Journal of Applied Informatics and Computing (JAIC)
Vol.9, No.5, October 2025, pp. 2068~2075
e-ISSN: 2548-6861

Enhancing The Security of E-Invoicing for Distribution Companies Through Image-Based PDF Conversion and QR Verification

Arief Noor Rochmatullah [1]*, Ignatius Aris Wibowo [2]*, Sarwosri [3]*
* Department of Informatics, Institut Teknologi Sepuluh Nopember
6025242005@student.its.ac.id [1], 6025242004@student.its.ac.id [2], sarwosri@its.ac.id [3]

Article Info

Article history:

Received 2025-06-26
Revised 2025-08-02
Accepted 2025-08-10

Keyword:

E-Invoicing,
Distribution Companies,
Security,
Image-based PDF,
QR Code.

ABSTRACT

Distribution companies face significant challenges in securing electronic invoices, as PDF files are susceptible to unauthorized text extraction and manipulation. Prior solutions include SHA-256 digital signatures, and QR code-based verification. There are often require specialized tools, stable internet, or user intervention posing barriers for general trade customers with limited digital access. To address these limitations, this study proposes a hybrid e-invoicing method by converting invoices into image-based PDFs embedded with QR codes. This approach enhances document security, increases resistance to text manipulation, and ensures file sizes remain under 1 MB for smooth distribution via WhatsApp. A dataset of 1000 invoices was tested using OCR and FuzzyWuzzy string similarity to compare extractability between text-based and image-based formats. A composite score was calculated by combining file size and manipulation resistance metrics. Results show that image-based PDFs achieve a significantly higher score (0.595) compared to text-based PDFs (0.005), confirming their superiority in terms of size efficiency and data security. The findings demonstrate that this method provides a robust, low-cost, and scalable solution for secure invoice distribution in environments with limited infrastructure and technical literacy.

I. INTRODUCTION

Advancements in digital technology have driven businesses across various sectors to adopt paperless solutions, including in the invoicing process (*Background*). Distribution companies, managing high transaction volumes of approximately 60,000 customers and 90,000 transactions monthly, face an urgent need to transition from paper-based to electronic invoices (e-invoices). This shift aims to enhance operational efficiency and reduce substantial costs, estimated at tens of millions of rupiah per month, associated with printing, logistics, and storage. However, implementing e-invoicing presents critical challenges, particularly for general trade customers with varying technological literacy and limited access to specialized tools or applications[1]. The primary challenges include ensuring the authenticity of e-invoices to prevent unauthorized manipulation and maintaining small file sizes for easy distribution through widely used platforms like WhatsApp and email, which are preferred by most retail customers.

Unauthorized editing of transaction data by sales staff can have serious financial and legal consequences. Financially, such fraudulent acts lead to inaccurate invoices, resulting in direct revenue loss, overpayments, or customer disputes that can permanently damage trust and profitability. Legally, falsifying tax invoices is a criminal offense that can be subject to imprisonment and fines through several laws and regulations, namely the ITE Law No. 11 of 2008 in conjunction with Law No. 19 of 2016 Article 35, Taxation Law (UU KUP No. 6 of 1983 Jo UU No. 28 of 2007 Article 39 paragraph (1). This can trigger costly audits, significant fines, and potential legal action against the company, while responsible employees can face termination of employment and even criminal prosecution. Discovery of such activities also causes lasting reputational damage, undermining the company's position in the market. These risks underscore the importance of strong security measures, such as cryptographic digital signatures and manipulation-resistant file formats proposed in this study, to ensure data integrity and compliance. Invoice modifications can be used to claim

unauthorized payments, increase invoice values, or create fictitious transactions.

Several solutions have been proposed to enhance PDF document security (*Related Work*). Digital signatures, such as those using SHA-256[2], embed electronic keys to verify document integrity and authenticity[3]. However, their verification often requires specialized tools or platforms (e.g., https://app.digisign.id/validation.html), posing barriers for non-technical users. Another approach involves embedding QR codes linking to the original file on a company server, enabling customers to verify authenticity by scanning[4]. Yet, this method demands active customer participation, which may not be practical for all users. Highlight the trade-offs between security and usability, often increasing file complexity or requiring sophisticated infrastructure.

This research proposes a hybrid e-invoicing system integrating digital certificates, image-based PDFs, and QR codes, implemented via mobile technology (Android and WhatsApp) to address these challenges (*Contribution*). The approach focuses on providing intuitive authenticity verification, enhancing resistance to manipulation, and optimizing file size for mass distribution. It prioritizes cost-effective distribution for companies with dispersed customers and limited technological resources, combining multiple security layers for practical and scalable implementation.

