TSABIT
Journal of Computer Science
https://journal.ilmubersama.com/index.php/tsabit

Click here and write your Article Category

Implementation of the Discrete Wavelet Transform (DWT) Algorithm for Video File Security

Dea Sintia [1], Mhd. Zulfansyuri Siambaton [2], Rachmat Aulia [3]

[1,2] *Department of Informatics Engineering, Faculty of Engineering, Universitas Islam Sumatera Utara, Medan, 20217, North Sumatra, Indonesia*
[3] *Department of Informatics Engineering, Faculty of Engineering and Computer, Universitas Harapan Medan, Medan, North Sumatra, Indonesia*

ARTICLE INFORMATION

Received: February 00, 00
Revised: March 00, 00
Available Online: April 00, 00

KEYWORDS

Discrete Wavelet Transform; video file security; digital data security; digital signal processing; multimedia data protection

CORRESPONDENCE

Phone: +6281360164791
E-mail: deasintia.mione@gmail.com

ABSTRACT

The growing reliance on digital video across education, entertainment, communication, and security applications has made video one of the most widely used forms of multimedia data, yet this growth has also exposed video files to risks such as unauthorized access, data theft, content manipulation, and misuse of digital information. This study addresses these risks by implementing the Discrete Wavelet Transform (DWT) algorithm as a signal-processing-based method for securing video files. DWT decomposes each video frame from the time domain into the frequency domain, producing a visual representation that is difficult to recognize, while the Inverse Discrete Wavelet Transform (IDWT) reconstructs the video back to a form approximating the original. A security key was applied to the wavelet coefficients to prevent reconstruction without proper authorization. The system was built in Python using OpenCV for video processing, PyWavelets for the DWT/IDWT operations, NumPy for numerical computation, and CustomTkinter for the desktop interface, following a Research and Development approach with a Waterfall development model. Black box testing confirmed that the encryption and decryption functions operated correctly across the tested video files, and quality evaluation using Peak Signal to Noise Ratio (PSNR), Mean Squared Error (MSE), Signal to Noise Ratio (SNR), and entropy showed that the encrypted videos exhibited substantial visual distortion relative to the originals, with PSNR values around 10–11 dB, MSE values in the range of roughly 5,100–5,700, and an entropy of approximately 7.37, indicating that the visual structure of the video content was concealed effectively. These findings suggest that DWT can serve as a practical signal-transformation-based approach to video data security, although its protection level remains below that of dedicated cryptographic algorithms such as AES or RSA, leaving room for future work that combines DWT with stronger cryptographic techniques.

INTRODUCTION

The rapid advancement of information and communication technology has driven a substantial increase in the use of digital data, particularly video files, across fields such as education, entertainment, social media, documentation, and security systems. Video has become one of the primary media for conveying information because it can present visual and audio content simultaneously. However, the same ease of storing and distributing video files over the internet has introduced security concerns, since video data remains vulnerable to threats such as theft, interception, unauthorized duplication, and content manipulation by irresponsible parties, all of which can be especially damaging when the video in question is private, confidential, or otherwise valuable [1,2,3].

Addressing this vulnerability calls for a security method capable of protecting video files from unauthorized access, copying, or alteration. One viable approach is to apply a digital signal transformation technique known as the Discrete Wavelet Transform (DWT), which converts data from the time domain into the frequency domain. Through this

https://doi.org/10.12345/xxxxx

transformation, the structure of video data can be converted into a form that is difficult to recognize directly, thereby raising the security level of the video file. In this study, DWT is used to scramble and obscure video content so that it cannot be understood without an inverse transformation process; once DWT has been applied, the resulting coefficient structure changes such that the file cannot be displayed normally without proper authorization. To return the video to its original form, the Inverse Discrete Wavelet Transform (IDWT) is applied, which allows the secured file to be restored with quality approaching that of the original, provided that the correct transformation parameters are used [4,5.

