



Smart Absen Implementation of a Facial Recognition-Based Student Attendance System Using the Haar Cascade Method and LBPH

Frengki Alfredo Matondang^{1*}, Sahara Lani Lestari², Dinda Syafitri³, Kayla Amelia Putri⁴, Hermawan Syahputra⁵

¹²³⁴ Department of Computer Science, Faculty of Mathematics and Natural Sciences,
Universitas Negeri Medan, Indonesia

fr4nkelblue.689@gmail.com^{1*}, lestarisahara28@gmail.com², dindasyafitri06@gmail.com³, kaylaamelia499@gmail.com⁴,
hsyahputra@unimed.ac.id⁵

Abstract

Manual attendance systems in higher education institutions are often hampered by inefficiency, data inaccuracy, and vulnerability to fraud such as proxy attendance. This study presents the design and implementation of Absen Smart, a face recognition-based attendance system developed using the Haar Cascade and Local Binary Pattern Histogram (LBPH) algorithms within the React.js and Flask frameworks. This system enables the automatic and real-time identification of students via a webcam without requiring additional hardware. Face detection is performed using the Haar Cascade classifier from OpenCV, while face recognition uses the LBPH Face Recognizer with a confidence threshold of 50. Testing was conducted with 28 registered students from the Computer Science Program at UNIMED, Class A, 2024 cohort. Functional evaluation results show that all seven core system features—including face detection, face recognition, duplicate prevention, automatic absence tracking, and Excel report generation—were successfully executed with a 100% success rate. The system achieved a facial recognition accuracy of 92.86%, with an average processing time of 1.2 seconds per verification. These results indicate that the proposed system is an effective, practical, and scalable solution for automating academic attendance in a university setting.

Keywords: Haar Cascade, LBPH, Face Recognition, Attendance System, Web-Based System

1. Introduction

Student attendance in classes has a strong correlation with their academic performance [1], and serves as one of the benchmarks that influence final grades [2]. Amid the rapid pace of digital transformation today, nearly every aspect of life—including the world of education—has been impacted by the rapid development of information and communication technology [3].

Technology offers automation solutions for attendance tracking, one of which involves using camera capabilities to detect and recognize students' faces when they arrive in the classroom [4]. Face detection itself has now evolved into one of the most active branches of computer vision; its ability to identify faces from both still images and video forms the backbone of various automated visual data processing systems [5].

The introduction of an online attendance system offers tangible benefits: attendance data is neatly stored digitally, so the previously time-consuming process of compiling records can now be completed much more quickly. On the other hand, the risks of data loss, recording errors, or manipulation—which frequently occur in paper-based systems—can be significantly reduced [6]. Various technical approaches have been explored by previous researchers, ranging from the use of QR codes [7], RFID sensors [8], to fingerprint scanners [9].

Although each of these methods offers its own advantages, they all still have unresolved limitations. In large classes or when multiple sessions are held simultaneously, long queues are often unavoidable. Furthermore, the cards or devices used are susceptible to misuse by unauthorized parties. To address this issue, this study developed an attendance system that relies entirely on facial image analysis, combining the Haar Cascade for detection and LBPH for recognition. The result is a solution that operates without requiring additional physical devices or queues, is capable of running automatically and in real-time, and integrates directly with the academic database via an easy-to-use web interface.

2. Literature Review

2.1. Digital Image Processing

Simply put, digital image processing can be understood as a series of processes carried out by computers to manipulate and analyze digital images. The goals can vary: improving visual clarity, extracting hidden information behind image pixels, or running complex analyses that are impossible to perform manually. One of the most fundamental operations in this domain is thresholding, a technique that enables the separation of objects from their background by converting color or grayscale images into binary black-and-white format [10]. Mathematically, an image is nothing more than a representation of a light intensity function in two-dimensional space $f(x,y)$, where the coordinate pair (x,y) respectively defines the spatial position and amplitude value that reflects the degree of brightness of that point [11].