The contributions include (1) a tailored solution for distribution companies with diverse customer bases, (2) emphasis on cost-efficient distribution over expensive server-based solutions, and (3) integration of layered security measures to prevent manipulation effectively.

This article is structured as follows: Section 1 introduces the background, challenges, and contributions of the study. Section 2 details the methodology, including the design of the hybrid e-invoicing system, QR code embedding using the ZXing library, and image-based PDF conversion. Section 3 presents the results and discussion, analyzing the system's performance in terms of text extraction accuracy, file size efficiency, and manipulation resistance using metrics like FuzzyWuzzy and composite scores. Section 4 concludes with key findings and implications for secure e-invoicing in distribution companies.

II. METHODS

This section outlines the proposed method for designing a secure and cost-effective e-invoicing system for distribution companies. The system addresses the challenges faced by Company A, including high paper invoice printing costs (approximately IDR 60 million per month) and the risk of document manipulation by sales personnel. The proposed method integrates three security mechanisms: QR code embedding, text-to-image PDF conversion, and digital signatures, implemented within an Android-based mobile application. The resulting documents are distributed to customers via WhatsApp or email directly from the sales personnel's mobile devices.

As illustrated in Figure 1, the e-invoicing system architecture comprises three components. Sales personnel input transaction data into the mobile application, which sends a request to a server API to generate a secure PDF. The mobile application then receives the PDF file and distributes it to customers through WhatsApp or email.

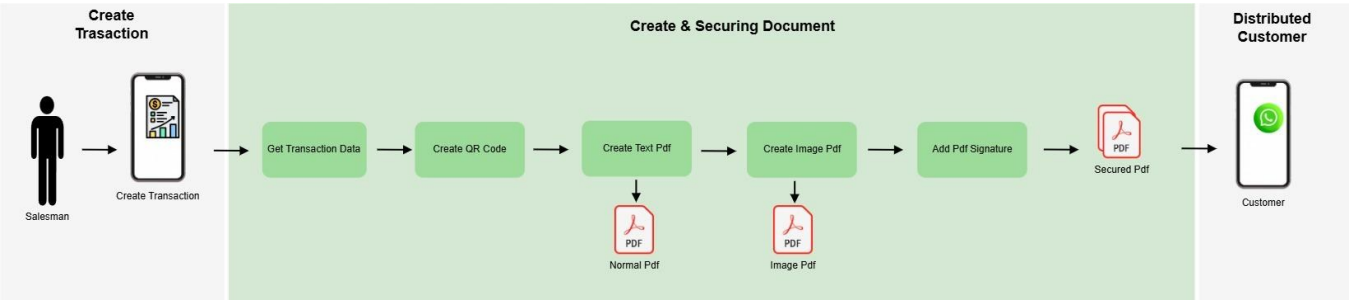

Figure 1. Architecture System

Based on Figure 1, the server-side e-invoicing system follows these steps to create secure PDF documents:

A. Create transaction

The transaction creation process begins with the salesperson, depicted in Figure 1 on the left, using an Android application to record the order from the customer. This process begins when the salesperson receives an order, such as the product details requested by the customer. In the Android application, the salesperson enters the data of the product ordered, such as the product name, quantity, price, or other specifications. This data becomes part of the transaction created.

B. Create & securing Document

Create and Securing Document the data inputted by the salesman is sent to the server. From there, the server API retrieves this data to create a secure document. The process, as shown in the diagram, includes the following steps:

1) QR Code

QR Code technology is used to provide unique identification for each inventory item in the system [5]. Embedding QR code is embedded in the transaction

document using the ZXing library. The QR code contains the transaction ID and a URL. The transaction ID is converted into a unique identifier through a random number generator (RNG), producing an 8-byte string (e.g., b7e1c4d5a9f3g2h8 for transaction ID 20027792), which is mapped to the transaction ID and stored in a database. The use of an 8-byte string is deliberate to ensure the QR code remains scannable and readable by mobile devices. A shorter string length prevents the QR code from becoming overly dense, which could hinder accurate scanning, especially on lower-resolution cameras. Additionally, the randomized 8-byte string enhances security by making the QR code value unpredictable compared to using the original invoice ID, which may follow a sequential or predictable pattern. This reduces the risk of unauthorized access or tampering attempts. Once scanned, customers can verify document authenticity using their mobile device.

