Published online on the page: https://jurnal.ilcagam.org/index.php/jsti

J S T I
Jurnal Serumpun Teknik Informatika
| ISSN (*Online*) 3123-965x | DOI: https://doi.org/10.66485/jsti.v1i2.19

Implementation of Two-Factor Authentication (2FA) Using a REST API-Based WhatsApp Gateway to Prevent Fake Bidders on an Online Auction Platform

Rizky Parlika[1,], Hamdi Indra[2], Tegar Satria Kirana[3]*

[1,3]Universitas Pembangunan Nasional "Veteran" Jawa Timur, Surabaya, Indonesia
[2]Universitas Persada Bunda, Pekanbaru, Indonesia

Article Information

Article History:
Submit: 22 Maret 2026
Accepted: 25 April 2026
Published: 30 April 2026

Keywords

Two-Factor; Authentication; WhatsApp Gateway; REST API; Webhook.

Correspondence

E-mail: rizkyparlika.if@upnjatim.ac.id*

ABSTRACT

Account security and identity validity are crucial aspects of online auction platforms to prevent price manipulation by fake bidders. Conventional authentication methods are often vulnerable to cyber-attacks or compromise user convenience for the sake of security. This study aims to implement a Two-Factor Authentication (2FA) system on the Mokasindo auction platform using WhatsApp Gateway integrated via REST API technology. The development method includes Webhook mechanisms for real-time user phone number validation and AJAX Short Polling techniques to deliver auto-login features without page refreshing. Black Box testing results indicate that the system successfully verifies user identity accurately and mitigates the risk of fictitious account registration. This implementation offers an optimal balance between system security and User Experience (UX), with an average recorded verification process latency of only 3.5 seconds. This solution proves effective in creating a more secure, responsive, and trustworthy auction ecosystem for users.

1. Introduction

The rapid development of information technology has driven the transformation of the global economy toward a dynamic digital ecosystem (Parlika, 2020), where social commerce has emerged as one of its key pillars. In this context, electronic auction systems (e-auction) have become an effective trading mechanism for determining market prices in a transparent and competitive manner (Nabeel Al-Qirim, Kamel Rouibah, Hasan Abbas, 2022). The success of auction platforms largely depends on users' trust in the system's integrity. Empirical investigations show that users' perceived security has a significant positive correlation with their active participation in bidding processes (Gsu, 2025). Without adequate security assurance, the auction ecosystem risks being abandoned by its users.

However, as the volume of digital transactions increases, cybersecurity threats are also becoming more complex and harder to detect. Online platforms are now primary targets of various cyberattacks aimed at user accounts (Imanova & Mahmudova, 2025). Weak access control in many web-based systems makes sensitive user data vulnerable to exploitation by irresponsible third parties (Kashmar et al., 2016). comprehensive review of online social network security emphasizes the need for stricter privacy protection mechanisms to safeguard user data from leakage and misuse (Kumar et al., 2021).

One of the major security vulnerabilities in modern authentication systems is the reliance on single-factor verification methods. In today's end-to-end encryption era, conventional web authentication mechanisms that rely solely on username and password combinations are no longer sufficient to

guarantee user identity validity (Blessing et al., 2021). Modern authentication frameworks require the implementation of layered security factors or Multi-Factor Authentication (MFA), especially for cloud-based and publicly accessible systems (Mostafa et al., 2023). The absence of a second security layer makes it easier for attackers to take over user accounts through brute force attacks or credential theft.

In the specific domain of online auctions, security threats manifest in the form of fake accounts and manipulative behaviors. The spread of false information and fictitious identities poses a serious challenge that disrupts auction price stability (Asfari et al., 2025). In addition, social engineering attacks are often used to trick users into unknowingly granting access to their accounts (Mehta et al., 2021). Human-computer interaction in auction environments requires specialized security interfaces capable of detecting such behavioral anomalies without compromising the comfort of legitimate users (Waldemar Karwowski, 2022).

Various identity verification methods have been implemented to address these issues, but each has its own limitations. Email-based verification, although popular, is often ineffective because verification messages frequently end up in spam folders or are delayed. Furthermore, email is a primary vector for phishing attacks designed to steal user credentials (Nonye Benedeth Ezeaka, 2024), (Nyasvisvo & Chigada, 2023). SMS (Short Message Service)-based verification offers a higher level of security and service personalization (R. K. Amin et al., 2025). However, the relatively high operational cost of SMS becomes a burden for service providers, especially for platforms with large user bases.

