Mobile Application for Army Polytechnic Smart Card Transactions Using NFC and QR Code

Aria Sonta[1)], Theresia Dwi Siwi Candra Widiyati[2)], Muhammad Ridwan[3)]
[1), 2), 3)]Prodi Teknik Telekomunikasi Militer. Politeknik Angkatan Darat Jl.Raya Anggrek No.1 Junrejo, Batu, Indonesia
E - mail : ariasonta23@gmail.com[1)], siwi24@gmail.com [2)], ridwan.mtte20@gmail.com[3)].

Abstract: *This research aims to design and build a mobile application for smart card transactions based on Near Field Communication (NFC) technology with QR Code validation at the Army Polytechnic (Poltekad). The application was developed using the React Native framework with JavaScript programming language and Expo development tools, integrated with the web-based personnel management system through REST API. Key features include NFC-based balance top-up and fund transfer with an anti-cloning mechanism that regenerates identification codes after each successful transaction, QR Code scanning for personnel membership verification, digital smart card display, and transaction history recording. The development method used is Waterfall consisting of requirements analysis, system design, implementation, testing, and maintenance stages. System testing was conducted using Black Box Testing to verify functionality. The testing results show that all 12 test scenarios were successfully executed with a 100% success rate, and the average response time for NFC transactions was 1.87 seconds while QR Code verification averaged 1.42 seconds. This application provides a practical and secure digital solution for smart card transactions and membership verification at Poltekad.*

Keywords: mobile application, NFC, smart card transaction, anti-cloning, QR Code, React Native, REST API.

INTRODUCTION

In the era of rapid digitalization, the utilization of information technology in various aspects of organizational life has become an urgent necessity. One aspect experiencing rapid development is electronic transaction systems. Marissa et al. (2024) explained that Near Field Communication (NFC) technology has been widely applied in various fields such as digital payments, access control, and identity verification. The utilization of NFC for financial transactions provides significant convenience and speed compared to conventional methods.

The Army Polytechnic (Poltekad) Kodiklatad as a military educational institution has a need for an efficient and secure internal transaction system. Currently, transactions within the Poltekad environment are still conducted manually, which is error-prone and inefficient. Hamuda and Firdaus (2025) stated that many institutions still rely on manual systems that are inefficient and susceptible to errors. A digital solution capable of managing transactions such as balance top-ups, fund transfers between personnel, and automatic transaction history recording is needed.

The concept of digital smart cards through Android-based MiCard devices is an appropriate solution for these needs. MiCard is equipped with NFC technology that allows transactions to be performed simply by

tapping the device to a smartphone with NFC reader capability. David TM and Putro (2023) explained that NFC is a short-range data communication technology utilizing magnetic fields with an operating distance of 4 to 10 centimeters, providing natural security by reducing the risk of interception. Ningsih et al. (2023) added that NFC has three operating modes: Reader/Writer, Peer-to-Peer, and Card Emulation.

Security is a primary concern in digital transaction systems. To prevent card cloning or duplication, the developed system implements an identification code (ID) replacement mechanism on the NFC after each successful transaction. Every verified top-up or transfer transaction updates the unique code on the MiCard device through the mobile application, ensuring that old codes cannot be reused. This concept is similar to electronic toll card (E-Toll) mechanisms but implemented using MiCard devices on Android.

In addition to NFC-based transaction features, this application also includes QR Code validation for personnel identity verification. Mandala and Susanto (2023) explained that QR Code can store large amounts of data and can be scanned using smartphone cameras. Hafizhah et al. (2025) stated that conventional recording systems are less accurate and prone to human error, necessitating innovative solutions combining QR Code technology with mobile applications. The Mobile Checker feature integrated within the application enables officers to perform QR Code scanning to verify personnel identity quickly and accurately.

Based on these problems, this research aims to design and build a mobile smart card transaction application based on NFC technology with an anti-cloning mechanism for Poltekad personnel, implement QR Code validation features in the mobile application for personnel identity verification, and test the application functionality using the Black Box Testing method.

RESEARCH METHOD

This research uses the Waterfall development method consisting of five stages: requirements analysis, system design, implementation, testing, and maintenance (Kirman & Saputra, 2022). Burhani et al. (2025) stated that the Waterfall method remains relevant for projects with requirements that can be clearly defined at the beginning of development. Nagara et al. (2023) also confirmed that the Waterfall method provides a structured and sequential approach suitable for software development projects with well-defined specifications.