Figure. 2. Sample QR Code

The QR code value, for example, is https://servercompany.com/einv/b7e1c4d5a9f3g2h8, as shown in Figure 2. Sample QR Code. The QR code is generated using the library at https://www.npmjs.com/package/qrcode with a 15% error resistance, suitable for electronic documents with standard density. By scanning the QR code customers can verify document authenticity using their mobile devices

2) Creating Text-Based PDF

In process creating text based pdf, transaction data (1) product ID (A unique identifier assigned to each product, ensuring accurate tracking and classification within the invoicing and inventory systems) , (2)purchase quantity (The number of product units purchased in the transaction, which is essential for calculating the total cost and managing inventory),(3) invoice amount (The total monetary value of the transaction, derived from the product of the purchase quantity and unit price, including any applicable taxes or discounts) and the QR code are combined into a single PDF file. The file sizes were recorded in bytes using file system metadata after the PDF generation process. This was done by programmatically reading the file size using functions like os.path.getsize(filepath) in Python. The results are presented in Table 1, comparing file sizes for both formats across several invoice samples.

TABLE 1
PDF FILE SIZE

Id	File Size	
	size_pdftext byte	size_pdfimage byte
1471	167614	148811
1553	192923	193768
1554	160126	136292
1555	157212	130330
1556	162702	142470
1557	145950	113638
1558	149090	115044
1559	174432	166258
1560	149215	114947
1561	160373	132671

3) Conversion text PDF to image PDF

Each page of the generated text-based PDF invoice is converted into a high-resolution static image (PNG format), using image rendering tools. These images are then compiled sequentially into a single PDF file using an image-to-PDF converter. This image-based rendering removes all selectable or extractable text layers, effectively transforming the invoice into a "read-only" visual document. To maintain compatibility with commonly used communication platforms such as WhatsApp and email, the final file size is constrained to under 1,000 KB. This ensures reliable delivery and access by end-users, particularly those with limited internet bandwidth or outdated mobile devices.

Furthermore, this conversion significantly complicates automated content extraction, requiring the use of Optical Character Recognition (OCR) for any text retrieval attempts. OCR is a computer vision technique that converts characters from images into machine-encoded text tools [6], [7]. While

OCR has shown strong performance across varied layouts and fonts, it remains error-prone, especially in compressed or stylized invoice designs. [8] In this paper, we utilize the Pytesseract library to extract text from images into strings, while the PyPDF2 library is used to extract text from PDF documents.

Example invoice rendering and conversion (1) A text-based invoice containing structured data (e.g., "Product ID: PRD-12345", "Qty: 10", "Amount: IDR 150,000") is initially saved as a 150 KB PDF file. After conversion to PNG and recombination into a single PDF, the final file size is 138 KB. The visual layout remains unchanged. When OCR is applied to the image-based PDF, the extracted string reads: "Product ID: PRD-12345, Qty: 10, Amount: IDR 150000" showing that punctuation was lost (missing comma), and date format may shift (e.g., "15 June 2025" becomes "15 Jun 2025"). These small errors indicate a reduction in extractable fidelity, reinforcing manipulation resistance.

The described process therefore adds a security layer by degrading the fidelity of text recognition, thereby increasing the effort and reducing the success rate of unauthorized edits. This aligns with the study's objective to enhance invoice

security without requiring specialized viewing or validation software on the recipient's end.

4) PDF Digital Signature

The PDF digital signature based image and document will be digitally signed using a PDF digital signature with the RSA-2048 algorithm[9], [10]. The process creating the PDF Table 2 begins sequentially by creating an RSA-2048 key pair, which includes a public key and a private key. The document is then hashed using the SHA-256 algorithm to produce a unique digital fingerprint [11]. This hash is signed using the sender's private key, creating a secure digital signature that proves authenticity. The digital signature is embedded and displayed in the PDF file, so users can easily view and verify it. Verification is performed using the sender's public key to confirm that the signature is valid and that the document has not been tampered with. This ensures the integrity of the data and the identity of the signer is as follows:

TABLE 2
DESCRIPTION PDF DIGITAL SIGNATURE

No	Process	Description
1	RSA-2048 key generation	The creation of the digital signature utilizes public-key cryptography, generating a pair of keys: a private key used to sign the PDF file and a public key used to verify the document's authenticity. These keys can be generated using OpenSSL
2	Hassing process with SHA-256	The PDF file is read, converted into a buffer, and hashed using SHA-256 to produce a data digest (e.g., a1b2c3d4e5f6g7h8i9j0)
3	Document signing process with private key	This hash digest is then signed using the private key and embedded in the PDF's metadata
4	Showing signature in pdf file	The PDF also displays information about the signer, as shown in Figure 4. Digital Signature
5	Digital signature verification	The digital signature can be validated by customers online by uploading the file to https://app.digisign.id/validation.html.

