00–08:00), with the observed traffic volumes as follows: 197 vehicles per hour (vph) for the northern approach, 329 vph for the southern approach, 443 vph for the eastern approach, and 465 vph for the western approach.

Cycle Time Calculation

The cycle time for the intersection was determined based on the performance metrics calculated using the PKJI 2023 guidelines, which included:

a) Saturation Flow Rates (J)
   - Northern approach: 2,952 vph (basic saturation flow, J_0 = 3,000 vph, and flow ratio, FR = 0.07).
   - Southern approach: 3,014 vph (J_0 = 3,180 vph, FR = 0.11).
   - Eastern approach: 3,648 vph (J_0 = 4,404 vph, FR = 0.12).
   - Western approach: 2,825 vph (J_0 = 3,480 vph, FR = 0.16).

   These saturation flow rates consider both basic saturation flow and correction factors to account for deviations from ideal conditions.

b) Green Time (G)

   The green times for each phase were calculated based on the saturation flow rates and protected approach type:
   - Northern approach: 6 seconds
   - Southern approach: 11 seconds
   - Eastern approach: 12 seconds
   - Western approach: 16 seconds

   The western approach exhibited the highest flow ratio (FR), initial flow ratio (IFR), and peak ratio (PR), necessitating its longer green time allocation of 16 seconds.

c) Amber Time and All-Red Phase

   The yellow (amber) signal duration was uniformly set at 3 seconds across all approaches, while the all-red phase lasted 2 seconds

d) Total Cycle Time

   The total cycle time for the intersection was calculated to be 65 seconds, incorporating all phases

System Performance

The integration workflow between Python and Android in the Portable APILL system is designed to ensure seamless communication and control. The workflow begins with data collection from sensors installed at the intersection, which detect vehicle presence and traffic flow. This data is processed in real time by the Arduino microcontroller running Python scripts, which calculate optimal signal timings based on the Indonesian Highway Capacity Manual (PKJI) 2023 guidelines. The Python backend utilizes libraries such as Pandas and NumPy to handle data manipulation and computation efficiently. Once the cycle time calculations are completed, the results are transmitted via a Bluetooth module to the Android application. The Android app, developed using Android Studio with Java or Kotlin, serves as a user-friendly interface for monitoring and controlling the traffic signals. Through the app, users can visualize real-time traffic data, adjust signal timings if necessary, and send commands back to the microcontroller. This bidirectional communication loop ensures that any changes made on the app are immediately reflected in the signal operations.

Additionally, the integration allows for logging and visualization of cycle times within the app, providing users with insights into the system's performance. This streamlined workflow enhances the responsiveness and adaptability of the APILL system, enabling real-time adjustments and efficient traffic management while maintaining simplicity and accessibility for educational purposes.