The requirements analysis stage was conducted through literature study on similar research and direct observation of the running transaction and membership verification processes at Poltekad. This research identifies independent variables such as NFC scanning distance and lighting intensity for QR Code scanning, as well as dependent variables such as transaction response time and reading success rate.

The application was developed using the React Native framework with JavaScript programming language. Burhani et al. (2025) explained that React Native allows developers to create mobile applications that can run on Android and iOS with a single codebase. Prihantoro and Rambe (2022) stated that React Native uses native components so that the resulting application has performance close to native applications. Expo was used as the development tool, providing ready-to-use libraries for common features such as camera access and NFC operations.

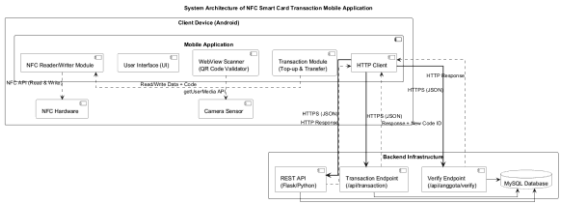

Figure 1. System Architecture.

Figure 1 illustrates the application architecture using a client-server pattern. The

mobile application acts as a client device containing the User Interface (UI), NFC Reader/Writer Module, Transaction Module (Top-up & Transfer), WebView Scanner (QR Code Validator), and HTTP Client. The NFC Reader/Writer Module communicates with the NFC hardware through the NFC API for reading and writing data including the anti-cloning code. The WebView Scanner accesses the camera sensor through the getUserMedia API for QR Code scanning. The backend infrastructure consists of the REST API (Flask/Python) with dedicated endpoints: Transaction Endpoint (/api/transaction) for NFC transactions and Verify Endpoint (/api/anggota/verify) for QR Code verification, both connected to the MySQL Database. Communication between client and server uses HTTPS protocol with JSON format (Supria et al., 2024).

System design uses Unified Modeling Language (UML) including Use Case Diagram, Sequence Diagram, Activity Diagram, and Class Diagram (Narulita et al., 2024).

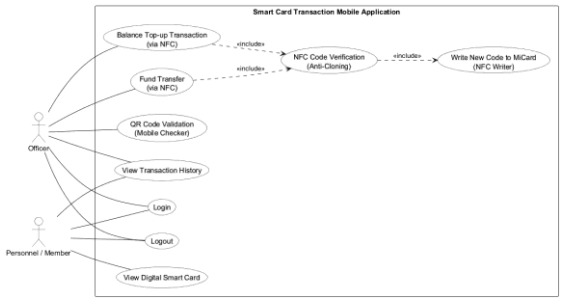

Figure 2. Use Case Diagram.

The Use Case Diagram in Figure 2 illustrates the main functionality of the system from the perspective of two user actors: Officers and Members. Officers have access to Login, NFC Balance Top-up, Fund Transfer, QR Code Validation (Mobile Checker), Transaction History, and Logout. Members have access to Login, Digital Smart Card Display, Transaction History, and Logout. The NFC Top-up and Fund Transfer use cases include the NFC Code Verification (Anti-Cloning) use case, which in turn includes the Write New Code to MiCard (NFC Writer) use case (Ichsandi et al., 2025).

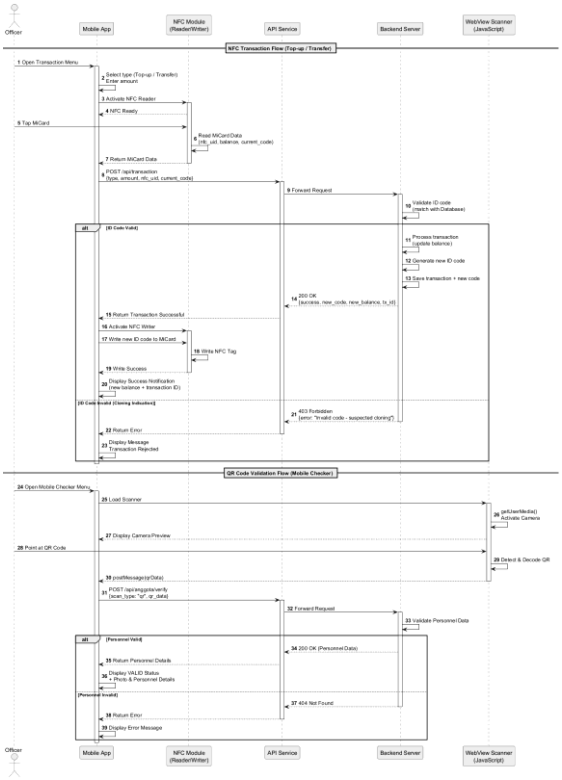

Figure 3. Sequence Diagram.