RSA-2048 key generation creates a pair of cryptographic keys: a 2048-bit public key and a private key. The public key is shared openly, while the private key is kept secret. These keys are essential for digital signatures, ensuring secure communication and verifying the authenticity of signed documents. SHA-256 Hash Algorithm is a compression operation for the message whose length is less than 264 bits, and the length of output hash value is 256 bits [12]. Even a small change in the data will produce a completely different hash. This process ensures data integrity and is commonly used in digital signatures to securely summarize document contents before signing.

The hash of the document is encrypted using the sender's private key, creating a digital signature. This signature uniquely links the signer to the document's contents. Only the signer's private key can generate this signature, providing authenticity, data integrity, and non-repudiation in digital communications and legal documents. The generated digital signature is embedded in the PDF file. Most PDF readers display a signature panel indicating its validity. This allows recipients to instantly verify whether the document has been modified and confirms the signer's identity, enhancing trust and transparency in digital document handling. Verification is done by decrypting the signature with the sender's public key and comparing it to a freshly computed hash of the document. If both hashes match, the signature is valid. This proves the document's integrity and that it was signed by the legitimate private key holder.

We used SHA-256 over SHA-1 because it offers a higher level of security. SHA-256 produces a 256-bit hash value, which is much more resistant to attacks than SHA-1, which only produces a 160-bit hash. Furthermore, SHA-1 has known security vulnerabilities, making it no longer recommended for applications requiring strong data integrity and security. Therefore, SHA-256 was chosen as a more reliable solution and compliant with modern cryptographic standards.

C. Distributed

In the e-invoice distribution process, PDF documents are distributed through various digital communication channels to ensure easy access for recipients. After the electronic invoice is generated, sales personnel send the document via WhatsApp and email. Distribution through WhatsApp offers a faster and more practical method for delivering e-invoices to customers, This PDF file is not recompressed via WhatsApp, what can be compressed are images. Its real-time nature enables customers to promptly receive, confirm, and follow up on payments based on the received e-invoice.

With these two distribution channels, e-invoices can be sent more flexibly, quickly, and efficiently, aligning with the preferences of recipients, whether for business or individual purposes. Email distribution allows for formal and well-documented delivery, as it can include a subject line, message content, and additional document attachments. Additionally, invoices sent via email can be easily accessed by recipients at any time for administrative or record-keeping purposes.

D. Dataset

The dataset contains information from testing the process of converting transaction data into PDF documents and transforming them into image-based PDFs using various software and programming libraries. It consists of 1000 PDF files from different transaction types, encompassing diverse structures and content such as text, tables, and other visual elements, with varying character and text counts in each file.

TABLE 3.
DESCRIPTION DATASET

No	Name	Description
1	Id	Data row identity (here, only 1 row with Id = 1471).
2	Size_pdftext byte	Text-based PDF file size (in bytes).
3	Size_pdfimage byte	Image-based PDF file size (in bytes).
4	FuzzyWuzzy Results pdf text	Percentage of text successfully extracted from text PDF using FuzzyWuzzy.
5	FuzzyWuzzy Results pdf img	Percentage of text successfully extracted from image PDF using FuzzyWuzzy.
6	Accuracy difference	Difference in extraction accuracy between text and image in FuzzyWuzzy.

Table 4**.** Sample Dataset consists of 1000 entries with 6 main columns, described in Table 3**.** Description Dataset.

TABLE 4
SAMPLE DATASET

Id	File Size		Extract string FuzzyWuzzy Result		
	size_pdftext byte	size_pdfimage byte	FuzzyWuzzy Results pdf text	FuzzyWuzzy Results pdf img	Accuracy difference
1471	167614	148811	99	88	11
1553	192923	193768	99	89	10
1554	160126	136292	100	84	16
1555	157212	130330	99	79	20
1556	162702	142470	99	82	17
1557	145950	113638	100	88	12
1558	149090	115044	99	89	10
1559	174432	166258	99	87	12
1560	149215	114947	99	87	12
1561	160373	132671	99	69	30

E. Text Extraction Ease Analysis with FuzzyWuzzy Levenshtein Distance

To test the hypothesis that image-based PDFs are more difficult to extract text from compared to text-based PDFs, we conducted a comparative experiment. We used 1000 invoice datasets. For each dataset, text extraction was performed on both text-based PDFs and image-based PDFs (using OCR[13]). The similarity between the extracted text and the original text was measured using the FuzzyWuzzy method. [14]