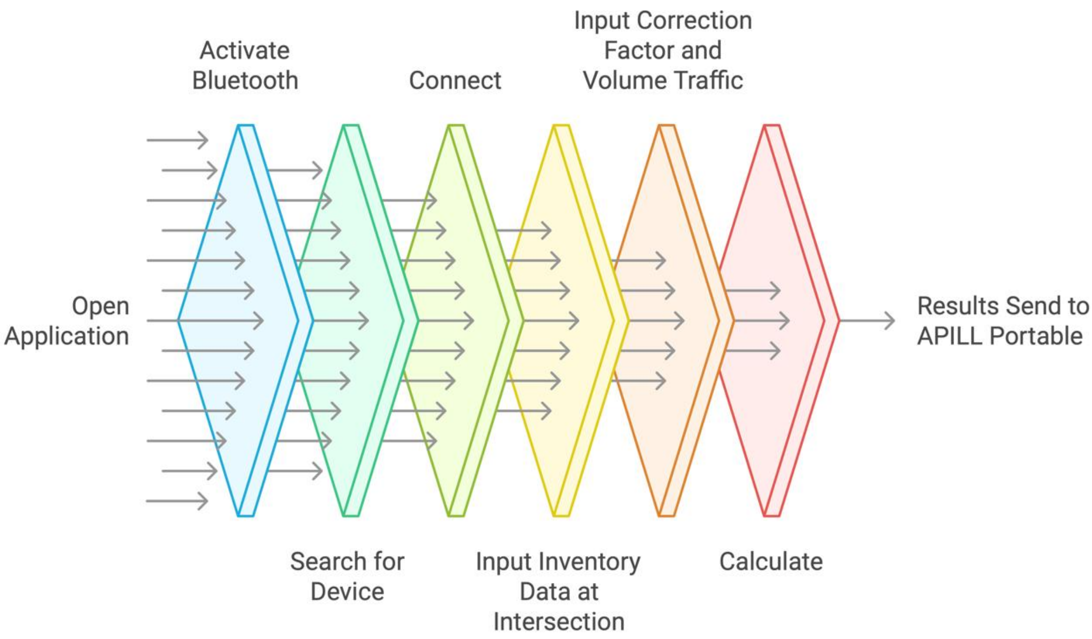

Figure 4. Python-Android Integration Workflow

After inputting data for each approach, the Python-based application automatically calculated and displayed the cycle time for each APILL phase. These results were then transmitted to the portable APILL system for real-time operation. The key findings revealed a green time distribution of 6 seconds for the northern approach, 16 seconds for the western approach, 11 seconds for the southern approach, and 12 seconds for the eastern approach. The yellow (amber) signal duration was uniformly set at 3 seconds across all approaches, while the longest red signal time was observed for the southern approach, lasting 55 seconds. These results demonstrate the system's capability to allocate signal timings dynamically based on approach-specific traffic conditions.