The anticipated outcome of this research is a system capable of securing video files effectively, such that processed videos become visually unrecognizable while still being reconstructable with preserved quality, ultimately strengthening the confidentiality of video content during both storage and transmission. At the same time, DWT carries certain limitations that are important to acknowledge: it is not inherently a cryptographic algorithm, and as such its security level remains below that of dedicated encryption methods like AES or RSA. The transformation and reconstruction of large video files also demand considerable computation time, and any error introduced during the transformation or inverse-transformation stages can degrade the quality of the reconstructed video. For these reasons, the present study treats DWT specifically as a signal-transformation-based security method, leaving open the possibility of combining it with other cryptographic algorithms in future development.

Based on this background, the research questions guiding this study concern how the DWT algorithm can be applied to secure video files, how the algorithm functions to prevent unauthorized access, what results are produced after the transformation and inverse-transformation processes, and to what extent DWT succeeds in preserving both the security and the quality of video files. The scope of the study is limited to securing video files using DWT in the MP4 format, implemented in Python, without comparing DWT against other security or cryptographic algorithms such as AES, DES, or RSA, and the system testing focuses solely on verifying the success of the security process and the resulting video quality after reconstruction.

This research is intended to contribute theoretically to the field of informatics engineering, particularly in multimedia data security and digital signal processing, while also serving as a practical starting point for securing video files from unauthorized access and as a reference for further academic work, such as combining DWT with other cryptographic algorithms to achieve stronger protection.

METHOD

Research Methods

This research was conducted within the Fakultas Teknik Informatika environment and at the researcher's residence, using a personal computer for designing, building, and testing the video security application. The study spanned approximately four months, from the literature review stage through final report preparation, covering system requirements analysis, application design, program implementation, system testing, and analysis of results. The hardware used in development consisted of a laptop with an Intel Core i3 or Ryzen 3 processor, a minimum of 4 GB of RAM, 256 GB of SSD or HDD storage, and standard input devices [6,7,8]. On the software side, the system was built using Python as the primary programming language, supported by OpenCV for video and image processing, PyWavelets for implementing the DWT and IDWT algorithms, NumPy for numerical data processing, CustomTkinter for constructing the application interface, and VLC Media Player for playback during testing.

The study adopted a Research and Development (R&D) approach, reflecting its focus on developing a software product for digital video file security, and followed the Waterfall development model because of its systematic, structured stages that facilitate application development from initial analysis through final testing. The first stage, requirements analysis, involved a literature review covering the DWT algorithm, digital video processing, and multimedia security techniques, together with an analysis of functional and non-functional system requirements to establish the foundation for development[9,10]. The second stage, system design, translated the requirements analysis into system flowcharts, a video security flowchart, a video reconstruction flowchart, and a user interface design. The third stage, implementation, involved coding the application in Python using the OpenCV, PyWavelets, NumPy, and CustomTkinter libraries, covering both the video security process and the video reconstruction process built around DWT and IDWT. The fourth stage, testing and validation, verified that all system functions operated correctly and met the research requirements, using black box testing for functional verification and the PSNR and MSE parameters for video quality evaluation [11,12].

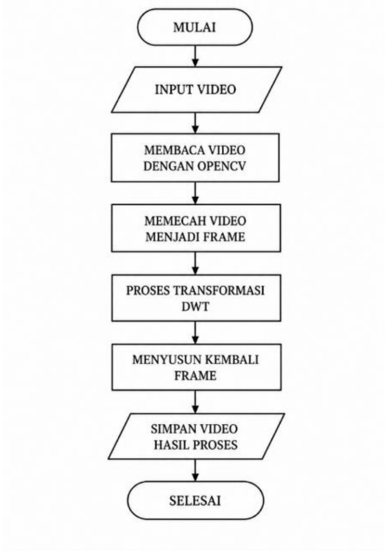

Figure 1. System Flowchart

The research distinguished between input and output variables. The input variable consisted of digital video files in MP4 format, varying in size and duration, that were processed through frame reading, DWT transformation, and reconstruction. The output variables consisted of the secured video and the reconstructed video: the secured video resulted from DWT processing and became difficult to recognize directly, while the reconstructed video was restored to a form approximating the original through IDWT and was subsequently evaluated using the PSNR and MSE parameters. Data collection combined a literature study of journals, books, research articles, and prior theses related to multimedia security, digital video processing, and the DWT algorithm; direct observation of video security issues such as data theft, misuse, and content manipulation in unsecured video files; and direct experimentation in which several video files of varying size and duration were processed through the built application to assess its security and reconstruction performance. The resulting data were analyzed descriptively based on the PSNR and MSE quality parameters, where PSNR values above 40 dB, between 30 and 40 dB, between 20 and 30 dB, and below 20 dB were interpreted as indicating very good, good, adequate, and poor video quality respectively, while MSE values approaching zero indicated very small differences between videos and larger values indicated greater differences.

