Available *online* at : http://chipset.fti.unand.ac.id/

Two-Wheeled Vehicle Access Control System Using QR Code Based on Optical Character Recognition

| ISSN (Online) 2722-4422 |

Artificial Intelligence and Embedded System

Two-Wheeled Vehicle Access Control System Using QR Code Based on Optical Character Recognition

Muhammad Hadrian Johanna[1], *Dody Ichwana Putra**[2],

[1,2]*Departemen Teknik Komputer, Universitas Andalas, Limau Manis Kec. Pauh, Kota Padang, Sumatera Barat 25163 Indonesia*

ARTICLE INFORMATION

Received: August 29, 2025
Revised: October 8, 2025
Available online: October 31, 2025

KEYWORDS

Number Plate, Raspberry Pi, Detection Camera, Optical Character Recognition

CORRESPONDENCE

E-mail: dody.ichwana@it.unand.ac.id

A B S T R A C T

The increasing number of motorcycles owned by students at Universitas Andalas has led to inefficiencies in the manual parking card system, including frequent card loss and reduced operational effectiveness. To address this issue, a motorcycle access control system is designed using QR codes integrated with Optical Character Recognition (OCR). The system utilizes a Raspberry Pi 4 Model B, Logitech C270 HD webcam, and a barcode scanner to capture motorcycle license plate images, process them with Convolutional Neural Network (CNN) and Tesseract OCR to extract plate text, and generate a unique QR code for each vehicle. The QR code is used for both entry and exit validation, replacing the need for manual parking cards. Data is stored and managed in a MySQL database, enabling quick retrieval and verification. System testing showed high accuracy in license plate detection and QR validation, ensuring security, efficiency, and ease of use while reducing operational costs. This solution offers a scalable approach for campus, residential, and office access control systems.

INTRODUCTION

The increase in student enrollment at a university has led to a rise in the number of vehicle users within the campus environment. To ensure vehicle security, each student receives a parking card as proof of entry through the gate [1]. However, several issues often arise, one of which is the loss of parking cards. Students who lose their parking cards are not allowed to exit the gate without presenting them, resulting in penalties in the form of fines. Moreover, the ineffectiveness of parking cards is another problem that contributes to frequent losses around the campus parking area.

Common problems faced by students include forgetting where they placed their parking cards or losing them due to accidental drops. This indicates that the loss of parking cards mainly occurs due to student negligence. Consequently, the number of parking cards no longer matches the number of vehicles on campus. In fact, according to an interview conducted by *Genta Andalas*, the number of parking cards in 2023 reached 4,000, while student vehicles exceeded 5,000, leading to shortages, losses, and damages of the cards [2].

Previous studies have proposed various approaches to address similar issues. Radio Frequency Identification (RFID)-based parking systems have been proven to improve security and parking management efficiency, as demonstrated in research that integrated RFID with MySQL for motorcycle parking management systems [3], and its implementation at IPB University, which successfully reduced queues and improved vehicle management [4].

In addition to RFID, Automatic Number Plate Recognition (ANPR) technology is also widely used. ANPR utilizes Optical Character Recognition (OCR) to automatically read vehicle license plates and has been applied in various intelligent transportation systems [5]. Methods based on Convolutional Neural Networks (CNN) further enhance the accuracy of ANPR, such as the use of YOLO, which enables real-time detection with an accuracy of up to 93.5% [6]. Furthermore, recent studies show that integrating QR Code with CNN-based ANPR can improve the efficiency and security of parking systems, since QR Codes can serve as a unique vehicle identifier for the validation process [7].

Based on this review, this study proposes the design of a two-wheeled vehicle access control system using QR Code and OCR, implemented on a Raspberry Pi. The system is expected to

https://doi.org/10.25077/chipset.6.02.160-166.2025

overcome the weaknesses of manual parking card systems by providing an automatic validation process that is more efficient, secure, and easy to operate.

METHOD

System Design

Figure 1. It presents a general overview of the system design, showing how the camera, monitor, and barcode scanner are configured to minimize potential conflicts within the designed system. The barcode scanner is equipped with two LEDs, which function as validation indicators for the QR Code obtained by students through the monitor, displayed in the form of a web interface.