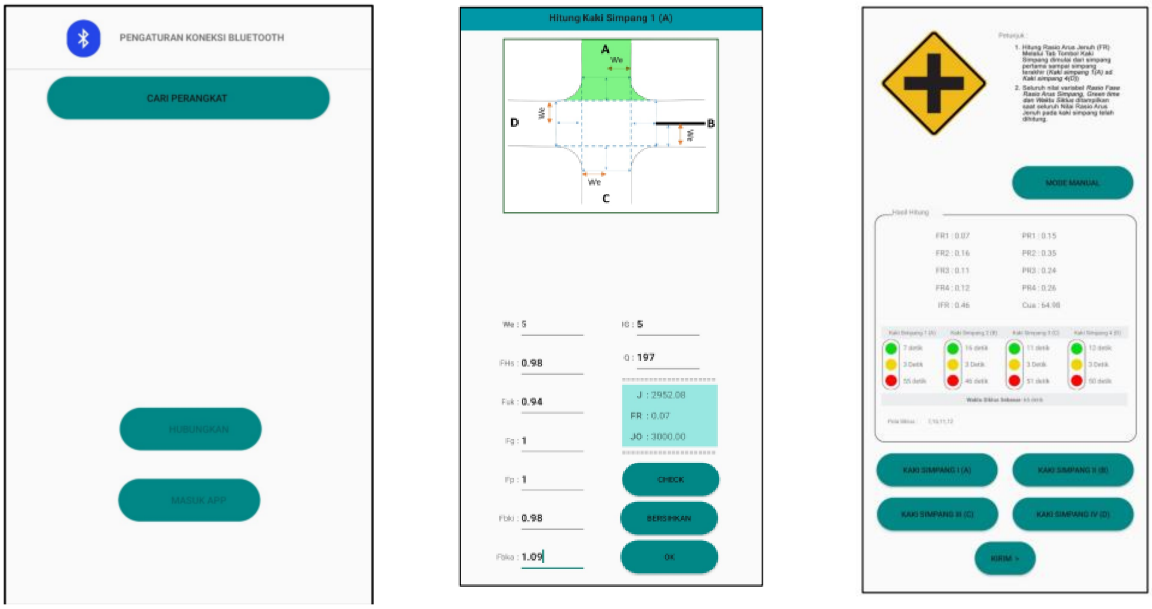

Figure 5. Aplication Process

The validation results indicate a comprehensive comparison of cycle time calculations derived from three distinct methods: Excel, the Python application, and real-world testing. The cycle times calculated using Excel and the Python application were found to be consistent, with both methods yielding a cycle time of 65 seconds for the Batu Bolong – Babakan Canggu intersection. However, during real-world testing, the observed cycle time was slightly different, reflecting the complexities of live traffic conditions. This discrepancy highlights the importance of real-time data and adaptability in traffic signal management systems. The validation testing of the Portable APILL system was conducted to compare cycle time calculations between the application, Microsoft Excel, and the visualized lamp operations on the Portable APILL system. This testing was repeated five times under the same cycle time conditions to evaluate the stability of the APILL lamp operations. The results showed that the cycle time calculations from the application and Excel were consistent and synchronized with the lamp operations on the Portable APILL system. However, slight discrepancies were observed in the timing of the lamp operations. These differences were attributed to rounding errors in the calculation factors within Excel and inaccuracies by the surveyor in timing the lamp operations using a stopwatch during direct observation.

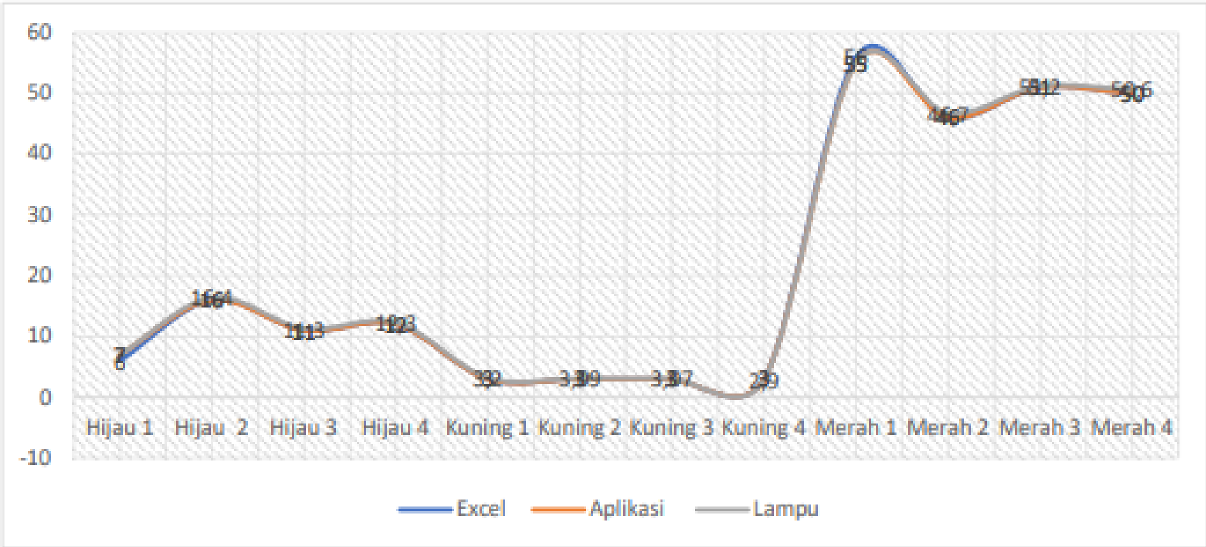

Figure 6. Cycle Time Calculation Curve with Counting Media

Figure 5 illustrates the comparative cycle time calculations derived from three distinct methods: Microsoft Excel, the Python-based application, and real-world observations of the Portable APILL system. The curve highlights discrepancies between these methods, particularly in the green light duration for the northern approach. As shown in the figure, the green time calculated using Excel is 6 seconds, while the Python application calculates 7 seconds, and the observed duration of the APILL lamp operation is 7.1 seconds.

