JSCE (Journal of System and Computer Engineering)
Vol.7, No.2, April 2026, pp. 166-174
ISSN (online): 2723-1240
DOI : https://doi.org/10.61628/jsce.v7i2.2625

Implementation of Tesseract OCR for Automated Payment Validation in E-Commerce

Annisa Salsabila Apriliya Wijaya [1,a]; A Inayah Auliyah [2,b,*]; Jeffry [3,c]; Firman Aziz [4,d]; Syahrul Usman [5,e]

[1] *Information System, Institut Teknologi Bacharudiin Jusuf Habibie, Parepare Indonesia*
[2] *Information System, Institut Teknologi Bacharudiin Jusuf Habibie, Parepare Indonesia*
[3] *Computer Science, Institut Teknologi Bacharudiin Jusuf Habibie, Parepare Indonesia*
[4] *Computer Science, Pancasakti University, Makassar, Indonesia*
[5] *Computer Science, Pancasakti University, Makassar, Indonesia*
[a]*annisasalsabilaaprilyawijaya.221031027@mahasiswa.ith.ac.id;* [b]*Inayah@ith.ac.id ;* [c]*Jeffry@ith.ac.id;* [d]*firman.aziz@unpacti.ac.id ;* [e]*syahrul.usman@unpacti.ac.id*
* *Corresponding author*

Abstract

Indonesia's thriving e-commerce sector has generated an exponential growth in bank transfer transactions, placing considerable strain on manual payment confirmation workflows. This study presents a web-integrated OCR solution developed for SweetJab, a small hijab retailer whose administrative staff were burdened by the time-consuming task of manually reading customer-uploaded payment screenshots. The proposed system harnesses the Tesseract 5.0 engine within a PHP/CodeIgniter environment and incorporates three preparatory image operations—resolution doubling, luminance-based greyscale conversion, and contrast amplification—prior to invoking the LSTM-mode recogniser. Structured transaction data are subsequently isolated from the raw OCR output through a suite of platform-tailored regular expressions covering five major Indonesian payment services: DANA, BRImo, ShopeePay, GoPay, and BYOND BSI. Two rounds of empirical testing were conducted: a baseline evaluation on 40 receipts before system integration achieved 89.5% accuracy, while a second round on 40 post-deployment samples reached 92.5%. All eight Black Box test scenarios passed without functional errors. A key design principle is the 'Admin-in-the-Loop' model, which surfaces extracted data as advisory information to a human verifier rather than triggering automatic payment approvals—thereby balancing operational efficiency with financial security

***Keywords**: Optical Character Recognition, Tesseract, OCR, Payment Validation, E-commerce.*

1. Introduction

The e-commerce sector in Indonesia has undergone a remarkable expansion over the last decade, fuelled by rising internet penetration rates and the mass adoption of mobile devices as the primary gateway to online commerce (Maharsi, 2024). This growth has fundamentally altered purchasing behaviour, shifting transactions away from physical retail towards digital storefronts. Among the payment methods preferred by Indonesian online shoppers, bank transfers remain prevalent—particularly among customers of micro, small, and medium enterprises (MSMEs) that may not be integrated with real-time payment gateways (Fahimah & Harsono, 2023).

A persistent operational challenge for such merchants is the manual reconciliation of incoming bank transfers. In the absence of automated notification systems, store administrators must individually open, read, and cross-reference every payment receipt image submitted by customers—a process that becomes increasingly untenable as sales volume grows. Research has shown that repetitive manual document handling introduces systematic risks: transcription errors in payment amounts, misidentification of the originating account, and verification backlogs that delay order fulfilment and erode buyer trust (Bahar et al., 2023). SweetJab, an online clothing retailer specialising in hijab products and operating on a custom CodeIgniter platform, faced precisely these difficulties before the development described in this paper.

Optical Character Recognition technology offers a well-established pathway for converting document images into machine-readable text at scale. Prior applications in Indonesian financial contexts have demonstrated its utility: Susanty and Setiawan (2024) integrated OCR with string-matching algorithms to automate internal expense reporting; Panggabean et al. (2025) investigated how binarisation preprocessing using Otsu's method can substantially improve OCR legibility for payment receipts; and Bahar et al. (2023) combined Tesseract with regular expressions to achieve high-precision denomination detection in Rupiah banknote imagery. Taken together, these studies affirm that OCR-based extraction is both technically viable and practically beneficial in Indonesian financial document contexts.