As a more efficient alternative, instant messaging applications such as WhatsApp offer great potential as a verification medium. Digital forensic analysis indicates that the evolving social media landscape positions WhatsApp as an application with high data security standards (Brown, 2025). Although spyware threats exist on mobile devices, WhatsApp's encrypted security architecture still makes it a reliable communication channel. Moreover, the adoption of digital technology in critical services, such as elderly care, has demonstrated that WhatsApp-based authentication is easily accepted across various user groups due to its familiarity (Care, 2025).

The use of WhatsApp Gateway to send One-Time Password (OTP) codes has been technically proven as a valid login verification method (Nasution et al., 2024). Integration of the WhatsApp API enables systems to validate user phone numbers in real time before granting access to core services, thereby minimizing the risk of registering accounts with fictitious numbers (Pardede & Marbun, 2024). The development of Android-based mobile applications has also increasingly adopted OTP mechanisms as a new security standard (Ashari et al., 2022). This aligns with public sentiment analysis of phone number identification applications, which shows a high market demand for user identity validity in digital communication (Kurniawan et al., 2025).

Although security is a priority, user experience (UX) must not be overlooked. There is a close relationship between system security and customer satisfaction; overly rigid systems can reduce user interest (N. Amin & Salim, 2025). Security interfaces are often considered important but not intuitive and difficult for general users to trust (Daffalla et al., 2023). Therefore, the main challenge in developing 2FA systems is balancing enhanced security with usability in web-based systems (R. K. Amin et al., 2025). Users expect a verification process that is fast, transparent, and minimally disruptive.

To address UX challenges, technical innovations in data transmission and validation methods are required. Lightweight data access systems are essential for public devices with limited resources (Pandey & Chauhan, 2025). The use of time-limited OTP must be supported by automatic detection mechanisms on the user interface side. Integration of graphical authentication or periodic data polling methods can provide immediate visual feedback to users (Jeevarathinam & Akilan, 2025). Complex system approaches such as blockchain and other emerging technologies also emphasize the importance of authentication mechanisms that respond dynamically to changes in security status (Rights, 2023).

Based on the problems and technological opportunities described above, this study aims to implement a Two-Factor Authentication (2FA) system based on WhatsApp Gateway on the Mokasindo auction platform. This research offers novelty in integrating third-party APIs with a real-time polling method using AJAX. This method allows the system to automatically detect user verification status and perform auto-login without requiring users to reload the page. This solution is expected to effectively eliminate fake bidder accounts while delivering a seamless and modern user experience.

2. Research Methods

This study applies a modified prototyping-based system development method to support third-party API integration. The primary focus of the methodology lies in designing a secure data exchange mechanism between the application server, the WhatsApp service provider (Gateway), and the end-user interface.

2.1. Proposed System Architecture

The system is designed using a Client-Server architecture integrated with external services through a REST API. The main entities in this system consist of the User (Client), the Application Server (Laravel), the Database Server (MySQL), and the WhatsApp Gateway (Fonnte). The data communication flow begins when a user registers, where the server does not immediately store the data as an active account, but instead holds it in a pending status until verification is successfully completed through the WhatsApp channel.