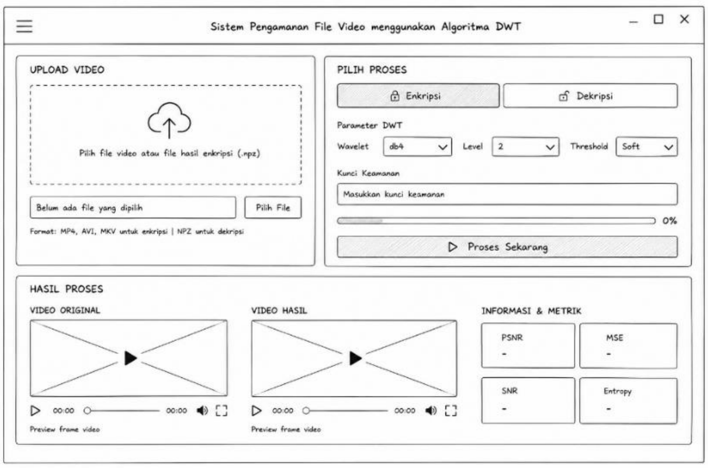

Figure 2. Main System Page Mockup

The functional requirements established for the system specified that it should be able to accept video input, perform video security processing using the DWT algorithm, perform video reconstruction using IDWT, store both the secured and reconstructed video output, display a video preview, and display the results of video quality analysis using the PSNR and MSE parameters. The non-functional requirements specified that the system should present an easily understandable interface, operate stably, process video without errors, and be built using Python.

System design was visualized through flowcharts to clarify the overall application workflow. At the system level, the process begins with the user providing video input, after which the system reads the video using OpenCV and decomposes it into image frames; the DWT transformation is then applied, the resulting output is reassembled into a new video file, and the secured video is stored once processing is complete. The video security flowchart details the security-specific stage: after the video is read and decomposed into frames, each frame is processed using the DWT algorithm to produce frequency coefficients that obscure the visual information, after which the processed frames are reassembled into the secured video and stored. The reconstruction flowchart describes the corresponding restoration stage: the secured video is read and decomposed into frames, each frame is processed using the inverse DWT to restore the original data structure, the restored frames are reassembled into a complete video, and the reconstructed video is stored. Quality evaluation using MSE and PSNR was incorporated into this overall workflow to assess the resulting video against the original.

The user interface was designed around a single main page intended to simplify the security workflow for the user. The main page provides video upload, a choice between encryption and decryption processes, DWT parameter settings such as wavelet type, decomposition level, and threshold, a security key input field functioning as a password for the security process, a progress bar indicating processing status, previews of the original and processed video, and an information panel displaying quality metrics including PSNR, MSE, SNR, and entropy. The upload interface allows the user to select a video file through a dedicated button, after which the selected file name is displayed, with support for MP4, AVI, and MKV formats. The encryption and decryption interface presents separate buttons for initiating the security and reconstruction processes respectively, alongside the DWT parameter settings, the security key field, and a button to begin processing. The testing results interface displays the original and processed video alongside the resulting PSNR, MSE, SNR, and entropy values, allowing the user to evaluate the outcome of the security process directly within the application. To illustrate the underlying mechanism of the DWT algorithm, a manual calculation was performed using a simple 2×2 image frame matrix containing the values 100, 120, 80, and 60. The row transformation produced averages of 110 and 70 and differences of −10 and 10 for the first and second rows respectively, yielding an intermediate matrix of 110, −10, 70, and 10. The subsequent column transformation produced the four DWT sub-band coefficients: LL = 90, HL = 20, LH = 0, and HH = −10, confirming that the original data had been successfully decomposed into the four expected sub-bands. Applying a security key of $K = 12$ to the principal LL coefficient through the encryption equation $C' = C + K$ transformed the value from LL = 90 to LL′ = 102, after which the corresponding decryption equation $C = C' - K$ restored the original value of LL = 90 using the same key, demonstrating that the security mechanism could be reversed correctly when the proper key was supplied. Reconstruction via IDWT then reversed both the column and row transformations in turn, first recovering the intermediate matrix of 110, −10, 70, and 10 from the sub-band coefficients and subsequently recovering the original frame values of 100, 120, 80, and 60, confirming that the combination of DWT, key-based security, and IDWT could restore the original data accurately.