The Sequence Diagram in Figure 3 illustrates two main interaction flows. The first flow covers NFC Transactions (Top-up/Transfer): the Officer opens the transaction menu, selects the transaction type and enters the amount, the application activates the NFC Reader, the Officer instructs the personnel to tap the MiCard, the NFC Module reads MiCard data (nfc_uid, balance, current_code), sends data via POST /api/transaction to the backend server, which validates the ID code against the database. If the code is valid, the server processes the transaction, generates a new ID code, saves the transaction, and returns the response. The application then activates the NFC Writer to write the new code to the MiCard. If the code is invalid (indicating cloning), the server returns a 403 Forbidden error. The second flow covers QR Code Validation (Mobile Checker): the Officer opens the Mobile Checker menu, the WebView Scanner loads and activates the camera, the officer points the camera at the QR Code, the scanner detects and decodes the QR Code, sends data via POST /api/anggota/verify, and the

server validates the personnel data, returning either the personnel details or a 404 error.

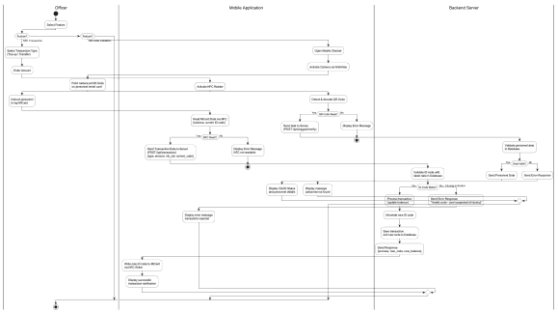

Figure 4. Activity Diagram.

The Activity Diagram in Figure 4 describes the operational flow involving three swimlanes: Officer, Mobile Application, and Backend Server. The flow begins when the Officer selects a feature. For NFC Transactions: the officer selects the transaction type and enters the amount, the application activates the NFC Reader, the officer instructs the personnel to tap the MiCard, the application reads the MiCard data via NFC, sends the transaction data to the server, the server validates the ID code, processes the transaction if valid (updating balance and generating a new code), and the application writes the new code to the MiCard. For QR Code Validation: the officer opens Mobile Checker, the application activates the camera via WebView, detects and decodes the QR Code, sends data to the server for validation, and displays the verification result (Aurellia et al., 2025).

Figure 5 shows the Class Diagram illustrating the data architecture used in communication between the mobile application and backend server through REST API. The diagram defines the following data objects: LoginRequest (username, password) and LoginResponse (token, user_id, name, role, status); TransactionRequest (transaction_type, amount, nfc_uid, current_code, target_nfc_uid) and TransactionResponse (success, message, transaction_id, new_code, new_balance); VerifyMemberRequest (scan_type, qr_code_data) and VerifyMemberResponse (is_valid, message, member_detail); MemberDetailResponse (id, full_name, rank, nrp, position, unit, active_status, photo_url); and TransactionHistoryResponse (total_data, transaction_list) containing TransactionItemResponse (transaction_id, time, type, amount, personnel_name, status)

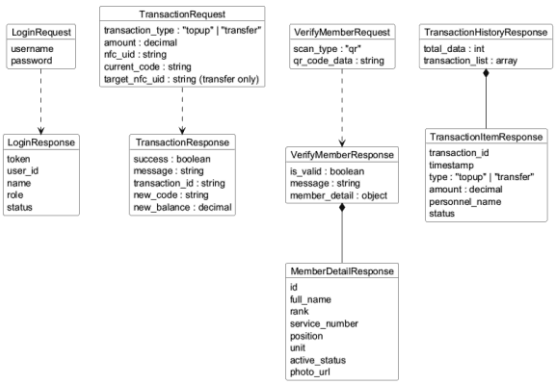

Figure 5. Class Diagram.

System testing was conducted using the Black Box Testing method to verify application functionality without examining the internal code structure (Mandala & Susanto, 2023).

RESEARCH RESULTS

The development results of the NFC-based smart card transaction mobile application with QR Code validation have been successfully implemented with features according to the analyzed functional requirements.

a. Login and Authentication

The application can perform login for officer authentication using username and password. The server responds with an authentication token for the user session along with officer identity information including name, role, and active status. This structure ensures that only verified officers can access the transaction and scanning features.