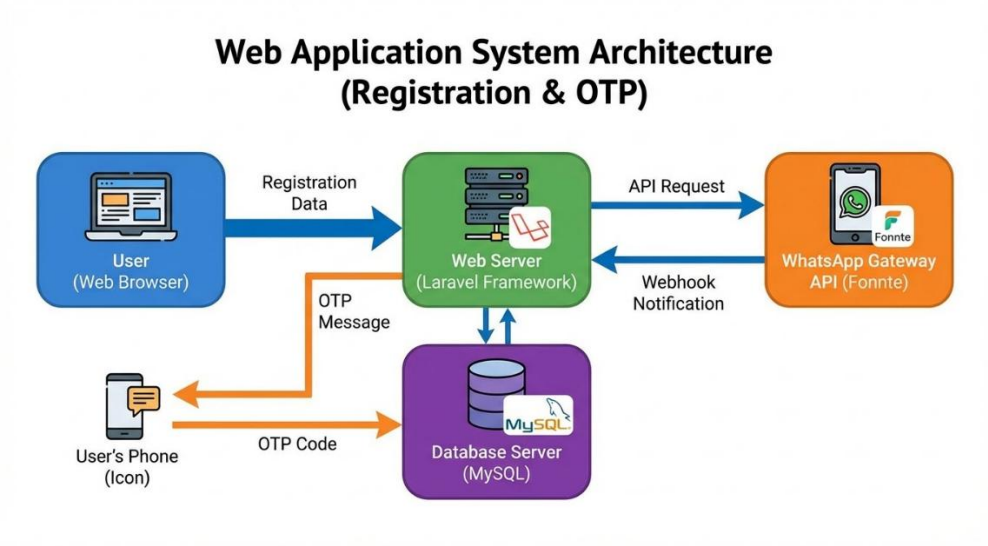

Figure 1. Data Communication Architecture of the 2FA Authentication System

Figure 1 illustrates the system topology. Its main focus is to show which entities communicate with one another.

Flow Description:

1. Client (User): Accesses the website using a browser.
2. Web Server (Laravel): Serves as the central logic unit. It manages the database and API.
3. Database (MySQL): Stores user data, including accounts with pending status.
4. WhatsApp Gateway (Fonnte): A third-party service that bridges our server with the WhatsApp application on the user's mobile phone.
5. Webhook Path: A critical communication path where Fonnte reports back to our server when the user replies to a message.

2.2. OTP Generation and Delivery Algorithm

The security process begins with the generation of a unique verification code. This algorithm ensures that each registration session has a simple cryptographic token that is unique in nature. The verification code C is formed by a string randomization function R with a specified character length L, which is formulated as follows:

$$C = \text{"PREFIX"} + \text{Upper}(R(L)) \qquad (1)$$

Where:

a. PREFIX is the constant string "REG-".
b. R is a pseudo-random string generator function.
c. L is the length of the random characters (5 characters).

After the code C is generated, the system stores it in the database D along with the user identity U and sets the initial status S_init to false (inactive).

$$U_{new} = \{"email", "phone", "password", C, S_{init} = 0\} \qquad (2)$$

The pseudocode for the OTP code delivery logic via the Fonnte API is described in Algorithm 1 below:

Code 1. OTP Generation and Delivery

```
Input: Data Registrasi (Nama, Email, NoHP)
Output: Pesan Terkirim, Record User Tersimpan

1. BEGIN
2.   Validasi format NoHP (pastikan awalan 62/08)
3.   Generate Kode Unik C sesuai persamaan (1)
4.   Simpan User ke Database dengan status is_active = FALSE
5.   Siapkan Payload HTTP Request:
       target = NoHP
       message = "Halo [Nama], Kode verifikasi Anda: " + C
6.   Kirim HTTP POST ke "https://api.fonnte.com/send"
7.   IF Response Code == 200 THEN
8.      Redirect User ke Halaman Verifikasi (Waiting Page)
9.   ELSE
10.     Return Error Message
11.  END IF
12. END
```

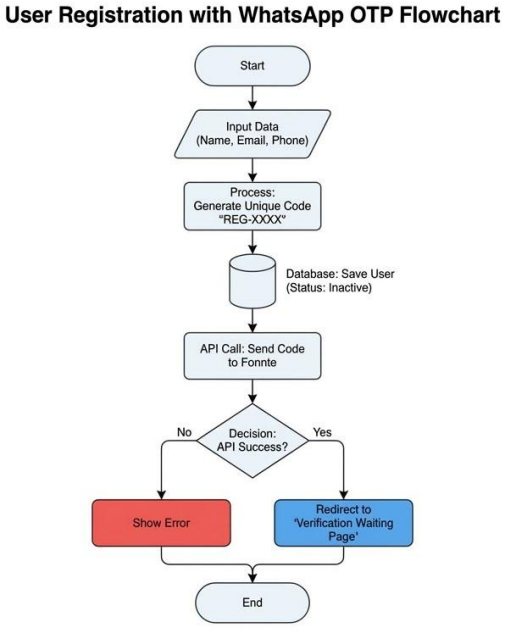

Figure 2. Flowchart of Registration and OTP Delivery Process

Figure 2 focuses on the backend logic when the "Register" button is pressed. It visually represents the flow described in Algorithm 1.

Flow Description:

1. The system receives user input.
2. The system generates a unique code (e.g., REG-A1B2C).
3. The user data is stored in the database but remains “locked” (is_active = 0).
4. The system attempts to communicate with Fonnte.
5. If Fonnte responds with "OK", the user is redirected to a waiting page. If an "Error" occurs, the user is returned to the registration form.