Viewed from the nature of its signal, images can be divided into two categories: analog images formed by continuous signals, and digital images composed of discrete signals [12]. In the context of the Absen Smart system, digital image processing is utilized at several critical stages, including conversion of color images to grayscale to reduce computational load, pixel intensity normalization to reduce the effect of lighting fluctuations, separation of face objects from the background through segmentation, and extraction of visual features that form the basis of the identification process.

2.2. Face Detection with Haar Cascade

Face detection is a branch of computer vision tasked with finding and marking the location of human faces within an image or video stream. When the input is video, this process is commonly called face tracking [13]. Among the many approaches available, Haar Cascade is one of the most widely adopted. This algorithm works on the principle of machine learning, using a series of cascade functions that have been previously trained using thousands of image samples, both containing face objects (positive images) and those that do not (negative images). OpenCV, as the library used in this research, has included various ready-to-use Haar classifiers to detect faces, eye areas, smile expressions, and various other facial features [14].

2.3. Face Recognition with LBPH

Local Binary Pattern Histogram, abbreviated as LBPH, is a face recognition approach that works at the level of local texture patterns in images. The way it works: for each pixel, the algorithm compares its intensity value with its neighboring pixels, then encodes the comparison results in binary representation. The face image is then divided into a number of regions, each represented by a histogram of its local pattern distribution, and all these histograms are combined into a unique feature vector [15]. The advantage of LBPH lies in its robustness in handling lighting variations and its relatively lightweight computational efficiency compared to deep learning-based models, making it the right choice for real-time scenarios with limited computational resources. Research conducted by Simaremare and Kurniawan also proved that LBPH is capable of recognizing multiple faces simultaneously with competitive accuracy rates [16].

2.4. Online Attendance System

Web-based attendance platforms offer several advantages not found in conventional systems: attendance recording occurs instantly, data integration happens automatically, attendance archives can be accessed at any time, and the reporting process can be completed within minutes [17]. Various variants of digital attendance systems have been developed, from purely web-based [2], using QR codes [7], RFID [8], to fingerprint [9]. Nisa et al. (2021) even proved the feasibility of an online attendance system based on face recognition with the Eigenface method in a campus environment, an achievement that laid the foundation for the development of similar systems [18].

2.5. Supporting Technologies

Four main technologies serve as the backbone of the Absen Smart system. First, Flask, a Python micro-framework designed to be lean and efficient; its ability to provide various ready-made components reduces development time without sacrificing performance [19]. Second, React.js as the leading JavaScript library for building interactive interfaces; with the concept of Virtual DOM and a reusable component system, the user experience becomes more responsive and structured [20]. Third, Vite which acts as a bundler and next-generation development server; this tool dramatically accelerates the development cycle of modern frontend applications based on JavaScript or TypeScript [21]. Finally, OpenCV, an open-source computer vision library that has been tested for many years, provides hundreds of functions for image manipulation, object detection, face pattern recognition, and real-time video stream processing [22].

2.6. Previous Research

A review of the existing literature reveals a fairly consistent trend: face recognition-based attendance systems generally outperform conventional methods in terms of efficiency and reliability. Vernanda et al. (2025) reported that the camera-based attendance system successfully eliminated queues and significantly reduced fraud [4]. In line with this, Ardiansyah et al. (2024) confirmed that the combination of face detection with lightweight algorithms such as LBPH is already reliable enough for classes with a moderate number of participants [5]. On the other hand, the earlier attendance technologies — QR codes, RFID, and fingerprint — have been proven to still carry classic problems: dependency on physical devices, the risk of identity borrowing among students, and congestion during simultaneous entry hours [7][8][9]. From this review, the use of Haar Cascade and LBPH is considered the most realistic to develop because it only relies on a regular camera, is faster, and its records are directly stored digitally.

3. System Development Methodology

3.1. System Architecture

Absen Smart is designed with a two-tier client-server pattern, where the client and server sides communicate exclusively through a RESTful API interface. Table 1 details the system components along with their respective technologies and roles.