Existing studies share a notable limitation, however: they evaluate OCR performance on datasets with highly homogeneous visual characteristics—physical currency, standardised scanned forms, or single-platform receipts—and under conditions far more controlled than those encountered in a real transaction environment. When a merchant must handle receipts generated by multiple distinct banking applications—each with its own colour scheme, font selection, date notation, and account-masking policy—the extraction challenge grows substantially. Furthermore, no prior work has examined the design implications of positioning OCR as an advisory rather than decisional tool: one that informs rather than replaces the human verification step. These unexplored dimensions motivate the present contribution.

This paper makes three original contributions to the literature: (1) a systematic image conditioning pipeline calibrated specifically for mobile banking screenshots sourced from diverse Indonesian applications; (2) a multi-format regex extraction framework that simultaneously handles the heterogeneous receipt structures of DANA, BRImo, ShopeePay, GoPay, and BYOND BSI; and (3) an 'Admin-in-the-Loop' integration model that embeds OCR outputs as structured advisory data within the administrator console, preserving human oversight over final payment decisions. The system is evaluated using Character Error Rate (CER) metrics and Black Box functional tests on a dataset of 80 real-world payment receipts.

2. Method

System development followed a six-stage iterative process: problem analysis of SweetJab's existing verification workflow; targeted literature review covering OCR techniques, digital image conditioning, and financial information systems; architectural planning; feature implementation within the live CodeIgniter/PHP codebase; empirical evaluation; and interpretation of findings. The experimental subject was the bank transfer confirmation module of the SweetJab platform. Receipt images used for testing were obtained through controlled simulations replicating actual customer transactions, spanning eight banking providers: Mandiri, BRI, BNI, DANA, GoPay, OVO, ShopeePay, and BSI.

2.1 Receipt Image Conditioning Pipline

Raw payment receipt images submitted by customers frequently suffer from resolution limitations, colour variation, and non-uniform illumination—conditions that degrade OCR output if left unaddressed. The conditioning pipeline mitigates these issues through three sequential operations. First, a validity check confirms that the submitted file exists on the server and carries a recognised raster format (JPG or PNG), using PHP's native file_exists() and getimagesize() functions. Second, the confirmed image is enlarged to 200% of its submitted dimensions via the imagecopyresampled() function, which applies bicubic interpolation to increase pixel density without blurring glyph boundaries. Third, the upscaled colour image is converted to greyscale through imagefilter() invoking the IMG_FILTER_GRAYSCALE parameter, which internally applies a luminance-weighted channel blend. A contrast intensification pass concludes the pipeline, sharpening the tonal difference between text strokes and background areas—critical for receipts captured as mobile application screenshots where background gradients are common.

2.2 OCR Engine Configuration

Text extraction is handled by Tesseract version 5.0, called from PHP through shell_exec(). The engine operates in LSTM mode (--oem 3), utilising a recurrent neural network trained on diverse typographic styles, which outperforms legacy template-matching recognition on varied receipt fonts. The page segmentation parameter is set to --psm 6, directing Tesseract to treat the input as a single uniform text block rather than attempting column or table detection—appropriate given the linear layout of most payment receipts. Because Indonesian banking applications frequently embed English labels alongside Bahasa Indonesia text, the engine is configured for dual-language processing via the -l ind+eng flag. All recognition is performed on the conditioned image file (processed_ocr.png), not the customer's original upload, ensuring that the preprocessing gains described above are fully leveraged. The resulting character string is retained in the $rawText variable for downstream parsing.

2.3 Regex-Based Transaction Field Extraction

Before pattern matching begins, the raw text string undergoes normalisation: carriage returns are stripped and multiple consecutive whitespace characters are collapsed to a single space. Four transaction attributes are then isolated through custom-designed regular expressions. The transaction date is captured by a pattern recognising day-month-year sequences with abbreviated month tokens (Jan, Feb, ..., Dec). The transfer value is located by scanning for all currency strings prefixed with 'Rp' and selecting the numerically largest match as the principal amount. The recipient account identifier is extracted via bank-specific sub-patterns that accommodate both fully visible and partially masked account strings (e.g., ****7418). The sender's name is isolated using label-based anchors tied to platform vocabulary—for example, 'Account Source' in DANA receipts and 'Sent From' in BRImo or ShopeePay formats. After extraction, the detected recipient account is compared against SweetJab's registered account list via PHP's strpos() function; any discrepancy triggers a red warning banner in the administrator panel.