The complete security process for an actual video file followed the same logic at scale: the video was read and decomposed into frames using OpenCV, each frame was transformed using DWT to produce frequency coefficients, the resulting coefficients were modified to obscure the visual content, and the processed frames were reassembled into the secured output video, which was then stored for the subsequent reconstruction stage. Reconstruction reversed this sequence, applying the inverse DWT to each frame of the secured video to restore its data structure before reassembling the frames into a complete, restored video. System testing was conducted on both system functionality and the quality of the resulting video, applying the PSNR and MSE parameters to evaluate reconstruction quality against the original video.

RESULTS AND DISCUSSION

The video security application was implemented as a desktop program in Python, drawing on OpenCV for reading and processing video, NumPy for numerical operations, PyWavelets for the wavelet transformation, and CustomTkinter for the user interface, and was run on a computer using the Windows 10 operating system. The implementation followed the design established in the methods section, applying DWT to the encryption process and IDWT to the corresponding decryption process for digital video. The application allows the user to select a video file through a dedicated button, after which the system reads and displays information such as file name, format, size, and resolution so that the user can review the characteristics of the video before processing. The user then selects either the encryption or decryption process and configures the DWT parameters, including wavelet type, decomposition level, and threshold, before the system applies the DWT transformation to each frame of the video to produce the secured output.

The application interface was designed to be straightforward and easy to use. The main page integrates the file selection button, the security key input field, the encryption and decryption buttons, and an information panel displaying the results of each process, functioning as the central point from which the user controls the entire workflow from video selection through final storage. The encryption and decryption selection menu allows the user to choose between securing a video using the DWT algorithm and the security key, or restoring a previously secured video using the corresponding key, serving as the initial step before the system carries out the chosen process. The security key input interface requires the user to enter the key used for either encryption or decryption, ensuring that only a user with the correct key can access or restore the video to its original form, a step that constitutes an important part of strengthening the security provided by the DWT-based process. The results display presents previews of the original and processed video so that the user can confirm that the selected operation completed successfully, serving as an indicator that the system has correctly secured or restored the video as requested. The metrics panel presents the file information together with the resulting MSE and PSNR values, where MSE reflects the level of error between the original and reconstructed video and PSNR reflects the quality of the reconstruction, together providing a basis for evaluating how well the DWT algorithm preserved video quality through the security and recovery cycle. A system log panel records the activity occurring during processing, including file selection, the encryption or decryption process, processing time, the storage location of the output file, and the success status of the operation, helping the user monitor the system's progress.

Finally, a set of action buttons allows the user to open the folder containing the output file, save a report of the testing results, and reset the application interface to its initial state, supporting convenient management of the processed output. The DWT algorithm itself was implemented using the Daubechies (db4) wavelet at a specified decomposition level to achieve an optimal transformation. Applying this transformation to each frame produced the LL, LH, HL, and HH sub-bands used to obscure the visual content of the video, and this process was repeated across all frames until the entire video had been transformed into a secured, visually unrecognizable form. During encryption, the system read the video using OpenCV, decomposed it into frames, transformed each frame using DWT to produce frequency coefficients, modified those coefficients to obscure the visual content, and reassembled the processed frames into the encrypted output video, displaying a progress bar throughout to indicate processing status. Decryption reversed this sequence, applying the inverse DWT to each frame of the encrypted file to restore its visual structure to a form approximating the original before reassembling the frames into the final, restored video for display within the application. A visual comparison among the original, encrypted, and decrypted video confirmed the intended behavior of the system. The original video retained a clear and easily recognizable appearance, while the encrypted video appeared as noise-like content that could not be identified directly. Decryption using the correct security key then successfully restored the video to a form closely approximating the original, demonstrating that the system could perform both the securing and the restoring of video content effectively using the DWT and IDWT algorithms.