This visualization underscores the minor variations caused by rounding processes in Excel and slight inaccuracies during manual stopwatch-based observations in real-world testing. The graphical representation in Figure 5 emphasizes the importance of synchronization across methods and highlights the need for precision in data rounding and observational techniques to ensure consistency. Furthermore, it demonstrates the overall stability of the Portable APILL system, as the total cycle time remained consistent across all methods at 65 seconds. By providing this detailed context, Figure 5 serves as a valuable tool for understanding the system's performance and the factors contributing to discrepancies in signal timing. This deeper insight supports the narrative in the results and discussion sections, emphasizing the reliability and potential areas for improvement of the Portable APILL system.

The Python-based Adaptive Portable Intelligent Traffic Signal Controller (APILL) system has demonstrated a high level of accuracy and reliability in calculating traffic signal timings. The integration of real-time data processing and automated calculations allows for precise adjustments to signal timings based on current traffic conditions. However, the system's performance is contingent upon the accuracy of the input data and the calibration of sensors used for traffic monitoring. Studies have shown that the reliability of traffic signal control systems can significantly impact overall traffic flow and congestion management (Ma et al., 2020; Win et al., 2018). Despite the system's capabilities, discrepancies in timing can arise from several factors. Rounding errors in calculations can lead to minor differences in green light durations, as seen in the observed 7.1 seconds compared to the 7 seconds calculated by the Python application. Additionally, manual observation inaccuracies during real-world testing can contribute to these discrepancies, as human error is an inherent risk in observational studies. The communication between the Android application and the APILL system via Bluetooth may introduce delays that affect the synchronization of signal timings. Latency in data transmission can lead to discrepancies between the intended signal changes and their actual implementation at the intersection. Hardware constraints, such as sensor limitations and communication delays between the Android application and the APILL system, may also affect the timing accuracy (Win et al., 2018). The performance of the traffic signal controller is heavily reliant on the accuracy of the sensors used to detect vehicle presence and flow. If sensors are miscalibrated or experience delays in response time, this can lead to incorrect data being fed into the system, resulting in suboptimal signal timings. For example, if a sensor fails to detect a vehicle promptly, the system may not adjust the green light duration appropriately, leading to increased delays and congestion. Understanding these factors is crucial for refining the system and enhancing its performance in practical applications.

Potential solutions that can be implemented include: establishing a routine calibration schedule for sensors is essential to maintain accuracy in detecting vehicle presence and flow. Regular checks can help identify and rectify any deviations in sensor performance, ensuring that the data collected reflects actual traffic conditions. This practice can significantly reduce discrepancies in signal timing caused by sensor inaccuracies. Upgrading to higher-resolution sensors can minimize detection delays and improve the responsiveness of the traffic signal system. Advanced sensors, such as those utilizing LiDAR or high-definition cameras, can provide more precise data on vehicle counts and movements, thereby enhancing the overall effectiveness of the APILL system. Transitioning from manual timing methods to automated timing tools can greatly enhance the accuracy of data collection. Integrating time-logging features directly into the APILL system allows for real-time recording of signal timings without human error. This automation ensures that the timing data is consistent and reliable.

Creating software that automatically logs and compares cycle times during testing can streamline the validation process. This software can analyze data in real-time, providing immediate feedback on performance and facilitating quicker adjustments to signal timings as needed. Implementing advanced rounding techniques in both Excel and Python scripts can help minimize discrepancies in timing calculations. Techniques such as rounding to the nearest significant figure or using statistical methods to determine optimal rounding thresholds can enhance precision in time-sensitive calculations. Integrating error margins into calculation algorithms can account for real-world variations in traffic conditions. By allowing for a buffer in timing calculations, the system can better adapt to fluctuations in traffic flow and improve overall performance.

The development and implementation of the APILL system have significant educational implications for training future traffic management professionals. By providing a hands-on learning experience, students can engage with real-time traffic data and understand the complexities of traffic signal control. The system serves as a practical tool for demonstrating theoretical concepts in traffic engineering, such as cycle time calculations and signal timing optimization. Moreover, the integration of modern technologies, such as Bluetooth communication and mobile applications, prepares students for the evolving landscape of intelligent transportation systems (Guo, 2023; Silas et al., 2021). This experiential learning approach is essential for equipping the next generation of traffic engineers with the skills needed to address contemporary traffic challenges effectively.