2.4 Evaluation Design

System quality was assessed through two independent evaluation frameworks. Functional completeness was verified via Black Box Testing, exercising each node of the end-to-end payment confirmation workflow—from initial file upload to the administrator's final status decision—against a defined set of acceptance criteria. Extraction fidelity was measured using the Character Error Rate (CER) index (Rahmadani et al., 2023):

$$CER = (S + D + I) / N \quad (1)$$

where S, D, and I denote the count of substituted, deleted, and inserted characters respectively, and N is the total character count in the reference transcription. Overall system accuracy was then expressed as:

$$Accuracy = (1 - CER\_avg) \times 100\% \quad (2)$$

The pre-integration test drew on 40 diverse receipt samples gathered across multiple payment channels prior to system deployment. The post-integration test used a distinct set of 40 authentic receipts collected directly through the live SweetJab system across the five target platforms.

3. Results And Discussion

3.1 Developmend System Workflow

Integration of the OCR module into the SweetJab CodeIgniter environment proceeded without structural modifications to the existing application. Once deployed, the payment confirmation cycle follows six steps: a customer uploads a receipt image (JPG or PNG) through the order page; the application stores the file server-side and flags the transaction as 'Awaiting Verification'; the administrator opens the transaction record and triggers OCR processing; the system runs the conditioning pipeline and feeds the result to Tesseract; the extracted fields—amount, date, account number, sender name—are rendered as a structured summary beneath the original receipt thumbnail; and the administrator issues a final Accept or Reject decision. The OCR output carries no write permissions over payment status; its exclusive function is to present pre-digested information that accelerates the administrator's own judgment, consistent with the 'Admin-in-the-Loop' principle.

3.2 Black Box Testing Results

Functional testing confirmed that the complete transaction workflow operated without errors across all eight defined test scenarios. Table 1 details the test cases and outcomes.

Table 1. Black Box Testing Results

No	Test Scenario	Input	Expected Output	Actual Output	Status
1	Upload valid receipt	JPG/PNG file	File saved to database	Uploaded successfully	Pass
2	Admin views transaction	Click Detail	Detail page rendered	Correct data displayed	Pass
3	Admin triggers OCR	Click Process OCR	Raw OCR text rendered	Text displayed	Pass
4	Amount extraction success	Clear-text receipt	Amount in extraction field	Displayed correctly	Pass
5	Account mismatch detected	Wrong account no.	Red warning displayed	Red alert shown	Pass
6	Admin confirms payment	Click Accept/Paid	Status → Paid	Updated correctly	Pass
7	Admin declines payment	Click Reject	Status → Rejected	Updated correctly	Pass
8	OCR unreadable image	Blurry/low-res	Field shows 'Unreadable'	Displays 'Unreadable'	Pass

3.3 Baseline OCR Accuracy (Pre-Integration)

Prior to system deployment, a diagnostic assessment was performed on 40 receipt images spanning several banking applications. Among these, 33 yielded character-perfect output (CER = 0), 3 contained one or more recognition errors (CER > 0), and 4 could not be processed at all (CER = 1). Erroneous extractions were most often traceable to ambiguous denomination notation—for example, a BCA receipt showing Rp100,000 was returned as '100' after three characters were dropped, giving CER = 0.5 for that sample. Total failures occurred exclusively on images that were too heavily cropped or too low in resolution for the engine to locate coherent text blocks. The aggregate accuracy calculation is shown below:

$$\text{Sum of CER values} = 1.2 \text{ (partial errors)} + 4.0 \text{ (total failures)} = 5.2$$

$$\text{Mean CER} = 5.2 \div 40 = 0.105$$

$$\text{Baseline Accuracy} = (1 - 0.105) \times 100\% = 89.5\% \quad (3)$$

The breakdown by outcome category appears in Table 2.

Table 2. Summary of Initial OCR Testing Results (40 Samples)

Category	Number of Samples	Percentage (%)
Correct (CER = 0)	33	82.5%
Incorrect (CER > 0)	3	7.5%
Failed (CER = 1)	4	10.0%
Total	40	100%

3.4 Post-Iintegration OCR Accuracy Testing

After full deployment, the assessment was repeated on a fresh set of 40 payment receipts drawn from live SweetJab transactions across the five target platforms. Extraction was fully correct for 37 samples; the remaining 3 contained errors. Table 3 provides a representative sample of individual test results.