2.3. Webhook Mechanism and Message Validation

To verify users, the system employs a Webhook method that listens to incoming messages in real time. Unlike conventional OTP methods where users manually enter the code on a website, this approach requires users to send the code back via WhatsApp. This simultaneously validates two aspects: ownership of the WhatsApp account and the validity of the phone number.

When an incoming message is received by the Webhook endpoint, the system parses the message content M and the sender’s number P. Validation is performed by matching M with the verification code C stored in the user table.

The validation function $V(M, P)$is defined as:

$$V(M,P) = \begin{cases} \text{True}, & \text{if } \exists U \in D: (U_{code} = M \wedge U_{is\_active} = 0) \\ \text{False}, & \text{otherwise} \end{cases} \quad (3)$$

If the validation function returns **True**, the system updates the user status in the database as follows:

$$Update(U) \rightarrow \{\, U_{phone} = P,\ U_{is\_active} = 1,\ U_{code} = \text{NULL} \,\} \quad (4)$$

2.4. Real-time Polling Method on the User Interface

To provide a seamless and responsive user experience, the web interface implements a Short Polling mechanism using AJAX (Asynchronous JavaScript and XML). This technique allows the browser to periodically request the verification status from the server without requiring user intervention.

The polling interval T is set to 3 seconds (3000 ms) to balance interface responsiveness and server load. The status-checking logic on the client side is described in Algorithm 2.

Code 2. Real-time Verification Status Polling

```
Input: User ID ($id)
Output: Redirect ke Dashboard

1. BEGIN
2.   Set Interval Timer T = 3000 ms
3.   LOOP setiap T:
4.      Kirim GET Request ke "/cek-status-verifikasi/{$id}"
5.      Terima JSON Response (R)
6.      IF R.status == "success" THEN
7.         Hentikan Loop (ClearInterval)
8.         Tampilkan Notifikasi "Akun Aktif"
9.         Redirect ke Halaman Utama (Dashboard)
10.     ELSE
11.        Tetap di halaman, tunggu interval berikutnya
12.     END IF
13.  END LOOP
14. END
```

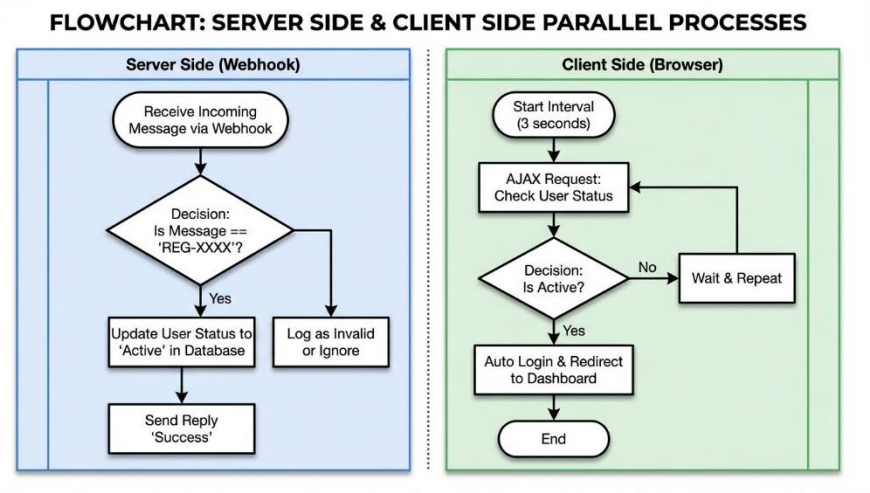

Figure 3. Flowchart of Polling Logic and Auto-Login

Figure 3 illustrates the frontend logic (browser-side). It represents a background script that continuously runs on the waiting page.

Flow Description:

1. When the page loads, a timer starts (every 3 seconds).
2. The browser sends a request to the server: "Is this User ID already active?"
3. The server checks the database.
4. If the status is still 0 (inactive), the browser remains idle and waits for another 3 seconds before repeating the request.
5. If the status becomes 1 (active)—triggered by the Webhook—the browser automatically logs in the user and redirects them to the Dashboard.