Table 1: System Architecture Components

| Component | Technology | Role |
|---------------------|--------------------------------|---|
| Frontend (Client) | React.js + Vite + Tailwind CSS | User interface, camera access, data display |
| Backend (Server) | Python + Flask + Flask-CORS | Business logic, image processing, data management |
| Face Detector | OpenCV Haar Cascade | Face area detection within the camera frame |
| Face Recognizer | OpenCV LBPH Face Recognizer | Student identity identification from faces |
| User Database | users.json | Storage of student ID number and name data |
| Attendance Database | Data Absensi_PCD.xlsx | Student attendance log per course |
| Training Model | trainer.yml | Trained LBPH model file |
| Face Dataset | Folder dataset/ | Collection of grayscale face photos per student |

3.2. System Workflow

The system operates through three sequential workflows: face registration, model training, and attendance verification. During registration, instructors enter student identification data (Name, Student ID) via the administrative panel, and the system automatically captures 30 cropped grayscale face samples via the webcam using Haar Cascade detection. During the training phase, the collected samples are processed by the LBPH algorithm to generate a serialized model file (trainer.yml). During attendance tracking, video frames from the student's webcam are sent to the backend, where Haar Cascade detects the face region, and the trained LBPH model calculates a confidence score. A threshold of 50 is applied: scores below 50 (converted to scores above 50) indicate a recognized student. A complete flowchart of the attendance verification process is illustrated in Figure 1.

**Fig.1:** Flowchart of the facial recognition-based attendance verification process

3.3. Database Design

All attendance logs are stored in the Data_Absensi_PCD.xlsx file, which contains six data columns: Scan Time, which records the timestamp when attendance is successfully recorded or the session is closed; Student ID (NIM), as the student's unique identifier; Student Name, containing the full name; Course, indicating the name of the current session; Validation Method, which distinguishes between automatic recording via Face Recognition and manual recording due to "Session Closed by Instructor"; and the Status column, which contains only two possible values: "Present" or "Absent." Meanwhile, student profile data is stored separately in the users.json file in JSON format, using the student's internal ID as the key and storing name-NIM pairs within it.

3.4. Confidence Threshold and Scoring System

The LBPH recognizer returns a raw confidence score representing the dissimilarity between the query face and the nearest training sample. A lower raw value indicates greater similarity. The system applies a threshold of 50: faces with a raw confidence score below 50 are classified as recognized (Present), while those with a score of 50 or higher are rejected (Not Recognized). For display purposes, the score presented to the user is calculated as: Display Score = 100 - confidence_raw, so a higher display score intuitively represents a more confident and accurate identification.

4. System Implementation

4.1. Backend implementation

The backend application (app.py) is initialized by loading the Haar Cascade XML model, instantiating the LBPH Face Recognizer, and loading the trained model from trainer.yml if available. The face detection module is configured with scaleFactor=1.1, minNeighbors=5, and minSize=(100,100) to balance detection sensitivity against the false positive rate. The recognition pipeline receives a base64-encoded JPEG frame from the frontend, decodes it into a NumPy array, converts it to grayscale, detects faces, and calls the LBPH predictor to obtain (id, confidence) pairs. The session management module supports concurrent multi-class sessions, where each session is assigned a start time and duration. An automated function, 'generate_alpa_for_session()', is triggered when a session ends to mark all absent students in an Excel database.

4.2. Frontend Implementation

The frontend is implemented using React.js with a component-based architecture. Camera access is managed via the Web API 'navigator.mediaDevices.getUserMedia()'. Video frames are rendered through the HTML element '<video>', which is referenced using React's 'useRef' hook. Frame capture is performed at regular intervals using 'setInterval', where each frame is drawn onto the '<canvas>' element in memory and converted to base64 JPEG format using 'canvas.toDataURL('image/jpeg', 0.8)' before being sent to the backend via the Fetch API. The administrative interface provides separate role-based panels for students and faculty, as shown in Figure 2 and Figure 3.