Table 3. Representative OCR Extraction Results – Post Integration

No	Platform	Original Amount	OCR Amount	Original Date	OCR Date	Original Destination Account	OCR Destination Account	Status
1	DANA	15,000	15,000	14 Feb 2026	14 Feb 2026	BCA ****7418	7****418	Success
2	DANA	30,000	30,000	10 Feb 2026	10 Feb 2026	081241841958	0812418419****58	Success
3	DANA	25,000	25,000	12 Jan 2026	12 Jan 2026	08979616030	08979616****030	Success
4	DANA	500,500	500,000	10 Feb 2025	10 Feb 2025	BRI ****0535	BRI ****090053	Failed
5	DANA	15,000	15,000	14 Feb 2026	14 Feb 2026	BCA ****7418	7****418	Success
6	DANA	21,000	21,000	09 Feb 2026	09 Feb 2026	BRI ****0507	BRI ****0507	Success
7	DANA	25,000	25,000	09 Feb 2026	09 Feb 2026	081342821349	081342821****349	Success
8	DANA	29,000	29,000	12 Feb 2026	12 Feb 2026	BCA ****9500	9****500	Success
9	DANA	45,000	45,000	23 Dec 2025	23 Dec 2025	081241841958	081241841****958	Success
10	DANA	50,000	50,000	09 Feb 2026	09 Feb 2026	081342821349	081342821****349	Success
11	BRImo	20,000	20,000	09 Jan 2026	09 Jan 2026	081241841958	081241841958	Success
12	BRImo	67,500	65,000	18 Jan 2026	18 Jan 2026	081241841958	081241841958	Failed
13	BRImo	20,000	20,000	31 Dec 2025	31 Dec 2025	081241841958	081241841958	Success
14	BRImo	75,000	75,000	17 Jan 2026	17 Jan 2026	081241841958	081241841958	Success
15	BRImo	30,000	30,000	07 Oct 2025	07 Oct 2025	5017010208533	5017010208533	Success
16	BRImo	102,500	102,500	12 Oct 2025	12 Oct 2025	1784912086	1784912086	Success
17	BRImo	2,202,500	2,202,500	18 Oct 2025	18 Oct 2025	1539744427	1539744427	Success
18	BRImo	200,000	200,000	23 Jun 2025	23 Jun 2025	081241841958	081241841958	Success
19	ShopeePay	23,500	23,500	25 Mar 2025	25 Mar 2025	****6567	****6567	Success

20	GoPay	26,000	26,000	08 Feb 2026	08 Feb 2026	081241841958	081241841958	Success
21	BYOND BSI	3,000,000	3,000,000	12 Feb 2025	12 Feb 2025	95987001221031027	95987001221031027	Success
22	BYOND BSI	510,000	510,000	07 Feb 2026	07 Feb 2026	1000123990	1000123990	Success
23	BYOND BSI	450,000	450,000	10 Jan 2026	10 Jan 2026	1000123990	1000123990	Success
24	BYOND BSI	530,000	530,000	10 Jan 2026	10 Jan 2026	1000123990	1000123990	Success
25	BYOND BSI	1,000,000	1,000,000	13 Feb 2026	13 Feb 2026	7298788351	7298788351	Success
26	BYOND BSI	120,000	120,000	19 Jan 2026	19 Jan 2026	7217386866	7217386866	Success
27	BYOND BSI	2,000,000	2,000,000	21 Jan 2026	21 Jan 2026	7298788351	7298788351	Success
28	GoPay	39,000	39,000	18 Feb 2026	18 Feb 2026	081241841958	081241841958	Success
29	GoPay	26,000	52,000	18 Feb 2026	18 Feb 2026	081241841958	081241841958	Failed
30	GoPay	39,000	39,000	18 Feb 2026	18 Feb 2026	081241841958	081241841958	Success
31	BYOND BSI	750,000	750,000	20 Feb 2026	20 Feb 2026	1000123990	1000123990	Success
32	BYOND BSI	425,000	425,000	20 Feb 2026	20 Feb 2026	7298788351	7298788351	Success
33	ShopeePay	75,000	75,000	20 Feb 2026	20 Feb 2026	95987001221031027	95987001221031027	Success
34	ShopeePay	120,000	120,000	20 Feb 2026	20 Feb 2026	****4452	****4452	Success
35	DANA	85,000	85,000	20 Feb 2026	20 Feb 2026	081342821349	081342821349	Success
36	DANA	150,000	150,000	16 Feb 2026	16 Feb 2026	081241841958	081241841958	Success
37	ShopeePay	40,000	40,000	20 Feb 2026	20 Feb 2026	****8891	****8891	Success
38	BYOND BSI	600,000	600,000	20 Feb 2026	20 Feb 2026	1000123990	1000123990	Success
39	GoPay	26,000	52,000	20 Feb 2026	20 Feb 2026	081342821349	081342821349	Success
40	BYOND BSI	85,000	85,000	22 Feb 2026	22 Feb 2026	1000123990	1000123990	Success