FuzzyWuzzy is a Python library that calculates the similarity between two strings using the *Levenshtein distance,* a metric that counts the number of character changes (insertions, deletions, or substitutions) required to transform one string into another [15][16]. The result is a similarity score ranging from 0 to 100, with higher scores indicating greater similarity FuzzyWuzzy offers several similarity ratios, among which the most common are:

TABLE 5
FUZZYWUZZY CONCEPT

No.	Process	Description
1	Ratio	Measures the overall similarity of strings, giving a score between 0 and 100.
2	Partial Ratio	Useful when one string is a substring of another.
3	Token Sort Ratio	Sorts the tokens (words) in the string before calculating the ratio, so it is not affected by the word order.
4	Token Set Ratio	Uses the set of unique tokens to calculate the ratio, which is great for strings that have a lot of overlapping tokens but also unique tokens.

Table 5**.** Fuzzywuzzy concept and Table 6. Example Fuzzywuzzy concept.

TABLE 6
EXAMPLE FUZZYWUZZY CONCEPT

No.	Process	Description
1	Original text	Total amount due is IDR 150,000 due on 15 June 2025
2	Extracted text pdf string	Total amount due is IDR 150,000 due on 15 Jun 2025
3	Extracted text pdf image	Total amount due is IDR 150000 due on 15 Jun 2025
4	A score from original text compare Extracted text pdf string	Score is 100
5	A score from original text compare Extracted text pdf image	Score is 94

A score of 94 indicates a high degree of similarity, although there are small differences like a missing comma and the shortened month name. In this study, scores close to 100 indicate successful text extraction, while lower scores

(especially in image-based PDFs) signal that the text is harder to extract accurately due to OCR errors. This method helps quantify the difficulty of text extraction and provides an objective metric for comparing text- and image-based PDF formats in terms of their resistance to unauthorized editing.

F. File Size Score

The file size score is calculated using the following formula:

$$File\ Size\ Score = \frac{Size\ Maks - Size}{Size\ Maks - Size\ Min} \quad (1)$$

In Equation 1, the $File\ Size\ Score$ serves as an indicator of file storage efficiency. This score is calculated through normalization, where smaller file sizes yield higher scores. This aims to promote the use of lighter and more efficient files for storage and distribution.

G. Manipulation Resistance Score

The manipulation resistance score is calculated using the following formula:

$$Resistence\ Score\ = 1 - \frac{Accuracy\ OCR}{100} \quad (2)$$

Equation 2, $Resistance\ Score$, is used to assess the extent to which a file is resistant to text-based manipulation, particularly through Optical Character Recognition (ocr) technology. A lower OCR accuracy results in a higher resistance score, indicating that the file is more difficult to extract or modify digitally.

H. Composite Score

$Composite\ Score$ is a combined value that represents the performance or quality of a file based on three main aspects: generation time, file size, and resistance to manipulation. Each aspect is evaluated separately with normalized scores and then combined using specific weights to produce a final value that reflects a comprehensive evaluation of the file. The composite score is calculated as follows:

In Equation 3, the composite score is defined as follows:

$$Comp\ Score = w1 \cdot File\ Size\ Score + w2\ Rest\ Score \quad (3)$$

$Comp\ Score$ are the weights for each parameter, with their sum equaling 1 and equaling 2. The file size score represents the normalized file size, while the manipulation resistance score reflects resistance to manipulation based on OCR accuracy

The evaluation results demonstrate that FuzzyWuzzy provides quantifiable evidence of the difference in text extraction accuracy between file types, making it a reliable tool for assessing the manipulability of document content. By applying normalized scoring, the comparison remains objective and fair across heterogeneous file formats, regardless of their original size or structure. Although image-based PDFs are slightly less accurate in text retrieval due to OCR limitations, they offer enhanced protection against unauthorized modification and are often smaller in size than expected, thanks to efficient image compression. Furthermore, the composite score, which integrates both file size efficiency and manipulation resistance, effectively validates the practicality and security of the proposed method, especially in real-world distribution environments where low bandwidth, high transaction volume, and the risk of content tampering are critical operational concerns.