System testing, conducted using black box testing across the application's core functions, confirmed successful operation for video upload, the encryption process, the decryption process, output storage, and quality evaluation of the decrypted video. Testing on three sample video files (Video1.mp4, Video2.mp4, and Video3.mp4) showed that the encryption process succeeded for all three, producing visually encrypted output in each case, and that the subsequent decryption process likewise succeeded for all three, with each resulting file playable afterward. These outcomes indicate that the encryption process substantially altered the visual appearance of the video frames into noise-like content, confirming that the DWT algorithm had been successfully applied as a video file security method, while the decryption results showed that, despite a slight reduction in visual quality, the decrypted videos retained a high degree of similarity to the original. Taken together, the testing demonstrated that the application could carry out its core functions correctly, with the security process successfully rendering the video content unrecognizable and the reconstruction process successfully restoring it to a form close to the original, while the system also produced PSNR, MSE, SNR, and entropy values for evaluating the quality of the reconstructed video.

Table 1. Video Quality Test Results

No	Video Name	Resolution	PSNR	MSE	SNR	Entropy
1	Video mp1.mp4	720p	10.59 dB	5670.15	5.23 dB	7.37
2	Video mp2.mp4	720p	11.04 dB	5116.69	5.28 dB	7.37
3	Video mp3.mp4	720p	10.59 dB	5670.15	5.23 dB	7.37

Quality testing using MSE and PSNR was performed on three 720p video files. The first and third files (Video mp1.mp4 and Video mp3.mp4) produced identical results, with a PSNR of 10.59 dB, an MSE of 5,670.15, an SNR of 5.23 dB, and an entropy of 7.37, while the second file (Video mp2.mp4) produced a PSNR of 11.04 dB, an MSE of 5,116.69, an SNR of 5.28 dB, and an entropy of 7.37. These results show relatively low PSNR values together with comparatively high

MSE values across all three tested videos, indicating that the encrypted videos underwent substantial visual change relative to the originals, such that their content could not be recognized directly. The entropy value of 7.37 observed across all three files further indicates a comparatively high level of data complexity in the encrypted output, suggesting that the DWT-based security process successfully obscured the visual structure of the video. The SNR values likewise point to a substantial change in signal characteristics following the transformation process. Taken together, these findings indicate that the DWT algorithm can function as a method for securing digital video files by rendering their visual content difficult to recognize without the corresponding decryption process.