Post-integration accuracy was computed as follows:

$$\text{Accuracy} = (37 \div 40) \times 100\% = 92.5\% \quad (4)$$

$$\text{Residual error rate} = (3 \div 40) \times 100\% = 7.5\% \quad (5)$$

Forensic review of the three failure cases revealed three distinct fault mechanisms. The first involved digit-level confusability: the OCR engine substituted the numeral 2 for 5, inflating a GoPay transfer of IDR 26,000 to IDR 52,000. The second stemmed from image artefacts causing an imprecise denomination boundary, resulting in a DANA receipt's IDR 500,500 being read as IDR 500,000. The third reflected unanticipated layout variations within the BRImo platform that fell outside the scope of the current regex patterns. All three failure types indicate that extraction performance is ultimately bounded by image fidelity and receipt format consistency.

3.5 Comparative Analys with Related Studies

To contextualise the accuracy figures reported above, Table 4 provides a comparative overview of related OCR studies in the Indonesian financial document domain.

Table 4. Accuracy Comparison with Related Studies

Reference	Technique	Application Context	Accuracy (%)
Prananta et al. (2025)	Web-based Tesseract OCR	General text extraction	96.0
Bahar et al. (2023)	OCR + Regular Expression	Banknote denomination	100.0
Hamidah (2022)	OCR + image processing	Banknote denomination	94.0
Rahmawati et al. (2021)	Tesseract, optimal conditions	Digital documents	84-85
This study (2026)	Tesseract + Regex + Preprocessing	E-commerce payment validation	92.5

The 3-point gain between the baseline (89.5%) and post-integration (92.5%) phases directly reflects the benefit of the preprocessing pipeline. By doubling image resolution and optimising tonal contrast before recognition, the system substantially reduced misreadings on low-quality mobile screenshots—a finding consonant with Rahmawati et al. (2021), who demonstrated that image conditioning is the single greatest lever on Tesseract accuracy, and with Panggabean et al. (2025), who confirmed the efficacy of preprocessing specifically for Indonesian digital payment receipts.

The surface-level accuracy gap between this study (92.5%) and the top performers in Table 4 (94–100%) must be interpreted with reference to fundamental differences in dataset composition. Bahar et al. (2023) and Hamidah (2022) worked with physical banknotes—objects characterised by standardised printing, fixed typography, and photographic conditions controlled by the researchers. Prananta et al. (2025) extracted text from uniformly structured web pages, which present none of the visual noise inherent in mobile screenshots. By contrast, this study processed receipts originating from five independent fintech platforms, each generating documents with distinct visual identities: differing font families, colour gradients, date notation conventions (e.g., DD-MM-YYYY versus DD Mon YYYY), and account-masking strategies ranging from full display to partial concealment (e.g., ****7418). These factors collectively increase the parsing challenge far beyond that encountered in single-source studies, making 92.5% a strong result in context.

A methodologically significant distinction also separates this work from all prior studies in Table 4: none of the referenced systems embed OCR within a human-supervised decision workflow. Each prior work treats OCR output as a final answer evaluated against ground-truth labels. The 'Admin-in-the-Loop' architecture introduced here departs from this paradigm by intentionally reserving the ultimate payment decision for a human actor. OCR-extracted fields—amount, date, recipient account, and payer identity—are presented visually alongside the original receipt image, enabling the administrator to confirm, correct, or override the system's interpretation. This design choice yielded two demonstrable safety benefits during testing: in all three cases where OCR produced incorrect values, the platform's account-mismatch alert successfully flagged the anomaly before the administrator acted, preventing any erroneous payment confirmation from reaching the database. Additionally, even when extraction was only partially successful—for instance, the amount was captured correctly but the sender name was unreadable—the partial result still reduced the administrator's inspection workload relative to reviewing the raw image unassisted. This tolerance for imperfect extraction is a practical advantage over fully autonomous verification systems, which typically require higher accuracy thresholds before deployment is considered safe. The approach echoes guidance by Susanty and Setiawan (2024), who argue that combining algorithmic extraction with human oversight is the most prudent architecture for financial document processing in operational settings.