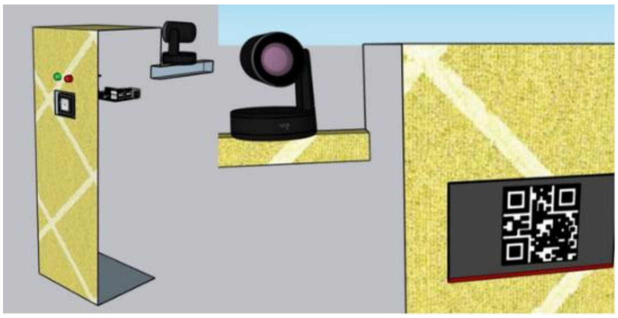

Figure 1. General system design

Hardware Design

The schematic of this hardware design connects all components to a Raspberry Pi 4. Two GPIO pins GPIO 17 and GPIO 27 are used to drive the green and red indicator LEDs, respectively. Other components, including an embedded 1D/2D barcode scanner (QR barcode reader module) and a Logitech C270 camera, are connected via USB. The monitor is connected via HDMI.

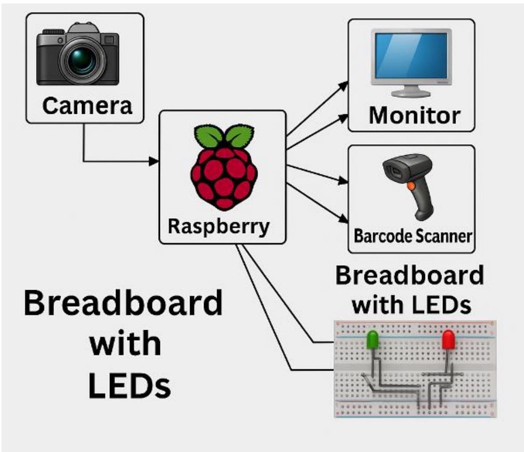

Figure 2. Schematic of the hardware design

Software Design

The system workflow begins with the camera capturing real-time images of vehicle license plates. These images are processed using the YOLO algorithm to detect the plate position, after which OCR is applied to extract the character text. The recognized text is then converted into a QR Code and stored in the database as the vehicle's digital identity. During entry or exit, the QR Code is verified using a scanner to ensure data consistency with the Firebase database, and the system automatically makes the access decision.

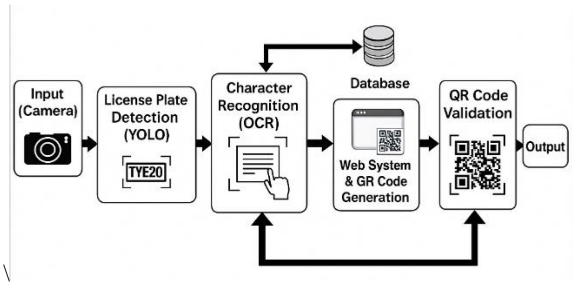

Figure 3. Schematic of the software design

Optical Character Recognition

The license plate image processing using EasyOCR begins with capturing an image in *.jpg* format, followed by preprocessing with Median Blur to reduce noise and improve text accuracy. Next, segmentation is performed to separate each character, followed by normalization to standardize the dimensions and thickness of the letters. The subsequent stage is feature extraction to obtain the characteristics of each character, and then recognition is carried out by matching the extracted features with the database. The final result is an ASCII text output of the vehicle's license plate number.

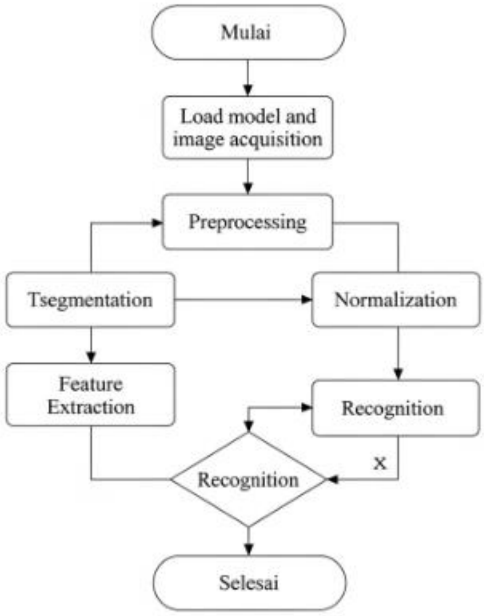

Figure 4. Schematic of license plate processing using OCR

YOLO (You Only Look Once)

You Only Look Once (YOLO) is an object detection algorithm based on convolutional neural networks (CNN), designed to operate in real time by scanning each frame only once. The algorithm divides an image into grids, for example 13×13, to estimate the position and class of objects using bounding boxes

https://doi.org/10.25077/chipset.6.02.160-166.2025

and class probabilities. In its implementation, YOLOv4-Tiny is used because it is more effective in processing colored (RGB) images without converting them to grayscale, while the data labeling process follows the YOLO format, which includes bounding box coordinates and object classes, enabling more optimal model training.