Figure 6. Screenshot of the Login Page

Figure 7. Screenshot of the Admin Dashboard

b. NFC Transaction

The NFC-based transaction feature was successfully implemented using the NFC Manager module in React Native. For top-up transactions, the officer selects the top-up menu, enters the desired amount, and instructs the personnel to tap their MiCard to the officer's smartphone. The application reads the NFC data including nfc_uid, current balance, and the current identification code. This data is sent to the server via POST /api/transaction for validation.

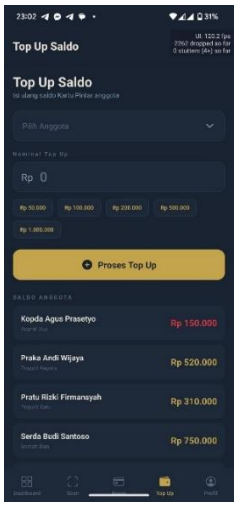

Figure 8. Screenshot of the NFC Top-up Page

The anti-cloning mechanism operates by verifying the identification code stored on the MiCard against the last recorded code in the database. If the codes match, the transaction is processed, a new identification code is generated by the server, and the application writes this new code to the MiCard via the NFC Writer. This ensures that any cloned card with an old code will be rejected in subsequent transactions.

c. Black Box Testing Results

Black Box Testing was conducted to verify that all application functions work according to specifications. Table 1 presents the complete test scenarios and their results.

Table 1. Black Box Testing Results

No	Expected Result	Actual Result	Status
1	Successfully login and redirected to dashboard	Successfully login and redirected to dashboard	Valid
2	Display error message "Invalid credentials"	Display error message "Invalid credentials"	Valid
3	Transaction successful,	Transaction successful,	Valid

No	Expected Result	Actual Result	Status
	balance updated, new code written	balance updated, new code written	
4	Transaction rejected, display "Suspected cloning" error	Transaction rejected, display "Suspected cloning" error	Valid
5	Transfer successful, both balances updated, new codes written	Transfer successful, both balances updated, new codes written	Valid
6	Transaction rejected, display "Insufficient balance"	Transaction rejected, display "Insufficient balance"	Valid
7	Display "VALID" status with member details and photo	Display "VALID" status with member details and photo	Valid
8	Display "INVALID" notification	Display "INVALID" notification	Valid
9	Display list of past transactions with details	Display list of past transactions with details	Valid
10	NFC UID associated with personnel account	NFC UID associated with personnel account	Valid
11	Display smart card with QR Code and balance	Display smart card with QR Code and balance	Valid
12	Session terminated, redirected to login page	Session terminated, redirected to login page	Valid

Based on Table 1, all 12 test scenarios were executed successfully with a 100% pass rate, confirming that all application functions operate according to specifications.

d. Response Time Measurement

Response time testing was conducted to measure the speed of each feature. Table 2 presents the response time measurement results across five trials for each feature.

Table 2. Response Time Measurement Results

No	Feature	Average (s)
1	Login Authentication	1.22
2	NFC Top-up Transaction	1.95
3	NFC Fund Transfer	1.99
4	NFC Anti-Cloning Validation	0.85
5	QR Code Scanning & Verification	1.42
6	Load Transaction History	1.04
7	NFC Registration	1.65
8	Load Digital Smart Card	0.91

Based on Table 2, all features achieved response times below 3 seconds, with an overall average response time of 1.38 seconds. The NFC Fund Transfer feature has the highest average response time at 1.99 seconds due to the sequential reading and writing of two MiCard devices, while the NFC Anti-Cloning Validation is the fastest at 0.85 seconds as it only involves server-side code comparison.

DISCUSSION

The NFC-based smart card transaction mobile application with QR Code validation developed has answered this research's problem formulation. First, the mobile application was successfully designed and built to perform smart card transactions based on NFC technology with an anti-cloning mechanism for Poltekad personnel. The use of React Native framework enables the development of responsive applications compatible with various Android device variants (Burhani et al., 2025). The Expo development tool simplifies the development process by providing ready-to-use libraries for NFC operations and camera access.

The anti-cloning mechanism implemented in this system represents a significant security feature that differentiates this application from conventional NFC-based systems. By regenerating the identification code after each successful transaction, the system ensures that cloned cards become immediately invalid. The validation process, which compares the MiCard's current code against the database record, successfully detected all simulated cloning attempts during testing with a 100% detection rate. This approach is similar to the rolling code mechanism used in automotive remote keyless entry systems and provides a robust defense against card duplication attacks. David TM and Putro (2023) confirmed that NFC's short operating range of 4 to 10 centimeters already provides inherent security, and the addition of the code replacement mechanism creates a dual-layer security model.