III. RESULT DAN DISCUSSION

In this testing, we used a sample dataset of electronic invoice data, and the following are the test results. The primary metrics calculated during testing include the average data manipulation time and the average file size

A. Average Text Extraction Accuracy (Data Manipulation)

In the Table 7. Average Data Manipulation displays the average success rate of string extraction using Method 1 on text-based and image-based PDF files. The results show that the success rate for text-based PDFs is consistently high at 99%, while for image-based PDFs, it varies with an average of 81%. This indicates that text extraction from text-based PDFs is far more reliable than from image-based PDFs. Based on the applied FuzzyWuzzy method, the text similarity for image-based PDFs extracted using OCR is approximately 81%, significantly lower than the 99% for text-based PDFs. This substantial decrease demonstrates that converting invoices to image format effectively complicates the text extraction process. Although OCR can extract some text, its accuracy is much lower, resulting in numerous errors or incomplete extractions. A lower similarity score directly indicates that data editing attempts are significantly more challenging and error-prone, thereby substantially enhancing invoice security.

TABLE 7
AVERAGE ACCURACY MANIPULATION FILE

Id	Extract string FuzzyWuzzy Result	
	FuzzyWuzzy Results pdf text	FuzzyWuzzy Results pdf img
1-200	99.4%	83.7%
201-400	99.9%	83.9%
401-600	99.9%	83.3%
601-800	99.9%	83.6%
801-1000	99.9%	83.4%
Average	**99.8%**	**83.6%**

The results in Figure 3. Average accuracy manipulation file indicate that OCR accuracy, corresponding to Equation 1 (manipulation resistance score), reaches 99.8% for text-based PDFs and 83.6% for image-based PDFs. Based on this, the resistance scores are 0.998 and 0.836, respectively. A higher accuracy value indicates that the file is easier to manipulate. Therefore, image-based PDFs demonstrate higher manipulation resistance due to their lower OCR accuracy.

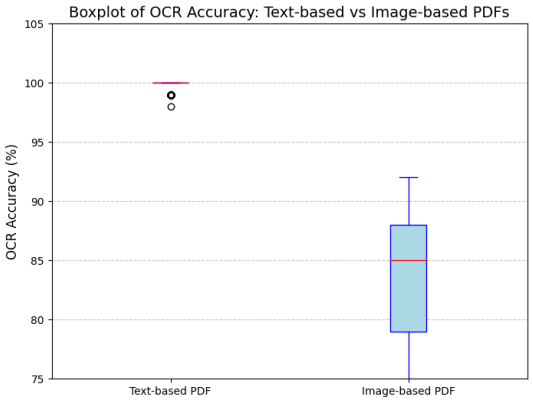

Figure 3. Boxplot Accuracy Manipulation File

B. *Average File Size*

Average File Size in Table 8 shows the average sizes of text-based PDFs (ukuran_pdftext) and image-based PDFs (ukuran_pdfimage) based on ID scale ranges. The results indicate that text-based PDFs have a larger average file size compared to image-based PDFs across all scales. The overall average size for text-based PDFs is 157,279 bytes, while for image-based PDFs, it is 131,081 bytes. This finding suggests that text-based PDFs, despite being visually simpler, result in larger file sizes compared to image-based PDFs. This is likely due to the presence of extensive metadata, text structures, and additional formatting information stored in text-based PDFs, as opposed to image-based PDFs, which primarily consist of a single, cohesive image without multiple structural layers. Invoices of this size still have good image quality for printing and machine reading.

TABLE 8
AVERAGE FILE SIZE

Skala Id	File Size	
	size_pdftext byte	size_pdfimage byte
1-200	157,290	131,103
201-400	157,301	131,139
401-600	157,230	131,011
601-800	157,264	131,059
801-1000	157,309	131,092
Average	**157,279**	**131,081**

The results in Figure 4. File Size Proportion show the distribution of file sizes between text-based and image-based PDFs. From this pie chart, it is evident that text-based PDFs account for 54% of the total file size, while image-based PDFs contribute 46%. This indicates that text-based PDFs tend to have larger file sizes compared to image-based PDFs. Although the difference is not extreme, this proportion suggests that text-based files store more structural data or additional elements compared to simpler image-based files. Therefore, if file size efficiency is a primary priority, using image-based PDFs could be a more optimal choice.