3. Results and Discussion

The results of this study indicate that the Two-Factor Authentication (2FA) system based on a WhatsApp Gateway has been successfully implemented on the Mokasindo auction platform. The system is capable of handling user registration, OTP code delivery, receiving replies via Webhook, and automatically updating user status using a polling method.

3.1. Implementation of Program Logic

The core of this security mechanism lies in the backend, which handles incoming requests from the WhatsApp server (Fonnte). In Code 1, the implementation of the handle function in the WebhookController is presented. This function is responsible for validating incoming messages. If the message matches the unique code format (REG-XXXX) stored in the database, the system updates the user status to active (is_active = 1) and assigns the user's phone number based on the valid WhatsApp sender number.

Code 3. Implementation of Webhook Logic in Laravel

```
public function handle(Request $request)
{
    $sender = $request->input('sender');
    $message = trim($request->input('message'));

    // Logika Validasi Kode REG-XXXX
    if (str_starts_with(strtoupper($message), 'REG-')) {
        $user = User::where('verification_code', $message)->first();
```

```
        if ($user) {
            $user->update([
                'phone_number' => $sender,
                'is_active' => true,
                'verification_code' => null
            ]);
            // Kirim balasan sukses ke user...
        }
    }
    }
```

In Code 3, it is shown that the system does not trust the phone number input provided in the initial registration form. Instead, it retrieves the actual phone number ($sender) directly from the WhatsApp payload received. This approach effectively closes a security loophole where users could register using fake numbers or numbers belonging to others.

3.2. User Interface Implementation

The user interface is designed to guide users through the verification process independently. In Figure 4, an instruction page is displayed immediately after the user completes the registration form. This page presents a unique code along with a Call-to-Action (CTA) button that directs the user straight to the WhatsApp application.

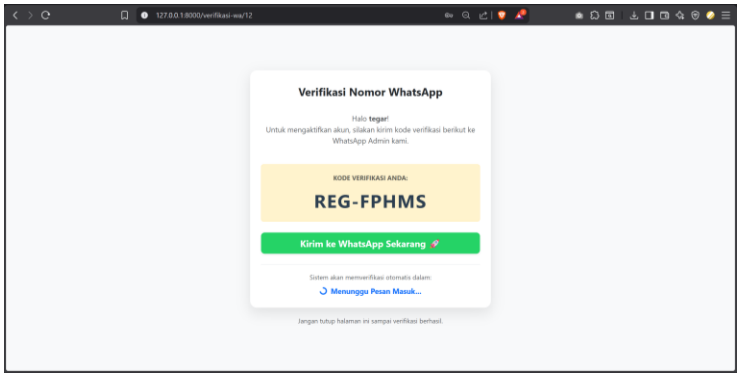

Figure 4. Verification Instruction Page and Unique Code

After the user sends the message, the system automatically provides visual feedback. In Figure 5, the conversation flow shows the bot responding to the user's verification message, indicating that the account has been successfully activated. At the same time, the web page in the user's browser automatically redirects to the main page (Dashboard) without requiring a manual refresh.

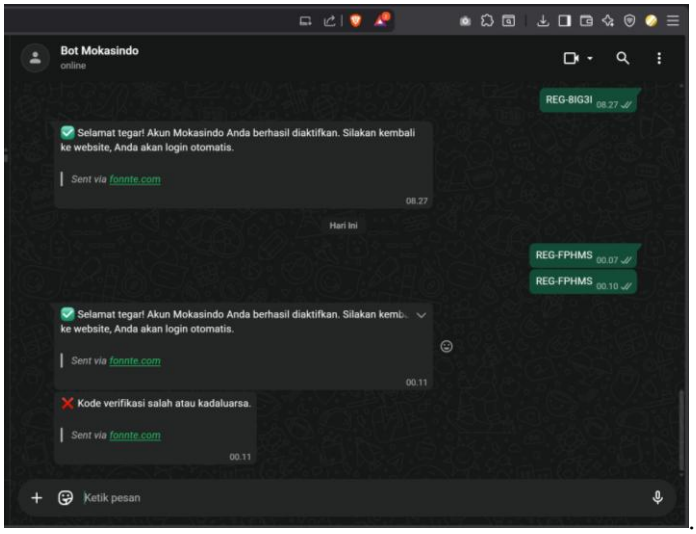

Figure 5. WhatsApp Bot Response

3.3. Functional Testing (Black Box Testing)

System testing was conducted using the Black Box Testing method to ensure that each function operates according to the specified requirements. The test scenarios include handling both valid and invalid inputs, as well as system responses to incorrect verification codes. The results of the functional testing are summarized in Table 1.