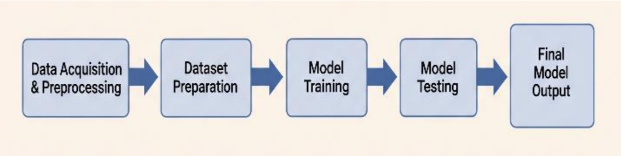

Figure 5. Schematic of YOLO processing

Dataset Preparation

The license plate dataset was collected from various sources, including video recordings using a webcam in the parking area and along the roads of Andalas University, which were converted into *.jpg* images, mobile phone camera captures, as well as additional datasets from Kaggle [Online]. Initially, the dataset consisted of only 350 images. However, since deep learning performance generally improves with larger datasets, an augmentation process was carried out to expand both the size and diversity of the dataset. The augmentation techniques applied included rotation, brightness adjustment, blurring, noise addition, and image skewing, resulting in additional variations. This augmentation process not only increased the dataset size but also enhanced the model's ability to generalize under real-world conditions, thereby contributing to improved accuracy in object recognition.

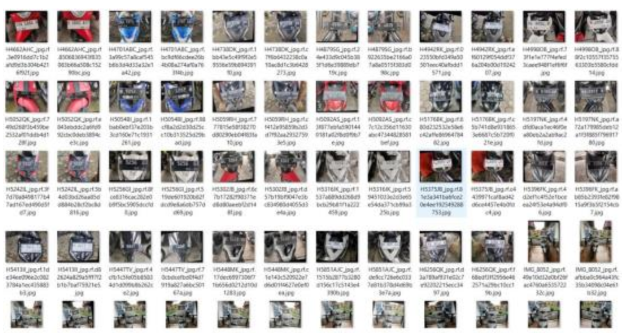

Figure 6. Implementation of dataset collection

Model Training

The YOLOv4-tiny model training was carried out using the Darknet framework on Google Colab with GPU acceleration to optimize computational performance. The labeled dataset was uploaded to Google Drive and processed with parameter configurations including an image size of 416×416, a batch size of 32, a learning rate of 0.00261, and 6,000 iterations. The dataset was divided into 90% training data (1,050 images) and 10% testing data (105 images). The YOLOv4-tiny architecture consists of convolutional layers, route layers, max pooling, upsampling, and two main detection layers that generate bounding box predictions, confidence scores, and class probabilities. The training results achieved a testing accuracy of 99.83% with a high mean Average Precision (mAP), balanced precision, recall, and F1-Score, as well as a final average loss of 0.0441. Among the generated weights, the model "yolov4-tiny-custom_best.weights" was selected as the best-performing model for implementing automatic license plate detection.

RESULT AND ANALYSIS

System Design Result

This system design integrates a Logitech C270 webcam as the image acquisition device, a Raspberry Pi 4 Model B as the processing unit, and a Barcode Scanner Module for QR Code validation. All components are assembled into a single compact device, illustrated through external, side, and internal views, enabling the system to optimally support license plate detection and vehicle access verification functions.

Figure 7. Implementation of the design system

YOLO and OCR

The results show that the implemented YOLO and OCR modules, utilizing the EasyOCR library, were able to recognize license plate text with an accuracy level above 90% on high-resolution images. In addition, the Logitech C270 webcam used as the image acquisition device successfully captured license plate images with sufficient quality at 720p resolution and 30 fps, producing stable captures under normal lighting conditions. Furthermore, the images obtained from this camera also supported the bounding box generation process in the YOLO algorithm, enabling more accurate and efficient license plate area detection.

Figure 8. License plate detection model results

Firebase

The detection results are sent to Firebase, where the recognized license plate data is stored in the *plat_masuk* collection with a unique document containing the plate number and timestamp, while the *qrcode_kendaraan* collection stores the unique code generated from the QR Code. When a vehicle exits, the scanned QR Code is verified against the database records, allowing the validation process to run quickly and synchronously.