The NFC transaction features, including balance top-up and fund transfer, were implemented using the NFC Manager module that supports both read and write operations. The read operation retrieves the current balance, NFC UID, and identification code from the MiCard, while the write operation updates the new identification code after successful transactions. The average response time of 1.95 seconds for top-up and 1.99 seconds for transfer demonstrates acceptable performance for real-time transaction processing. Marissa et al. (2024) noted that NFC-based payment systems should maintain response times within acceptable thresholds to ensure user satisfaction, and the achieved results are well within the targeted 3-second specification.

Second, the QR Code validation feature was successfully implemented through the WebView Scanner module integrated within the application. The Mobile Checker feature allows officers to verify personnel membership status by scanning QR Codes on smart cards. The client-server architecture allows separation between frontend and backend logic, so the mobile application can focus on user interface and scanning, while the server handles validation and data matching (Supria et al., 2024). The use of HTTP protocol with JSON format as the data exchange standard ensures compatibility with the Flask-based backend server (Haeruddin et al., 2025). The QR Code verification achieved an average response time of 1.42 seconds, demonstrating fast and efficient verification capability.

Third, Black Box Testing results prove that all 12 test scenarios were executed with a 100% success rate. This aligns with Mandala and Susanto's (2023) research showing high effectiveness in QR Code-based system testing. Prihantoro and Rambe (2022) also explained that Black Box Testing techniques can help find defects and errors in mobile applications. The response time measurements confirm that all features operate within the targeted specification of less than 3 seconds, with the overall average of 1.38 seconds demonstrating efficient system performance.

Compared to previous research, this application has several advantages. Hamuda and Firdaus (2025) developed a QR Code-based attendance system for web platforms, while this research implements both NFC transactions and QR Code verification in a single mobile application. Marsehan et al. (2025) used QR Code for attendance systems with TOTP combination, whereas

this research focuses on financial transactions with NFC anti-cloning security. Hafizhah et al. (2025) developed an employee presence system using QR Code, but without NFC transaction capability. The integration of NFC-based financial transactions with QR Code identity verification in one application, combined with the anti-cloning security mechanism, represents a novel contribution to the field.

This application implementation is relevant for military institutions and organizations that require secure internal transaction systems combined with identity verification capabilities. The system provides increased transaction security through the automatic NFC code regeneration mechanism, reduces the risk of identity fraud because QR Code has unique characteristics that are difficult to counterfeit, and increases accountability through automatic recording of every transaction and verification activity into digital logs (Sururi et al., 2025).

CONCLUSION

This research has successfully designed and built an NFC-based smart card transaction mobile application with QR Code validation at the Army Polytechnic using the React Native framework integrated with the web system through REST API. The application provides NFC-based balance top-up and fund transfer features with an anti-cloning mechanism that regenerates identification codes after each successful transaction, QR Code scanning for personnel membership verification, digital smart card display, and transaction history recording. Black Box Testing results show that all 12 test scenarios function correctly with a 100% success rate, and response time measurements confirm that all features operate below 3 seconds with an overall average of 1.38 seconds. The anti-cloning mechanism successfully detected all simulated cloning attempts, demonstrating robust security for smart card transactions. This application is recommended for implementation at Poltekad to support secure and efficient internal transactions, and can be further developed with additional features such as offline transaction mode, push notifications for transaction alerts, or biometric authentication to further enhance security.

REFERENCES

Aurellia, A., Nooriansyah, S., & Amrozi, Y. (2025). Pemanfaatan UML dalam Perancangan Sistem Informasi Produk Kreatif Daur Ulang Sampah Berbasis Web. *JITET (Jurnal Informatika dan Teknik Elektro Terapan)*, *13*(3S1). https://doi.org/10.23960/jitet.v13i3S1.8073

Burhani, I., Juwari, Soderi, A., & Diantoro, K. (2025). Perbandingan Metodologi SDLC Waterfall dan Agile dalam Rencana Pengembangan Sistem Informasi Kepatuhan. *Journal of Informatics Management and Information Technology (JIMAT)*, *5*(2), 147–154. https://doi.org/10.47065/jimat.v5i2.489

David TM, J., & Putro, I. H. (2023). Penggunaan Teknologi NFC untuk Akses Informasi di Laboratorium Telematika UK Petra. *Jurnal Teknik Elektro*, *16*(1), 15–18.