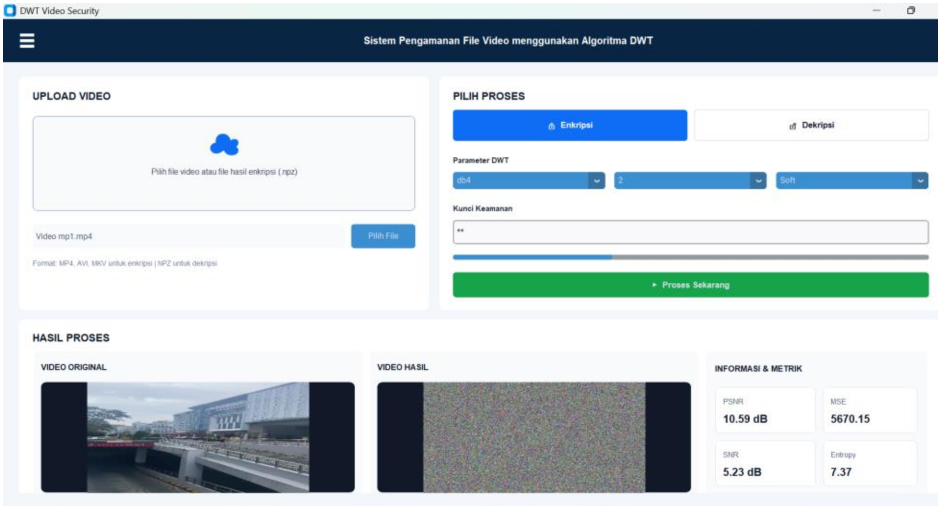

Figure 3. Video Encryption Process Display

The mechanism underlying these results can be illustrated through the worked calculation example using the simple frame matrix described in the methods section, which showed that DWT decomposes data into the LL, LH, HL, and HH sub-bands, that the security key modifies these coefficient values so that the data cannot be reconstructed without the corresponding key, and that applying the correct key during decryption restores the coefficients to their original values for subsequent reconstruction using IDWT. This combination of frequency-domain decomposition and key-based coefficient modification appears to be the basis for the substantial visual obscuring observed in the encrypted test videos, while the corresponding decryption results confirm that the original structure can be recovered once the correct key is supplied.

CONCLUSION

This study has demonstrated the implementation of the Discrete Wavelet Transform (DWT) algorithm for securing digital video files, where the DWT-based security process, applied through frequency-domain transformation of each video frame, successfully altered the visual appearance of the tested videos so that their content became difficult to recognize directly, and testing using the PSNR, MSE, SNR, and entropy parameters confirmed a significant difference between the original and encrypted videos, supporting the conclusion that DWT can meaningfully obscure video content for security purposes; the study further succeeded in producing a complete desktop application, built in Python, capable of carrying out both the encryption process using DWT and the corresponding decryption process using the Inverse Discrete Wavelet Transform (IDWT), with black box testing confirming that the system's core functions operated as intended, and building on these findings, future development could combine the DWT-based approach with other security algorithms to achieve a higher overall level of protection, addressing the inherent limitation that DWT alone does not provide cryptographic-grade security comparable to algorithms such as AES or RSA, while subsequent research could also extend the application to support a wider range of video formats and refine the user interface to be more interactive and accessible for general use.

REFERENCES

Book

[1] Indah Purnama Sari. Algoritma dan Pemrograman. Medan: UMSU Press, 2023, pp. 290.
[2] Janner Simarmata Arsan Kumala Jaya, Syarifah Fitrah Ramadhani, Niel Ananto, Abdul Karim, Betrisandi, Muhammad Ilham Alhari, Cucut Susanto, Suardinata, Indah Purnama Sari, Edson Yahuda Putra. Komputer dan Masyarakat. Medan: Yayasan Kita Menulis, 2024, pp.162.
[3] Mahdianta Pandia, Indah Purnama Sari, Alexander Wirapraja Fergie Joanda Kaunang, Syarifah Fitrah Ramadhani Stenly Richard Pungus, Sudirman, Suardinata Jimmy Herawan Moedjahedy, Elly Warni, Debby Erce Sondakh. Pengantar Bahasa Pemrograman Python. Medan : Yayasan Kita Menulis, 2024, pp.180
[4] Zelvi Gustiana Arif Dwinanto, Indah Purnama Sari, Janner Simarmata Mahdianta Pandia, Supriadi Syam, Semmy Wellem Taju Fitrah Eka Susilawati, Asmah Akhriana, Rolly Junius Lontaan Fergie Joanda Kaunang. Perkembangan Teknologi Informatika. Medan: Yayasan Kita Menulis, 2024, pp.158
[5] Indah Purnama Sari. Buku Ajar Pemrograman Internet Dasar. Medan: UMSU Press, 2022, pp. 300.
[6] Indah Purnama Sari. Buku Ajar Rekayasa Perangkat Lunak. Medan: UMSU Press, 2021, pp. 228.
[7] Purba, A. M. V., Pradipta, R., Zulherry, A., Sari, I. P., Ansya, Y. A. U., Basri, M., ... & Darnilasari, A. (2026). Pengetahuan Dasar Komputer. Yayasan Kita Menulis.
[8] Irfan, A. P., Susanto, C., Sari, I. P., Zulherry, A., Ansya, Y. A. U., Dewi, I. K., ... & Mubarak, R. (2026). Sistem Informasi Modern untuk Era Digital. Yayasan Kita Menulis.
[9] Kaban, R., Rachman, A. I., Sari, I. P., Zulherry, A., Basri, M., Fadilah, N., ... & Ahyuna, A. (2026). Pengantar Artificial Intelligence, Machine Learning, dan Big Data. Yayasan Kita Menulis.