Table 1. Black Box Testing Results

Test Scenario	Expected Result	Test Result	Conclusion
New User Registration	System stores data with inactive status and generates a unique code	As Expected	Valid
Login Without Verification	System denies access and displays an error message	As Expected	Valid
Send Correct Code to WhatsApp	Bot replies with an error message, user status remains inactive	As Expected	Valid
Send Correct Code to WhatsApp	User status changes to active, bot replies with success message	As Expected	Valid
Auto-Login (Polling)	Web page automatically redirects when status becomes active	As Expected	Valid

Based on Table 1, all test scenarios resulted in valid outcomes. This demonstrates that the 2FA mechanism successfully prevents unauthorized access (scenario 2) and properly handles user input errors (scenario 3).

3.3. Performance Analysis and Discussion

A crucial aspect of this implementation is latency, or the delay experienced by users from the moment they send a WhatsApp message until the web system performs automatic login. The total latency $L_{total}$is influenced by the WhatsApp network transmission time T_{wa}, the webhook processing time on the server T_{server}, and the AJAX polling interval T_{poll}. This relationship can be expressed as:

$$L_{total} \approx T_{wa} + T_{server} + \frac{T_{poll}}{2} \qquad (5)$$

In this study, the polling interval $T_{poll}$is set to 3 seconds. Based on average testing results, $T_{wa}$and T_{server}take approximately 1–2 seconds depending on network conditions. Therefore, the average user waiting time is around 3.5 seconds.

The discussion of these results confirms that using a polling method with a 3-second interval provides an optimal balance between server load and interface responsiveness. Compared to email-based verification methods—which typically require 1–5 minutes (including opening the email application and locating the message)—the WhatsApp Gateway approach offers a significantly faster and more seamless user experience.

Furthermore, real-time phone number validation significantly reduces the risk of fake bidder accounts, as each account is tied to a valid and active mobile number, which is far more difficult to falsify at scale compared to email addresses.

4. Conclusion

This study successfully achieved its main objective of designing and implementing a Two-Factor Authentication (2FA) security system based on a WhatsApp Gateway on the Mokasindo auction

platform. The functional testing results demonstrate that the integration of third-party APIs with a Webhook mechanism is capable of validating user identities in real time, ensuring that every registered account is linked to an active phone number. This effectively mitigates the risk of fake account (fake bidder) registrations that often disrupt the integrity of the auction ecosystem.

Beyond security, the implementation of the Real-time Polling method on the user interface has proven to be significant in bridging the gap between strong security and usability. The auto-login feature, triggered by changes in verification status on the server, provides a seamless user experience without requiring manual page refreshes, with an average system latency of approximately 3.5 seconds, which is considered efficient compared to conventional email-based verification methods.

As a future development, it is recommended to replace the Short Polling mechanism with WebSocket technology to improve server resource efficiency at scale, as polling can impose a higher load due to repeated requests. In addition, the existing WhatsApp Gateway infrastructure has strong potential to be extended to other transactional features, such as automatic auction winner notifications, outbid alerts, and payment reminders. The implementation of these additional features is expected to create an auction ecosystem that is not only secure against account manipulation but also responsive and informative for all users.

.

References

Amin, N., & Salim, S. A. (2025). *RESEARCH IN MANAGEMENT OF TECHNOLOGY AND RMTB The Relationship between Security Satisfaction in Digital Wallet Services*. *6*(2), 374–387.

Amin, R. K., Khayat, G. A. El, Sahn, F. El, & Amer, A. A. (2025). *Enhancing E-Banking Security and Personalization through Convolutional Neural Network-Based Facial Recognition*. *3*. https://doi.org/10.3844/jcssp.2025.2323.2336

Asfari, D. Y., Ruslan, D. D., Syahira, A., & Amir, A. S. (2025). *Students and Fake News : Exploring Digital Literacy and Information Security Among Young Adults*. *1*(6), 198–204.

Ashari, I. F., Zuhdi, M. F., Gagaman, M. T., & Denira, S. T. (2022). *Kolepa Mobile Application Development Based on Android Using SCRUM Method (Case Study : Kolepa Minigolf and Coffe Shop)*. *6*(1), 104–112.