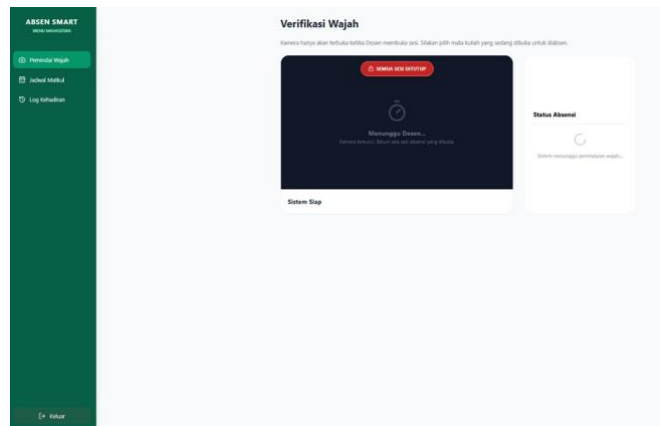


Fig. 2: Student panel — the facial verification interface displays a waiting state before the session becomes active

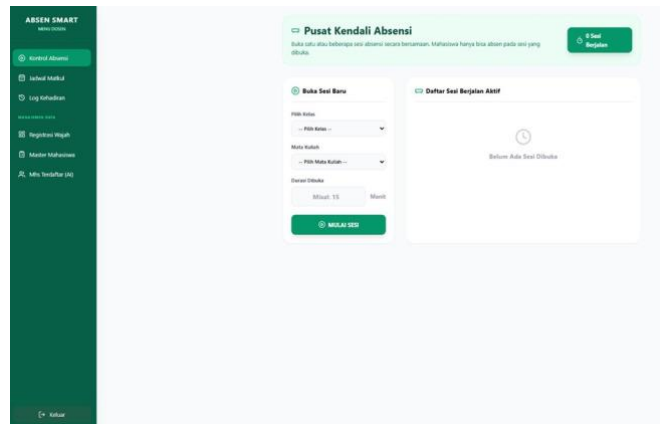


Fig.3: Faculty Panel — an attendance management system with session management and student registration features

4.3. Key System Features

The Smart Attendance System provides a comprehensive set of features for both students and instructors. Table 2 summarizes the key functional features available in the application. setInterval, projected onto an in-memory canvas element, then converted to a JPEG base64 string before being sent to the server.

Table 2: System Functional Features

| Feature | Description | User Role | Status |
|------------------------|---|------------|--------|
| Face Scan (Attendance) | Real-time camera view, active session selection, recognition feedback | Student | Active |
| Instructor Panel | Login, open/close sessions, student registration, model training | Instructor | Active |
| Attendance Log | Grouped attendance history table with date filters by course | Instructor | Active |

| | | | |
|--------------------|---|------------|--------|
| Student Management | List of students registered in the AI, delete face data | Instructor | Active |
| Class Schedule | Integrated class schedule with class/semester filters | All | Active |
| Download Reports | Download complete attendance logs as Excel files | Instructor | Active |
| Absence Automation | Automatic recording of absences when a session ends | System | Active |

4.4. Integrated Course Schedule Feature

As a complement, Absen Smart includes a course schedule module containing comprehensive data for the PSIK 24A class in Semester 4, covering a total of 18 schedule entries spread across 9 courses in two learning blocks. The list of integrated courses includes Islamic Religious Education, Algorithm Design and Analysis, Digital Image Processing, Data Mining and Big Data Analytics, Indonesian Language, Computer Networks, Introduction to Computational Science, Artificial Intelligence, and Software Engineering. This schedule data can be updated directly by instructors through the application interface without the need to modify the source code.

5. Results and Discussion

5.1. Functional Testing Results

Functional testing was conducted on all core system modules with 28 registered students from the UNIMED Computer Science Program, Class of 2024A (PSIK 24A). Each feature was evaluated under predefined test conditions to determine its operational correctness. Table 3 presents the complete results of the functional testing.