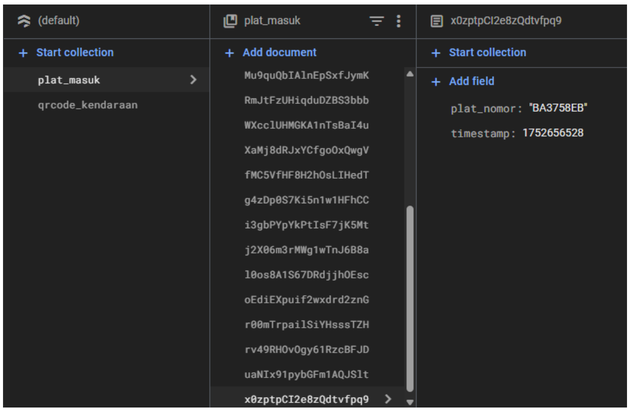

Figure 9. Display of plat_masuk collection on firebase firestore

Website

The results for the website interface, with data retrieval from Firebase, can be implemented by displaying the QR Code generated from the detected license plate.

Figure 10. Implementation of qr code on website

Barcoder Scanner

This Barcode Scanner is capable of reading QR Codes from various positions and angles. The component functions as an identification system for vehicle exit access, where the scanned result must match the data stored in the database.

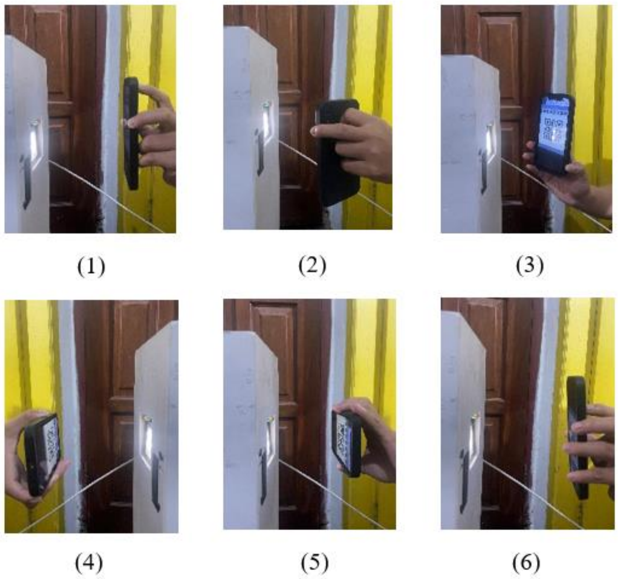

Figure 11. Barcoder scanner testing

TESTING

Testing of YOLO and OCR Detection

The testing was conducted by measuring the response time of YOLO in detecting license plates and generating the license plate text into a QR Code. This experiment was carried out using five motorcycles with different license plates.

https://doi.org/10.25077/chipset.6.02.160-166.2025

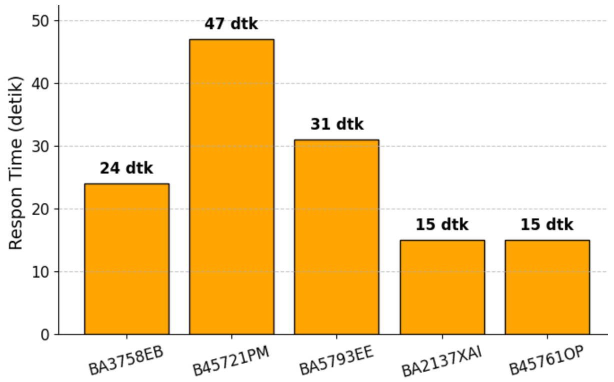

Figure 12. Comparison graph of response time in license plate detection

The testing results of the license plate detection system response time showed performance variations ranging from 15 to 47 seconds. The 4th and 5th trials produced the fastest response time of 15 seconds, while the 1st and 3rd trials fell into the medium category with 24 and 31 seconds, respectively. The 2nd trial recorded the longest response time of 47 seconds. Despite the significant differences in processing duration, the system consistently demonstrated a high detection accuracy level within the range of 90–100%.

Website and Database Testing

The testing was carried out by displaying the website under two conditions: when no vehicle was present and when a vehicle was detected. When a vehicle was detected, the system recorded new data into the database, displaying the QR Code, entry time, and license plate text. The data was shown on the website for 15 seconds, with new data retrieval occurring in approximately 1 second. Database testing using Firebase demonstrated that both the data retrieval process for the website and the data deletion after QR Code verification functioned as expected.