CONCLUSION

The Portable APILL system operates using a wireless control system connected via Bluetooth, utilizing an Arduino Uno R3 as the core controller. The intersection cycle time calculations, performed through an application-based system, were successfully implemented, demonstrating its potential as an effective learning tool for competencies related to road equipment management. The research process was conducted systematically, involving several stages: theoretical review, concept development, data collection at the studied intersection, analysis, data validation testing, and drawing conclusions. The findings of this study indicate that the cycle time calculations performed using Excel and the application, based on PKJI 2023 guidelines, were synchronized and accurately visualized through the lamp operations of the Portable APILL system. However, during the testing of the lamp operation timings, discrepancies were observed, particularly in the green light duration for the northern approach, where the lamp was observed to operate for 7.1 seconds compared to 6 seconds calculated using Excel and 7 seconds by the application. These differences were primarily caused by rounding processes in the calculations and inaccuracies during the manual observation of the lamp operation using a stopwatch. This highlights the need for precision in both computational processes and observation techniques to ensure the reliability of such systems in educational and practical applications.

REFERENCES

Adeleke, O. O., Oguntayo, D., Ayobami, I. T., & Ogunkunbi, G. A. (2023). Improving Traffic Flow in Emerging Cities: A SIDRA Intersection Based Traffic Signal Design. *Periodica Polytechnica Transportation Engineering*, *51*(2), 166–171. https://doi.org/10.3311/pptr.18018

Ahmad, Z., Khan, A. S., Nisar, K., Haider, I., Hassan, R., Haque, M. R., Tarmizi, S., & Rodrigues, J. J. P. C. (2021). Anomaly Detection Using Deep Neural Network for IoT Architecture. *Applied Sciences*, *11*(15), 7050. https://doi.org/10.3390/app11157050

Baz, A., Yi, P., & Qurashi, A. (2020). Intersection Control and Delay Optimization for Autonomous Vehicles Flows Only as Well as Mixed Flows With Ordinary Vehicles. *Vehicles*, *2*(3), 523–541. https://doi.org/10.3390/vehicles2030029

Chen, H. (2020). Battery electric vehicle eco-cooperative adaptive cruise control in the vicinity of signalized intersections. *Energies*, *13*(10). https://doi.org/10.3390/en13102433

Dinh, H., & Tang, H. (2017). Development of a Tracking-Based System for Automated Traffic Data Collection for Roundabouts. *Journal of Modern Transportation*, *25*(1), 12–23. https://doi.org/10.1007/s40534-017-0124-z

Direktorat Jenderal Bina Marga. (2023). *Pedoman Kapasitas Jalan Indonesia*.

Gade, D. S. (2019). ICT Based Smart Traffic Management System "iSMART" for Smart Cities. *International Journal of Recent Technology and Engineering*, *8*(3), 3920–3928. https://doi.org/10.35940/ijrte.c5137.098319

Ge, H., Song, Y.-C., Wu, C., Ren, J., & Tan, G. (2019). Cooperative Deep Q-Learning With Q-Value Transfer for Multi-Intersection Signal Control. *Ieee Access*, *7*, 40797–40809. https://doi.org/10.1109/access.2019.2907618

Gu, J., Fang, Y., Sheng, Z., & Wen, P. (2020). Double Deep Q-Network With a Dual-Agent for Traffic Signal Control. *Applied Sciences*, *10*(5), 1622. https://doi.org/10.3390/app10051622

Guo, F. (2023). Design of Intelligent Monitoring and Identification System for Abnormal Behavior. *2023 IEEE 3rd International Conference on Information Technology, Big Data and Artificial Intelligence (ICIBA)*, *3*, 1450–1454. https://doi.org/10.1109/ICIBA56860.2023.10165259