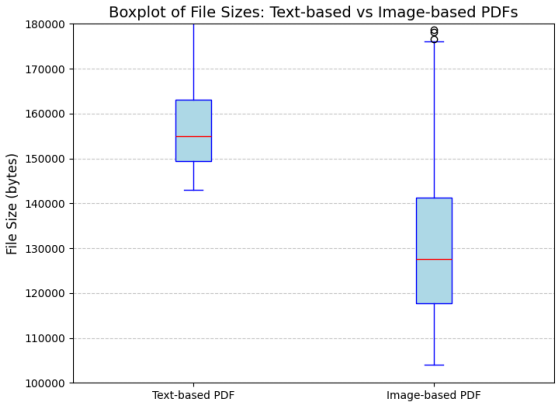

Figure. 4. Proportion File Size

C. *Composite Score Analysis*

The composite score analysis in Table 9 combines two main parameters: the *File Size Score* and the OCR Manipulation *Resistance Score*, each weighted according to its importance

TABLE 9
COMPOSITE SCORE WEIGHTED RESULTS

No.	Process	Description
1	File Size Score	The Image PDF file has the smallest size, which is 131,081 bytes, while the Text PDF file has the largest size, which is 157,279 bytes. Using the normalization formula (3), the Image PDF gets a size score of 1 because it has the smallest size, while the Text PDF gets a score of 0 because it has the largest size. This shows that the Image PDF is much more efficient in terms of size than the Text PDF.
2	Manipulation Resistence Score	The Text PDF file has an OCR accuracy of 99.8%, while the Image PDF file is only 83.6%. Based on formula (1), the Text PDF gets an OCR resistance score of 0.01, indicating that this file is very easy to recognize by OCR and therefore less resistant to manipulation. In contrast, the Image PDF gets an OCR resistance score of 0.19, indicating a higher level of resistance to automatic text detection and extraction.

In this calculation Table 10, as per Equation (2), the weights used are: 0.5 for the *File Size Score* and 0.5 for the OCR *Resistance Score*. The total value of these weights is 1 (0.5 + 0.5), ensuring proportionality in the overall assessment.

TABLE 10
COMPARATIVE ANALYSIS OF RESULTS

Name	Size Score	Manipulation Resistance Score	Composite score
PDF Teks	0	0.01	(0.5 * 0) + (0.5 * 0.01) = 0 + 0.005 = **0.005**.
PDF Image	1	0.19	(0.5 * 1) + (0.5 * 0.19) = 0.5 + 0.095 = **0.595**

From the results of testing 1000 PDF files, with an average size of 1000 PDF text files of 157,279 bytes and 1000 PDF image files of 131,081 bytes, it can be concluded that overall image-based PDF is superior in the evaluated aspects, mainly due to more efficient file size and better resistance to manipulation.

IV. CONCLUSION

Based on the analysis results, there are several significant differences between text-based PDFs and image-based PDFs. In terms of ease of file manipulation (text extraction), text-based PDFs are much easier to extract (99.8% accuracy) compared to image-based PDFs (83.6% accuracy), indicating that text-based structures are more complex to process in the context of unauthorized editing, thus demonstrating better resistance for image-based PDFs. Regarding file size, text-based PDFs have a larger average size compared to image-based PDFs (157,279 bytes versus 131,081 bytes). This is reinforced by the file size proportion results, which show that text-based PDFs account for 54% of the total file size, while image-based PDFs contribute only 46%.

Overall, text-based PDFs are more vulnerable to manipulation (text extraction) and have larger file sizes, whereas image-based PDFs exhibit better manipulation resistance and smaller file sizes. The composite score analysis indicates that image-based PDFs achieve a significantly higher score (0.595) compared to text-based PDFs (0.005), confirming their superiority in terms of size efficiency and data security. The practical implication is that if the priorities are storage efficiency and distribution speed (small file sizes for WhatsApp) as well as document security against unauthorized manipulation, image-based PDFs are the more suitable choice.

Furthermore, the proposed image-based PDF e-invoicing solution has the potential to generate operational cost savings. With the integration of digital signatures and QR codes, the proposed image-based PDF e-invoicing system effectively balances the needs for cost efficiency, transaction security, and customer accessibility, while supporting the operational scalability of distribution companies [17], Risks such as dead links and QR spoofing are not discussed in this article, because the main focus of the research is on the mechanism of converting documents to PDF images and validation via QR code. Future research may explore embedding lightweight verification codes that can be locally validated without requiring server-side queries, or integrating blockchain-based verification to enhance tamper-proofing.

REFERENCES

[1] A. K. Tiwari, Z. R. Marak, J. Paul, And A. P. Deshpande, "Determinants Of Electronic Invoicing Technology Adoption: Toward Managing Business Information System Transformation," *Journal Of Innovation And Knowledge*, Vol. 8, No. 3, Jul. 2023, Doi: 10.1016/J.Jik.2023.100366.