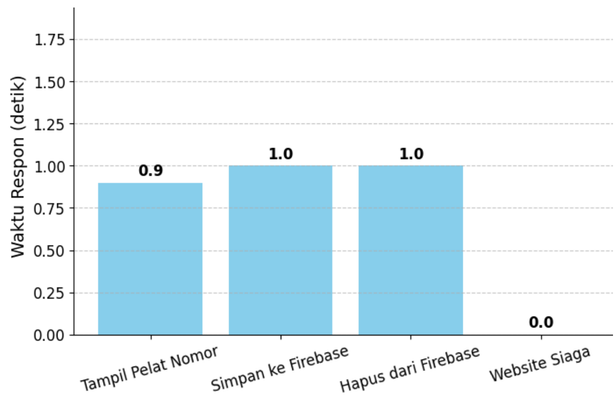

Figure 13. Respons time testing graph of website to database

The test results indicate that the system operates in real time with a response time of less than 1 second for data input, storage, and deletion. The website display requires approximately 15 seconds, which is still acceptable, while QR Code validation in Firebase runs quickly. Overall, the system performed stably as planned, with adequate response time for image-based vehicle monitoring and online database integration.

QR Code Validation Testing with Database

System testing focused on matching the vehicle QR Code data with the database, where each scanned QR Code was compared with the data recorded during vehicle entry. From four trials, all were successfully verified with a response time of approximately 1 second, indicated by the green LED, thereby demonstrating that the database integration and QR Code processing functioned optimally.

Figure 14. Validation testing of qr code on firebase

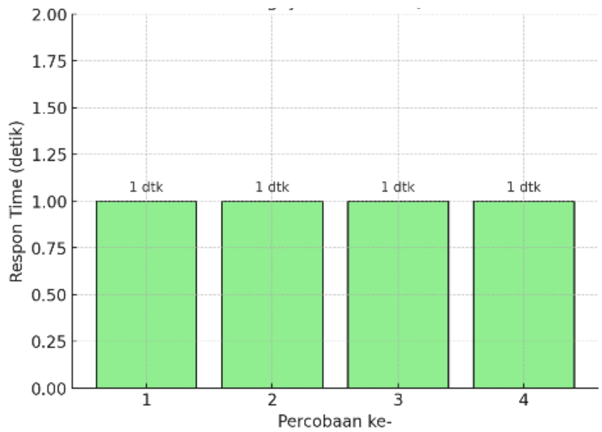

Figure 15. Validation testing graph of qr code on firebase

Testing of Overall System Functionallity

The system testing was carried out in five trials following the workflow of license plate detection, QR Code generation, and exit gate verification. From the user's perspective, the process provided a simple and clear experience, where license plates were automatically detected with an accuracy of 90–100%, QR Codes were always successfully verified with a 100% success rate, and the green LED indicator provided instant feedback. Although the response time varied between 15–47 seconds, it remained acceptable. Overall, the system was evaluated as practical, real-time, and sufficiently reliable in supporting two-wheeled vehicle entry and exit.