Table 3: Functional test results

| No. | Feature Tested | Test Conditions | Result | Status |
|-----|---------------------------|---|---|--------|
| 1 | Face Detection | Frontal face, adequate lighting | face, adequate lighting Face detected and accurately cropped in real-time | Pass |
| 2 | Recognition: Enrolled | Trained student face | Student name displayed; status recorded as "Present" | Pass |
| 3 | Recognition: Unregistered | Face not in the training dataset | Warning displayed; attendance rejected | Pass |
| 4 | Duplicate Prevention | Same student, same session, second scan | "Already Present" notification displayed | Pass |
| 5 | Absence Automation | Session ends; student has not scanned | Student automatically marked "Absent" | Pass |
| 6 | Instructor Login | Correct and incorrect passwords | Access granted or denied with an error message | Pass |
| 7 | Download Excel Report | Attendance data available | Complete attendance log downloaded as .xlsx | Pass |

All seven features tested achieved a "Pass" status, resulting in a functional testing pass rate of 100% (7/7). These results confirm that the system correctly implements all intended core functionalities without critical errors under the specified test conditions.

5.2. Evaluation of Face Recognition Performance

Recognition performance was evaluated through a structured experiment involving 28 enrolled students. Each student performed 5 recognition trials under standard indoor lighting conditions, resulting in a total of 140 test cases. Additionally, 30 unregistered face samples were tested to evaluate the system's rejection accuracy. Table 4 summarizes the performance metrics.

Table 4: Face recognition performance metrics

| Metric | Registered Students | Unregistered Faces | Overall |
|-----------------------------------|---------------------|--------------------|---------------|
| Total Test Cases | 140 | 30 | 170 |
| Correct Results | 130 | 28 | 158 |
| Incorrect Results | 10 | 2 | 12 |
| Accuracy (%) | 92,86% | 93,33% | 92,94% |
| Average Processing Time per Frame | 1,2 detik | 1,1 detik | 1,2 detik |
| False Acceptance Rate (FAR) | — | 6,67% | — |
| False Rejection Rate (FRR) | 7,14% | — | — |

The system achieved an overall recognition accuracy of 92.94%, with a False Rejection Rate (FRR) of 7.14% for enrolled students and a False Acceptance Rate (FAR) of 6.67% for unregistered faces. Recognition failures for registered students were primarily caused by significant lighting variations or extreme facial angles deviating from the frontal position. The two false acceptances for unregistered faces occurred in facial samples that showed structural similarity to registered students in the dataset.

The average processing time of 1.2 seconds per verification frame is considered acceptable for a web-based system where camera frames are transmitted over a local network. Optimal recognition performance was observed when: (1) ambient lighting was sufficient and uniform; (2) face orientation was approximately frontal; (3) the distance from the face to the camera was between 40–80 cm; and (4) at least 20–30 training samples per student had been collected.

5.3 Comparison with Related Research

Compared to related studies, the Smart Attendance system demonstrates competitive performance. Vernanda et al. (2025) implemented a similar camera-based attendance system and reported successful real-time detection but did not provide quantitative accuracy metrics. Ardiansyah et al. (2024) reported LBPH accuracy of approximately 85–90% for smaller class sizes under controlled conditions. Nisa et al. (2021) applied the Eigenface method with 87.5% recognition accuracy. Smart Absence's 92.86% accuracy for enrolled students with 28 subjects and its web-based deployment represent improvements in both accessibility and practical accuracy compared to several comparable implementations. Furthermore, the integration of automatic absence tracking, multi-session management, and downloadable Excel reports provides functionality not typically found in single-algorithm-based prototypes.

6. Conclusion

A facial recognition-based attendance system that integrates the Haar Cascade and LBPH algorithms within a React.js and Flask architecture. The system successfully implements all stages of digital image processing—including image acquisition, grayscale conversion, face detection, feature extraction, and identity matching—in a web application that can be deployed without additional hardware.