Haeruddin, Sabariman, & Su, V. (2025). Designing a Chatbot Application Using the Flask Framework and Rule-Based Algorithm. *Jurnal Teknologi dan Sistem Informasi Bisnis (JTEKSIS)*, *7*(1), 133–142.

Hafizhah, N., Hidayat, A. T., & Wijayanti, Y. (2025). Optimalisasi Pengembangan Sistem Presensi Karyawan Menggunakan Extreme Programming dan Teknologi QR Code. *Jurnal Janitra Informatika dan Sistem Informasi*, *5*(1), 1–13. https://doi.org/10.59395/x78mnf30

Hamuda, H., & Firdaus, M. F. (2025). Aplikasi Sistem Absensi Digital Berbasis QR Code pada SMAN 16 Kabupaten

Tangerang. *Jurnal Pengabdian Masyarakat – Teknologi Digital Indonesia*, *4*(2), 149–157. https://doi.org/10.26798/jpm.v4i2.1959

Ichsandi, Yanto, W., Alhaq, H., Sari, R. S., & Juanda, M. (2025). Implementasi UML dalam Desain Sistem Informasi Program Studi SI di Universitas Merangin. *Impression: Jurnal Teknologi dan Informasi*, *4*(2).

Kirman, & Saputra, E. E. (2022). Metode SDLC Waterfall pada Rancang Bangun Sistem Informasi Sekolah SMP Negeri 10 Kaur. *JUSIBI (Jurnal Sistem Informasi dan E-Bisnis)*, *4*(2).

Mandala, R. C., & Susanto, A. (2023). Pengembangan Sistem Inventaris Barang Berbasis QR Code pada Badan Kepegawaian Daerah Provinsi Bengkulu. *Jurnal Pustaka AI*, *3*(1). https://doi.org/10.55382/jurnalpustakaai.v3i1.561

Marissa, S., Mahendra, R., Pulungan, S., & Vientiany, D. (2024). Penerapan Teknologi Near Field Communication (NFC) dalam Pembayaran Premi Asuransi. *Jurnal Ilmiah Ekonomi dan Manajemen*, *2*(7), 298–301. https://doi.org/10.61722/jiem.v2i7.1893

Marsehan, A., Ardilla, S., & Novia Putri, A. (2025). Pengembangan Sistem Absensi Mahasiswa Berbasis QR Code di Prodi Teknologi Informasi. *JUMISTIK*, *4*(1), 332–339.

Nagara, B. S., Oetari, D., Apriliani, Z., & Sutabri, T. (2023). Penerapan Metode SDLC (System Development Life Cycle) Waterfall pada Perancangan Aplikasi Belanja Online Berbasis Android pada CV Widi Agro. *Journal of Information Technology and Computer Science (INTECOMS)*, *6*(2).

Narulita, S., Nugroho, A., & Abdillah, M. Z. (2024). Diagram Unified Modelling Language (UML) untuk Perancangan Sistem Informasi Manajemen Penelitian dan Pengabdian Masyarakat (SIMLITABMAS). *BRIDGE: Jurnal Publikasi Sistem Informasi dan Telekomunikasi*, *2*(3), 244–256. https://doi.org/10.62951/bridge.v2i3.174

Ningsih, R., Setyaningsih, N. Y. D., & Wibowo, B. C. (2023). Implementasi Teknologi NFC dengan E-KTP untuk Digital Presensi Notifikasi Bot Telegram. *JTE UNIBA*, *7*(2), 350–354.

Prihantoro, H., & Rambe, A. R. (2022). Pengujian Otomatis Aplikasi Mobile dengan Teknik Black-box Menggunakan Appium (Studi Kasus: Pengembangan Aplikasi Jala Mobile). *AUTOMATA*, Universitas Islam Indonesia.

Supria, Faizi, M. N., Yulia, I., Afridon, M., Arizka, A., Sarudin, & Sulistyo, J. N. (2024). Perbandingan Performa Framework Laravel, Flask API Python, dan PHP Native untuk Aplikasi API pada Data AIS Polbeng. *Seminar Nasional Industri dan Teknologi (SNIT)*, Politeknik Negeri Bengkalis.

Sururi, N., Thoib, I., Nugraha, D. S., Bayu, F., & Shah, M. S. (2025). Perancangan Aplikasi Membership Gym Berbasis Web untuk Optimalisasi Layanan Pelanggan. *JUKTISI: Jurnal Ilmu Komputer, Teknik Informatika dan Sistem Informasi*, *4*(2), 1121–1132.