Figure 16. System Functionality Testing Flow

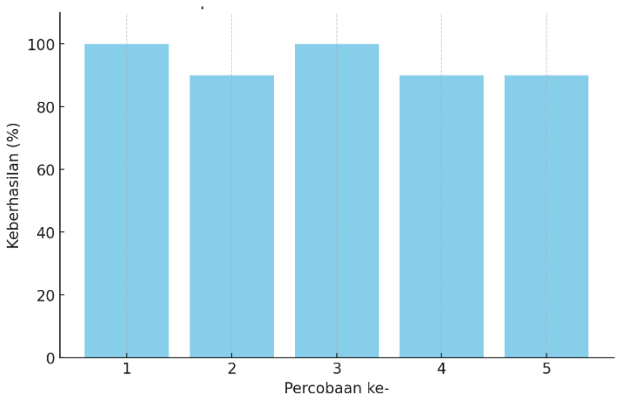

Figure 17. Graph of user experience success rate percentage

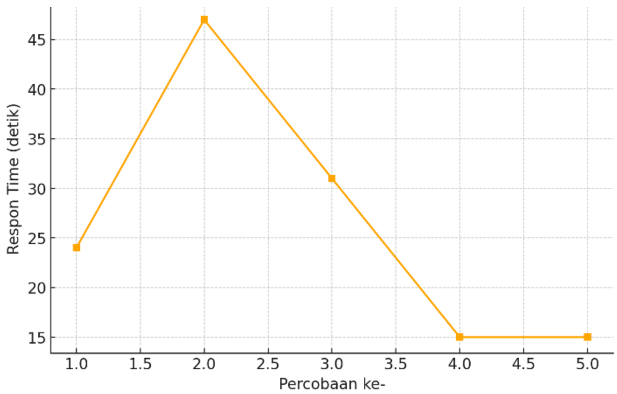

Figure 18. User response time graph in license plate detection

Discussions

The developed system demonstrated fairly good performance but still has certain limitations. Camera sensitivity is the main challenge, as license plate detection accuracy decreases under low-light conditions or when the capture angle is not optimal. In addition, reliance on a cloud database (Firebase) makes the system vulnerable to network disruptions, which may affect the QR Code validation and data storage processes. The test results also indicated errors under low-light and rainy conditions, highlighting the need for hardware improvements and algorithm enhancements to ensure greater system reliability in diverse environmental conditions.

CONCLUSIONS

The system was successfully implemented using a Raspberry Pi 4 Model B, a Logitech C270 camera, a barcode scanner, and LED indicators, with testing results showing performance parameters above 90% accuracy. The barcode scanner was able to read QR Codes accurately even when positioned at various angles (0°, ±45°, ±90°, up to 180°), while the YOLOv4-Tiny model demonstrated consistent performance in detecting license plates at close to medium distances (≤100 cm) under lighting conditions of ≥179 lux, although its accuracy decreased at distances greater than 150 cm or lighting below 46 lux. Integration with the website interface and Firebase also worked smoothly, where detection results were displayed on the website within 0.9 seconds, stored in the database within 1 second, and automatically deleted after successful validation within the same timeframe. Nevertheless, the system can still be improved by using devices with more powerful GPUs to enable YOLOv4-Tiny or newer models to perform real-time license plate detection with higher frame rates, as well as employing higher-quality cameras to improve accuracy under low-light conditions or glare. Furthermore, the system could be extended to a mobile application, allowing users to more easily store and utilize QR Codes digitally.

ACKNOWLEDGEMENT

Thanks to chatGPT for help this article formatting picture.

REFERENCES

[1] D. I. Putra, R. Aisuwarya, S. Ardopa, and I. Purnama, "Sistem Cerdas Reservasi dan Pemantauan Parkir pada Lokasi Kampus Berbasis Konsep Internet of Things," Jurnal Teknologi dan Sistem Komputer, vol. 6, no. 2, pp. 57–63, Mar. 2018, doi: 10.14710/jtsiskom.6.2.2018.57-63.

[2] T. Juwita and H. Rahmi, "Sistem Kartu Parkir UNAND Sengaja Ditiadakan," Genta Andalas. Accessed: Mar. 28, 2024. [Online]. Available: https://www.gentaandalas.com/sistem-kartu-parkir-unand-sengaja-ditiadakan/

[3] A. Syafrudin, R. Alfian, N. Fitriyani, and J. Rhee, "A motorcycle parking lot management system based on RFID," ResearchGate, 2014. [Online]. Available: https://www.researchgate.net/publication/269304019_A_motorcycle_parking_lot_management_system_based_on_RFID

[4] I. M. Saputra, A. F. Amrulloh, and A. F. Hakim, "Implementation of Smart Parking System with RFID

 https://doi.org/10.25077/chipset.6.02.160-166.2025

Technology in IPB Vocational School Campus," ResearchGate, 2023. [Online]. Available: https://www.researchgate.net/publication/386747278

[5] "Automatic number-plate recognition," Wikipedia, 2024. [Online]. Available: https://en.wikipedia.org/wiki/Automatic_number-plate_recognition

[6] W. Li, H. Zhao, J. Liu, and S. Zhang, "A Robust Real-Time Automatic License Plate Recognition Based on the YOLO Detector," arXiv preprint arXiv:1802.09567, 2018. [Online]. Available: https://arxiv.org/abs/1802.09567

[7] H. R. Ramadhan, M. M. Rosyidi, and M. F. Isnain, "Enhancing Parking Systems with QR Code-Integrated Automatic License Plate Recognition through Convolutional Neural Networks," Akademia Baru, 2024. [Online]. Available: https://www.akademiabaru.com/submit/index.php/ard/article/download/6397/6119

AUTHOR(S) BIOGRAPHY

Muhammad Hadrian Johanna
Born in Bukittinggi, Kota Bukittinggi on January 19, 2003, attended SMA N 1 Bukittinggi and is currently pursuing a degree in Computer Engineering at the Faculty of Information Technology, Andalas University