Huang, T., Poddar, S., Aguilar, C., Sharma, A., Smaglik, E. J., Kothuri, S., & Koonce, P. (2018). Building Intelligence in Automated Traffic Signal Performance Measures With Advanced Data Analytics. *Transportation Research Record Journal of the Transportation Research Board*, *2672*(18), 154–166. https://doi.org/10.1177/0361198118791380

Islam, S. M. A. B. Al, Tajalli, M., Mohebifard, R., & Hajbabaie, A. (2021). Effects of Connectivity and Traffic Observability on an Adaptive Traffic Signal Control System. *Transportation Research Record Journal of the Transportation Research Board*, *2675*(10), 800–814. https://doi.org/10.1177/03611981211013036

Jamil, A. R. M., & Nower, N. (2021). A Comprehensive Analysis of Reward Function for Adaptive Traffic Signal Control. *Knowledge Engineering and Data Science*, *4*(2), 85. https://doi.org/10.17977/um018v4i22021p85-96

Jin, J., & Ma, X. (2019). A Non-Parametric Bayesian Framework for Traffic-State Estimation at Signalized Intersections. *Information Sciences*, *498*, 21–40. https://doi.org/10.1016/j.ins.2019.05.032

Kamasetty, B., Renduchintala, M., Shetty, L. L., Chandarshekar, S., & Shettar, R. (2022). Design and Development of Portable Smart Traffic Signaling System With Cloud-Artificial Intelligence Enablement. *Indonesian Journal of Electrical Engineering and Computer Science*, *26*(1), 116. https://doi.org/10.11591/ijeecs.v26.i1.pp116-126

Kim, M., Schrader, M., Yoon, H., & Bittle, J. A. (2023). Optimal Traffic Signal Control Using Priority Metric Based on Real-Time Measured Traffic Information. *Sustainability*, *15*(9), 7637. https://doi.org/10.3390/su15097637

Liu, J., Zhang, X., Li, Y., Wang, J., & Kim, H.-J. (2019). Deep Learning-Based Reasoning With Multi-Ontology for IoT Applications. *Ieee Access*, *7*, 124688–124701. https://doi.org/10.1109/access.2019.2937353

Ma, Z., Huang, D., Li, C., & Guo, J. (2020). Travel Time Reliability-Based Signal Timing Optimization for Urban Road Traffic Network Control. *Mathematical Problems in Engineering*, *2020*, 1–11. https://doi.org/10.1155/2020/8898062

Neetu, G., Rajan, S., & Nitin, G. (2023). Dynamic Traffic Signal Control in Smart Cities Using Operations Research. *International Journal of Convergence in Healthcare*, *3*(2). https://doi.org/10.55487/jecex109

Saeed, W. (2024). Geometric Features and Traffic Dynamic Analysis on 4-Leg Intersections. *International Review of Applied Sciences and Engineering*, *15*(2), 171–188. https://doi.org/10.1556/1848.2023.00681

Saldivar-Carranza, E., Li, H., Platte, T., & Bullock, D. M. (2023). Systemwide Identification of Signal Retiming Opportunities With Connected Vehicle Data to Reduce Split Failures. *Transportation Research Record Journal of the Transportation Research Board*, *2677*(12), 587–603. https://doi.org/10.1177/03611981231168844

Silas, A. I., Jonathan, F. M., & Ekong, O. E. (2021). Traffic Management System for Prioritizing Emergency Over Time and Density. *Advances in Multidisciplinary & Scientific Research Journal Publication*, *29*, 23–36. https://doi.org/10.22624/aims/abmic2021-v2-p3

Win, S., Win, W. Y., & Theingi. (2018). Area Based Traffic Control System. *International Journal of Research and Studies Publishing*, *8*(12). https://doi.org/10.29322/ijsrp.8.12.2018.p8488

Yan, L. (2024). Deep Reinforcement Learning for Ecological and Distributed Urban Traffic Signal Control With Multi-Agent Equilibrium Decision Making. *Electronics*, *13*(10), 1910. https://doi.org/10.3390/electronics13101910.