Journal Article

[10] Ahmed, R., & Khan, M. 2023. “Multimedia Data Security using Discrete Wavelet Transform.” International Journal of Multimedia Security, 12(2), 45–53.
[11] Gonzalez, R. C., & Woods, R. E. 2022. Digital Image Processing (4th ed.). Pearson Education.
[12] Jogiyanto. 2017. Analisis dan Desain Sistem Informasi. Yogyakarta: Andi Offset.
[13] Lutz, M. 2023. Learning Python (6th ed.). O’Reilly Media.
[14] Mallat, S. 2022. A Wavelet Tour of Signal Processing. Academic Press.
[15] Munir, R. 2019. Pengolahan Citra Digital dengan Pendekatan Algoritmik. Bandung: Informatika.
[16] Pressman, R. S. 2022. Software Engineering: A Practitioner’s Approach (9th ed.). McGraw-Hill.
[17] Rahman, M., & Islam, S. 2022. “Digital Image Security using Discrete Wavelet Transform.” International Journal of Computer Applications, 15(3), 55–62.
[18] Rosa, A. S., & Shalahuddin, M. 2018. Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek. Bandung: Informatika.
[19] Singh, P., & Sharma, V. 2022. “Wavelet-Based Multimedia Protection Techniques.” Journal of Digital Signal Processing, 19(4), 120–129.
[20] Sutoyo, T., Mulyanto, E., Suhartono, V., Nurhayati, O. D., & Wijanarto. 2009. Teori Pengolahan Citra Digital. Yogyakarta: Andi Publisher.
[21] Zhang, Y., Li, H., & Chen, X. 2023. “Wavelet Transform for Digital Video Processing.” International Journal of Advanced Multimedia Systems, 10(1), 14–25.
[22] Zulherry, A., Sari, I.P., & Basri, M. (2025). Perancangan Aplikasi Monitoring Kehadiran Pegawai Menggunakan RFID. sudo Jurnal Teknik Informatika 4 (4), 378-384
[23] Zulherry, A., & Gunawan, M. (2025). Development of an Android-Based Smart Health Monitoring Device for Heartbeat Detection. *Al'adzkiya International of Computer Science and Information Technology (AIoCSIT) Journal*, *6*(2), 75-79.
[24] Zulherry, A., Ramadhani, F., & Satria, A. (2024). Klasifikasi Data Tracer Study Dengan Pemanfaatan Data Mining Menggunakan Algoritma Support Vector Machine dan Neural Network. Portal Riset dan Inovasi Sistem Perangkat Lunak 2 (1), 45-54
[25] Lubis, I. Q., & Zulherry, A. (2026). Implementasi Keamanan Website Dari Serangan Cross Site Request Forgery Menggunakan Algoritma HMAC-SHA256 Pada Framework Laravel. Hello World Jurnal Ilmu Komputer, 5(1), 47-58.