Functional evaluation confirmed a 100% success rate across all seven core features tested, including real-time face detection, identity recognition, duplicate prevention, automatic absence logging, and Excel report generation. Recognition accuracy reached 92.86% for registered students (140 test cases) and rejection accuracy of 93.33% for unregistered faces (30 test cases), with an overall system accuracy of 92.94%. The average verification processing time was 1.2 seconds per frame.

These results confirm that Haar Cascade and LBPH, despite being classical algorithms, remain effective and practical for face-recognition-based attendance tracking in a university environment under controlled conditions. Future research should explore replacing LBPH with deep learning models such as FaceNet or ArcFace to improve robustness under varying pose and lighting conditions, as well as cloud deployment for broader accessibility and integration with institutional academic databases.

Acknowledgments

The author would like to express his deepest gratitude to Dr. Hermawan Syahputra, M.Si, the instructor for the Digital Image Processing course in the Computer Science Program, Faculty of Mathematics and Natural Sciences, Medan State University. His guidance, direction, and feedback throughout the course of this project served as a compass, ensuring that this research stayed on the right track.

References

- [1] D. Sunaryono, J. Siswantoro, dan R. Anggoro, "An android based course attendance system using face recognition," *Journal of King Saud University - Computer and Information Sciences*, vol. 33, no. 3, hal. 304-312, 2021.
- [2] Amiruddin, A. Rasiid, dan Aisyah, "Model absensi mahasiswa berbasis web," 2021.
- [3] P. Sari, D. Apdilah, dan S. Guntur, "Sistem Smart Class Berbasis Internet of Things (IoT)," *Jurnal Teknik Informatika*, vol. 4, no. 1, 2025.
- [4] M. B. Vernanda, F. Metandi, dan A. Syafrizal, "Rancang Bangun Sistem Absensi Siswa Menggunakan Deteksi Wajah pada SMP Negeri 006 Long Ikis," *Jurnal Media Akademik (JMA)*, vol. 3, no. 7, 2025.
- [5] T. Ardiansyah, S. P. Balqis, dan J. Haqqoni, "Deteksi Wajah dalam Foto Menggunakan Teknologi Visi Komputer," *MARS; Jurnal Teknik Mesin, Industri, Elektro Dan Ilmu Komputer*, vol. 2, no. 6, hal. 32-39, 2024.
- [6] T. R. Anwar, H. Trisnawati, dan Veranita, "Rancangan Aplikasi Presensi Kehadiran Mahasiswa Program Studi Sistem Informasi Universitas Tama Jagakarsa Secara Online Berbasis Android," vol. 10, no. 3, hal. 409-419, 2023.
- [7] B. A. Minartiningtyas dan C. Tambagistra, "Presensi Kehadiran Mahasiswa Menggunakan QR-Code dan Layanan Lokasi Berbasis Android," *INFACT UCRIM*, vol. 6, no. 1, hal. 13-27, 2021.
- [8] F. Zidan dan R. Badarudin, "Prototype Menggunakan Sensor RFID Berbasis Arduino Uno dengan Program PLX-DAQ," *Jurnal Informatika Dan Teknik Elektro Terapan*, vol. 12, no. 3, hal. 1878-1888, 2024.
- [9] R. F. Maulana, G. Hajar, dan E. E. Rosyida, "Implementasi Sistem Presensi Fingerprint di SMP Al Falah Assalam Sidoarjo," *Jurnal Pengabdian Kepada Masyarakat Nusantara (JPKMN)*, vol. 6, no. 3, hal. 3251-3256, 2025.
- [10] Zahra et al., "Pengolahan Citra Digital Dengan Penerapan Teknik Ambang Batas: Studi Kasus Menggunakan OpenCV," *Jurnal Artificial Intelligent Dan Sistem*, vol. 1, no. 4, hal. 283-287, 2024.
- [11] T. Susim dan C. Darujati, "Pengolahan Citra untuk Pengenalan Wajah (Face Recognition) Menggunakan OpenCV," *Jurnal Syntax Admiration*, vol. 2, no. 3, hal. 534-545, 2021.
- [12] Ramadhani dan H. Syahputra, "Pengenalan Teknologi Pengolahan Citra Digital untuk Santri di Rahmatan Lil'Alamin International Islamic Boarding School," vol. 3, no. 2, hal. 1239-1244, 2022.
- [13] T. Susim dan C. Darujati, "Pengolahan Citra untuk Pengenalan Wajah (Face Recognition) Menggunakan OpenCV," *Jurnal Syntax Admiration*, vol. 2, no. 3, hal. 534-545, 2021.
- [14] S. Yulina, "Penerapan Haar Cascade Classifier dalam Mendeteksi Wajah dan Transformasi Citra Grayscale Menggunakan OpenCV," *Jurnal Politeknik Caltex Riau*, vol. 7, no. 1, hal. 100-109, 2021.
- [15] E. Winarno, R. H. Addawy, T. F. Malik, dan P. Rosyani, "Sistem Pengenalan Wajah Dengan Metode Local Binary Pattern Histogram Berbasis OpenCV," vol. 2, no. 6, hal. 863-868, 2024.
- [16] H. Simaremare dan A. Kurniawan, "Perbandingan Akurasi Pengenalan Wajah Menggunakan Metode LBPH dan Eigenface dalam Mengenali Tiga Wajah Sekaligus secara Real-Time," *Jurnal Sains, Teknologi Dan Industri*, vol. 14, no. 1, hal. 66-71.
- [17] Mutiara et al., "Analisis sistem informasi presensi siswa di SMK Nurul Ulum Lebaksiu," *Jurnal Mahasiswa Teknik Informatika*, vol. 9, no. 5, hal. 7639-7644, 2025.