[2] N. Zaatsiyah, "Implementing Digital Signature With Rsa And Md5 In Securing E-Invoice Document," 2021.

[3] M. Cagal, "Evaluating The Usability Of Qualified Electronic Signatures: Systematized Use Cases And Design Paradigms."

[4] S. Suhardi, "Use Of Qrcode And Digital Signature Using The Dsa Method To Authenticate Student Academic Documents," *Journal Of Computer Networks, Architecture And High Performance Computing*, Vol. 6, No. 4, Pp. 1913–1921, Oct. 2024, Doi: 10.47709/Cnahpc.V6i4.4765.

[5] M. Fajar, R. Azhar, Y. Anshori, And R. Laila, "Optimization Of Inventory Management With Qr Code Integration And Sequential Search Algorithm: A Case Study In A Regional Revenue Office," 2025. [Online]. Available: Http://Jurnal.Polibatam.Ac.Id/Index.Php/Jaic

[6] H. T. Ha And A. Horák, "Information Extraction From Scanned Invoice Images Using Text Analysis And Layout Features," *Signal Process Image Commun*, Vol. 102, Mar. 2022, Doi: 10.1016/J.Image.2021.116601.

[7] C. Irimia, F. Harbuzariu, I. Hazi, And A. Iftene, "Official Document Identification And Data Extraction Using Templates And Ocr," In *Procedia Computer Science*, 2022. Doi: 10.1016/J.Procs.2022.09.214.

[8] Z. Zhang, Y. Ding, R. Li, And K. Chen, "Enhancing Ocr With Line Segmentation Mask For Container Text Recognition In Container Terminal," *Eng Appl Artif Intell*, Vol. 133, Jul. 2024, Doi: 10.1016/J.Engappai.2024.108667.

[9] K. Somsuk, "The Special Algorithm Based On Rsa Cryptography For Signing And Verifying Digital Signature," *Heliyon*, Vol. 11, No. 4, Feb. 2025, Doi: 10.1016/J.Heliyon.2025.E42481.

[10] S. Jaiswal, "Signature Encryption Using Blockchain," *Int J Eng Adv Technol*, Vol. 14, No. 2, Pp. 1–5, Dec. 2024, Doi: 10.35940/Ijeat.B4555.14021224.

[11] A. Baihaqi And O. Candra Briliyant, "Implementation Of Rsa 2048-Bit And Aes 128-Bit For Secure E-Learning Web-Based Application," 2017. Accessed: Jun. 19, 2025. [Online]. Available: Https://Ieeexplore.Ieee.Org/Stamp/Stamp.Jsp?Tp=&Arnumber=8272903

[12] J. Wang, G. Liu, Y. Chen, And S. Wang, "Construction And Analysis Of Sha-256 Compression Function Based On Chaos S-Box," *Ieee Access*, Vol. 9, Pp. 61768–61777, 2021, Doi: 10.1109/Access.2021.3071501.

[13] H. T. Ha And A. Horák, "Information Extraction From Scanned Invoice Images Using Text Analysis And Layout Features," *Signal Process Image Commun*, Vol. 102, Mar. 2022, Doi: 10.1016/J.Image.2021.116601.

[14] N. Elmobark, "A Comparative Analysis Of Python Text Matching Libraries: A Multilingual Evaluation Of Capabilities, Performance And Resource Utilization," *International Journal Of Environment, Engineering And Education*, Vol. 7, No. 1, Pp. 48–60, Apr. 2025, Doi: 10.55151/Ijeedu.V7i1.188.

[15] K. Vukatana, "Ocr And Levenshtein Distance As A Measure Of Image Quality Accuracy For Identification Documents," In *International Conference On Electrical, Computer, And Energy Technologies, Icecet 2022*, Institute Of Electrical And Electronics Engineers Inc., 2022. Doi: 10.1109/Icecet55527.2022.9872824.

[16] M. Pikies And J. Ali, "Analysis And Safety Engineering Of Fuzzy String Matching Algorithms," *Isa Trans*, Vol. 113, Pp. 1–8, Jul. 2021, Doi: 10.1016/J.Isatra.2020.10.014.

[17] A. Sharif, D. S. Ginting, And A. D. Dias, "Securing The Integrity Of Pdf Files Using Rsa Digital Signature And Sha-3 Hash Function," In *2021 International Conference On Data Science, Artificial Intelligence, And Business Analytics, Databia 2021 - Proceedings*, Institute Of Electrical And Electronics Engineers Inc., 2021, Pp. 154–159. Doi: 10.1109/Databia53375.2021.9650121.