[26] Asadel, A., & Zulherry, A. (2025). Detecting Zero-Width Characters Obfuscated in Phishing URLs using the XGBOOST Algorithm. Hanif Journal of Information Systems 3 (1), 43-53
[27] Sari, I.P., Zulherry, A., Basri, M., & Hayani W. (2025). Pembelajaran Pemrograman berbasis Machine Learning sebagai Upaya Peningkatan Computational Thinking. Jurnal Penelitian, Pendidikan dan Pengajaran: JPPP 6 (3), 245-250
[28] Sari, I.P., & Batubara, I.H. (2021). User Interface Information System for Using Account Services (Joint Account) WEB-Based. International Journal of Economic, Technology and Social Sciences (Injects) 2 (2), 462-469
[29] Ramadhani, F., & Sari, I.P. (2021). Pemanfaatan Aplikasi Online dalam Digitalisasi Pasar Tradisional di Medan. Prosiding Seminar Nasional Kewirausahaan 2 (1), 806-811
[30] Sari, I.P., & Alfarisi, F. (2024). Perancangan Sistem Aplikasi Pendataan Membership Gym Menggunakan Metode Unified Software Development Process (USDP) Berbasis Web. Hello World Jurnal Ilmu Komputer 3 (1), 37-48
[31] Sari, I.P., Jannah, A., Meuraxa, A.M., Syahfitri, A., & Omar, R. (2022). Perancangan Sistem Informasi Penginputan Database Mahasiswa Berbasis Web. Hello World Jurnal Ilmu Komputer 1 (2), 106-110
[32] Satria, A., Ramadhani, F., & Sari, I.P. (2023). Rancang Bangun Sistem Informasi Penerimaan Peserta Didik Baru (PPDB) Sekolah Menengah Kejuruan Telkom 2 Medan Menggunakan Codeigniter. Wahana Jurnal Pengabdian kepada Masyarakat 2 (1), 23-31
[33] Sari, I.P., Azzahrah, A., Qathrunada, I.F., Lubis, N., & Anggraini, T. (2022). Perancangan sistem absensi pegawai kantoran secara online pada website berbasis HTML dan CSS. Blend sains jurnal teknik 1 (1), 8-15
[34] Hariani, P.P., Sari, I.P., & Batubara, I.H. (2021). Android-Based Financial Statement Presentation Model. JURNAL TARBIYAH 28 (2), 1-16
[35] Sari, I.P., Syahputra, A., Zaky, N., Sibuea, R.U., & Zakhir, Z. (2022). Perancangan sistem aplikasi penjualan dan layanan jasa laundry sepatu berbasis website. Blend sains jurnal teknik 1 (1), 31-37
[36] Sari, I.P., Batubara, I.H., Ramadhani, F., & Wardani, S. (2022). Perancangan Sistem Antrian pada Wahana Hiburan dengan Metode First In First Out (FIFO). Sudo Jurnal Teknik Informatika 1 (3), 116-123
[37] Ramadhani, F., Satria, A., & Sari, I.P. (2022). Aplikasi internet berbasis website sebagai E-Commerce penjualan komponen sport car. Blend Sains Jurnal Teknik 1 (2), 69-75
[38] Sari, I.P., Ramadhani, F., Satria, A., Apdilah, D., & Basri, M. (2023). Rancangan UI/UX Aplikasi Analytics pada Toko Online Wao Sneakers Menggunakan Figma Berbasis Mobile. Factory Jurnal Industri, Manajemen dan Rekayasa Sistem Industri 1 (3), 93-101
[39] Sari, I.P., Al-Khowarizmi, A., & Batubara, I.H. (2021). Cluster Analysis Using K-Means Algorithm and Fuzzy C-Means Clustering For Grouping Students' Abilities In Online Learning Process. Journal of Computer Science, Information Technology and Telecommunication Engineering 2 (1), 139-144
[40] Hutasuhut, B.K., Sari, I.P., & Al-Khowarizmi, A. (2023). Analysis the Effect of Digitalization and Technology on Web-Based Entrepreneurship. Journal of Computer Science, Information Technology and Telecommunication Engineering 4 (1), 350-354
[41] Sitompul, D.N., Rahmatika, A., & Sari, I.P. (2023). Application of The Sales and Purchase Program Using The Rapid Application Development Model. Al'adzkiya International of Computer Science and Information Technology (AIoCSIT) Journal 4 (1), 6-16
[42] Sari, I.P., & Batubara, I.H. (2021). Perancangan Sistem Informasi Laporan Keuangan Pada Apotek Menggunakan Algoritma K-NN. Seminar Nasional Teknologi Edukasi dan Humaniora (SiNTESa) (1).
[43] Ramadhani, F., Satria, A., & Sari, I.P. (2023). Implementasi Metode Fuzzy K-Nearest Neighbor dalam Klasifikasi Penyakit Demam Berdarah. Hello World Jurnal Ilmu Komputer 2 (2), 58-62
[44] Sari, I.P., Batubara, I.H., & Basri, M. (2022). Implementasi Internet of Things Berbasis Website dalam Pemesanan Jasa Rumah Service Teknisi Komputer dan Jaringan Komputer. Blend Sains Jurnal Teknik 1 (2), 157-163
[45] Sari, I.P., & Ramadhani, F. (2021). Pengaruh Teknologi Informasi Terhadap Kewirausahaan Pada Aplikasi Perancangan Jual Beli Jamu Berbasis WEB. Prosiding Seminar Nasional Kewirausahaan 2 (1), 874-878