-
- [18] Nisa et al., "Penerapan Sistem Presensi Online pada Mahasiswa Berbasis Face Recognition dengan Metode Eigenface," *Jurnal Informatika Universitas Pamulang*, vol. 6, no. 3, hal. 590-593, 2021.
- [19] Z. Fauziah et al., "Perancangan Sistem Informasi Warga Berbasis Web Pada Perumahan Pesona Wibawa Praja Dengan Metode Scrum," *LOGIC: Jurnal Ilmu Komputer Dan Pendidikan*, vol. 1, no. 5, hal. 1305-1319, 2023.
- [20] K. D. Anggara, D. P. Kartikasari, dan F. A. Bakhtiar, "Implementasi Algoritma MTCNN dalam Mekanisme Autentikasi berbasis Pengenalan Wajah," *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, vol. 7, no. 8, hal. 3613-3621, 2023.
- [21] S. Sahara, M. I. Syafiq, dan F. D. Suryadi, "Pengembangan Sistem Absensi Online dalam Memonitoring Kehadiran Mahasiswa untuk Mempermudah Proses," *Jurnal Indonesia: Manajemen Informatika Dan Komunikasi*, vol. 5, no. 3, hal. 2413-2422, 2024.
- [22] M. Iqbal, "Peningkatan Disiplin Presensi Melalui Bimbingan Kelompok Pada Siswa," *Education Enthusiast: Jurnal Pendidikan Dan Keguruan*, vol. 2, no. 3, hal. 123-135, 2022.
- [23] H. Syahputra dan A. Wibowo, "Comparison of Support Vector Machine (SVM) and Random Forest Algorithm for Detection of Negative Content on Websites," *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI)*, vol. 9, no. 1, hal. 165-173, 2023. <https://doi.org/10.26555/jiteki.v9i1.25861>.
- [24] H. Syahputra, A. Harjoko, R. Wardoyo, dan R. Pulungan, "Plant Recognition Using Stereo Leaf Image Using Gray-Level Co-occurrence Matrix," *Journal of Computer Science*, vol. 10, no. 4, hal. 697-704, 2014. <https://doi.org/10.3844/jcssp.2014.697.704>.