4. Conclusions

This study developed and evaluated a Tesseract OCR-based module for automating the extraction of transaction data from bank transfer receipts submitted to the SweetJab e-commerce platform. The work advances the state of practice in three respects: a systematic image conditioning pipeline tailored to the visual characteristics of mobile banking screenshots across multiple Indonesian applications; a regex extraction framework engineered to handle the diverse receipt formats of five payment platforms simultaneously, including non-standard date notations and partial account masking; and an 'Admin-in-the-Loop' integration model that positions OCR as a productivity aid rather than an autonomous decision-maker, preserving transactional accountability. Functional testing confirmed reliable operation across all eight workflow scenarios. Recognition accuracy improved from 89.5% at baseline to 92.5% following full system integration—a gain attributable to the preprocessing pipeline—and compares favourably with existing literature when dataset heterogeneity is considered. The 'Admin-in-the-Loop' model proved its practical value: automated mismatch detection successfully prompted manual review in every case where OCR produced an erroneous output, preventing incorrect payment statuses from being recorded.Three avenues are recommended for future investigation: enlarging the receipt corpus to include more platforms and a wider range of imaging conditions; replacing the current global contrast enhancement step with adaptive binarisation methods such as Otsu thresholding, which respond to local illumination variation; and evaluating end-to-end deep learning approaches, particularly CNN-LSTM hybrid models, as potential replacements for the Tesseract pipeline on challenging low-resolution inputs.

Acknowledgements

The research team expresses gratitude to the leadership and academic staff of the Department of Information Systems at Institut Teknologi Bacharuddin Jusuf Habibie for their guidance throughout this project. Special thanks are also extended to the management and staff of SweetJab for granting system access and actively supporting the data collection process.

References

Bahar, B., Raban, R. D. Y., & Arnie. (2023). Model pendeteksi nominal uang kertas rupiah menggunakan teknologi optical character recognition. Jurnal Ticom: Technology of Information and Communication, 12(1), 8–13. https://doi.org/10.70309/ticom.v12i1.101

Banu, K., Andreas, D., Anggoro, W., & Setiawan, A. (2023). OCR: Masa depan pengenalan karakter optik dan dampaknya pada kehidupan modern. Jurnal Teknologi Informasi, 9(2), 147–156.

Dijaya, R. (2023). Buku ajar pengolahan citra digital. Penerbit Akademik.

Ente, N. H., Rohandi, M., & Yusuf, R. (2023). Sistem informasi pengarsipan menggunakan optical character recognition berbasis web di kantor notaris dan PPAT. Diffusion: Jurnal Sistem Informasi dan Teknologi, 3(2), 195–203.

Fahimah & Harsono. (2023). Literature review of the evolution of payment system paradigms: From cash to cashless with digital payment. Social, Humanities, and Education Studies Conference Series, 6(3), 11–18.

Hamidah, W. (2022). Deteksi nominal uang kertas menggunakan OCR (Optical Character Recognition). Jurnal Sistem Informasi, 7(2), 72–76.

Hartanto, S., Sugiharto, A., & Endah, S. N. (2012). Optical character recognition menggunakan algoritma template matching correlation. Jurnal Masyarakat Informatika, 5(9), 1–12.

Maharsi. (2024). The growth of digital payments in Indonesia: Harnessing its influence for SMEs and Indonesia's competitiveness. Global South Review, 6(2), 54.

Panggabean, A. W., Lubis, F. H., & Julia, R. R. (2025). Peningkatan keterbacaan teks pada citra struk pembayaran menggunakan segmentasi OTSU. DEVICE: Journal of Information System, Computer Science and Information Technology, 6(1), 155–164.

Prananta, F., Atqiya, M. Z., & Zamhariry, M. F. (2025). Pengembangan aplikasi web untuk pemrosesan gambar dan ekstraksi teks menggunakan Tesseract OCR berbasis web. Jurnal Sistem Informasi Aplikasi Teknologi Informasi, 2(1), 38–44.

Rahmadani, T., et al. (2023). Pengukuran akurasi OCR menggunakan character error rate. Jurnal Rekayasa Sistem Komputer, 6(2).

Rahmawati, A. N., Wibowo, S. A., & Sunarya, U. (2021). Analisis sistem optical character recognition (OCR) pada dokumen digital menggunakan metode Tesseract. Jurnal Informatika Telekomunikasi Elektronika, 8(5), 4777–4785.

Susanty, M., & Setiawan, E. (2024). Implementasi optical character recognition dan text similarity untuk pelaporan uang muka kerja. Jurnal Informatika Universitas Pamulang, 9(4), 187–193.