Blessing, J., Hugenroth, D., Anderson, R. J., & Beresford, A. R. (2021). *SoK : Web Authentication in the Age of End-to-End Encryption*.

Brown, J. (2025). *LSU Scholarly Repository From Devices to the Cloud : Digital Forensics in the Changing Social Media Landscape FROM DEVICES TO THE CLOUD : DIGITAL FORENSICS IN THE CHANGING SOCIAL MEDIA*.

Care, E. (2025). *Digitalization for Improving Elder Care*.

Daffalla, A., Bohuk, M., Dell, N., Tech, C., Bellini, R., Ristenpart, T., Tech, C., & Symposium, U. S. (2023). *Account Security Interfaces : Important , Unintuitive , and Untrustworthy*.

Gsu, S. (2025). *Deciding to Fail : Three Essays*.

Imanova, S., & Mahmudova, S. (2025). *Cyberattacks and social media account security 1*. 105–113. https://doi.org/10.21303/2313-8416.2025.003766

Jeevarathinam, A., & Akilan, E. (2025). *Graphical Click Point Authentication : Enhancing Resistance against Shoulder Surfing*. *8*(3). https://doi.org/10.15680/IJMRSET.2025.0803209

Kashmar, N., Adda, M., Atieh, M., & Ibrahim, H. (2016). *ACCESS CONTROL IN CYBERSECURITY AND SOCIAL MEDIA*. *2002*, 69–105.

Kumar, A., Somya, J., Sahoo, R., & Kaubiyal, J. (2021). Online social networks security and privacy : comprehensive review and analysis. *Complex & Intelligent Systems*, *0123456789*. https://doi.org/10.1007/s40747-021-00409-7

Kurniawan, R. D., Yohannis, A., & Atmojo, W. T. (2025). *Sentiment Analysis of Getcontact Application Reviews on Google Play Store Using Naive Bayes Algorithm*. *6*(4), 2848–2858.

Mehta, A., Vora, D., & Khatri, J. (2021). *A Review of Social Engineering Attacks and their Mitigation Solutions*. *10*(10), 215–220.

Mostafa, E., Hassan, M. M., & Said, W. (2023). *An Interactive Multi-Factor User Authentication Framework in Cloud Computing*. *23*(8).

Nabeel Al-Qirim, Kamel Rouibah, Hasan Abbas, Y. H. (2022). *Factors Affecting the Success of Social Commerce in*

Kuwaiti Microbusinesses : 30(1), 1–31. https://doi.org/10.4018/JGIM.313944

Nasution, A. B., Yugo, A., & Hrp, N. (2024). *Implementation of OTP Code as Application Login Verification Via Whatsapp Implementasi Kode OTP Sebagai Verifikasi Login Aplikasi Via Whatsapp*. *3*(4), 395–402.

Nonye Benedeth Ezeaka, E. E. I. (2024). *INFLUENCE OF WHATSAPP ONLINE PHISHING MESSAGES ON DATA SECURITY AMONG UNDERGRADUATES IN ANAMBRA STATE*. *7*(4), 273–282. https://doi.org/10.52589/AJSSHR-LR7BIBZD

Nyasvisvo, B., & Chigada, J. M. (2023). *Phishing Attacks : A Security Challenge for University Students Studying Remotely Phishing Attacks : A Security Challenge for University Students Studying Remotely*. *15*(2).

Pandey, S., & Chauhan, N. (2025). *Eliminating Credential Risk : A Lightweight Data Access System For Public Devices*. 706–710. https://doi.org/10.48175/IJARSCT-25689

Pardede, I. A., & Marbun, N. (2024). *Journal of Computer Networks , Architecture and High Performance Computing Design of Goods Inventory Information System Using Visual Basic . Net (Case Study : CV . Barokah Medan) Journal of Computer Networks , Architecture and High Performance Computing*. *6*(3), 967–975.

Parlika, R. (2020). *IMPLEMENTASI API REGION VISUAL BASIC 6 UNTUK MEMBENTUK HURUF IMPLEMENTASI API REGION VISUAL BASIC 6 UNTUK MEMBENTUK*. *December*. https://doi.org/10.36564/njca.v5i2.191

Rights, M. (2023). *City , University of London Institutional Repository City , University of London Blockchain based ecosystems : a complex systems approach*.

Waldemar Karwowski, P. M. (2022). *Human-Computer Interaction and Cybersecurity Handbook*.