Jurnal Informatika Vol. 15, No. 1, Juni 2026, Page. 15~28
E-ISSN: 2528-0090 https://ejournal.unidayan.ac.id/index.php/JIU
P-ISSN: 2252-9160 DOI: https://doi.org/10.55340/jiu.v15i1.2684

IMPLEMENTASI TANDA TANGAN DIGITAL BERBASIS KOMBINASI ALGORITMA DIGITAL SIGNATURE ALGORITHM DAN RIVEST SHAMIR ADLEMAN

IMPLEMENTATION OF DIGITAL SIGNATURES BASED ON A HYBRID DIGITAL SIGNATURE ALGORITHM AND RIVEST SHAMIR ADLEMAN SCHEME

Candra Rizki Nanjung Kharisma[*1], Suprapto[2], Alhadi Saputra[3]
[1,2,3]Program Studi Teknik Informatika
Universitas Pelita Bangsa, Indonesia
e-mail: [*1]candrarizki812@gmail.com

Article Info:	Received	Revised	Accepted	Published:
	11 Mei 202)	11 Mei 202)	01 Juni 2026	(01 Juni 2026)

Abstrak

Dokumen digital membutuhkan mekanisme autentikasi dan integritas yang andal untuk memastikan keaslian penandatangan dan ketidakubahan isi setelah distribusi. Penelitian terdahulu umumnya hanya mengandalkan satu algoritma tanda tangan utama (DSA atau RSA secara terpisah) tanpa lapisan verifikasi tambahan pada RSA-PSS, sehingga terdapat gap pada model implementasi yang menggabungkan keduanya untuk mendeteksi pemalsuan isi, penggantian QR Code, dan penggunaan secret code tidak sah dalam satu alur terintegrasi. Selain itu, profil waktu eksekusi komponen kriptografi dan non-kriptografi pada implementasi PDF berbasis desktop belum diukur secara komprehensif. Penelitian ini bertujuan untuk: (1) mengimplementasikan skema tanda tangan digital DSA-RSA-PSS berbasis SHA-256 pada dokumen PDF menggunakan Python; (2) mengukur waktu eksekusi tiap komponen kriptografi (hashing, signing DSA, signing RSA-PSS) dan non-kriptografi (render PDF, deteksi QR Code) pada variasi dokumen 5 hingga 20 halaman; dan (3) mengevaluasi ketahanan sistem terhadap empat skenario keamanan. Dalam alur yang diusulkan, SHA-256 menghasilkan hash dokumen, DSA membentuk tanda tangan utama, dan RSA-PSS menambahkan lapisan verifikasi kedua yang disisipkan ke dalam QR Code. Hasil menunjukkan dokumen asli berhasil diverifikasi, sementara dokumen dengan secret code salah, perubahan isi, atau QR Code yang dimodifikasi berhasil ditolak. Rata-rata waktu hashing 0,0018 detik, signing DSA 0,0014 detik, signing RSA-PSS 0,0028 detik, dan render PDF menjadi beban terbesar sebesar 0,863 detik (±43,27% dari total proses). Kontribusi utama adalah prototipe verifikasi PDF offline yang mengintegrasikan deteksi pemalsuan isi, penggantian QR Code, dan validasi secret code dalam satu alur kerja. Penerapan produksi masih memerlukan integrasi Certificate Authority dan pengelolaan kunci berbasis Public Key Infrastructure.

Kata kunci : *Autentikasi dokumen; DSA; Kriptografi asimetris; PDF; RSA.*

Jurnal Informatika Vol. 15, No. 1, Juni 2026, Page. 15~28
E-ISSN: 2528-0090 https://ejournal.unidayan.ac.id/index.php/JIU
P-ISSN: 2252-9160 DOI: https://doi.org/10.55340/jiu.v15i1.2684

Abstract

Digital documents require reliable authentication and integrity mechanisms to ensure signer legitimacy and content integrity after distribution. Prior studies generally relied on a single primary signature algorithm (DSA or RSA separately) without an additional RSA-PSS verification layer, creating a gap in implementation models that jointly employ both algorithms to detect content forgery, QR Code substitution, and unauthorized secret code usage within a single integrated workflow. Additionally, execution time profiles across cryptographic and non-cryptographic components in desktop-based PDF implementations have not been comprehensively measured. This study aims to: (1) implement a DSA-RSA-PSS hybrid digital signature scheme based on SHA-256 for PDF documents using Python; (2) measure execution time of each cryptographic component (hashing, DSA signing, RSA-PSS signing) and non-cryptographic component (PDF rendering, QR Code detection) across document variations of 5 to 20 pages; and (3) evaluate system resilience against four security scenarios. In the proposed workflow, SHA-256 generates the document hash, DSA produces the primary signature, and RSA-PSS adds a second verification layer embedded in a QR Code. Results show that authentic documents were successfully verified, while documents with an incorrect secret code, modified content, or altered QR Code were rejected. Average hashing time was 0.0018 s, DSA signing 0.0014 s, RSA-PSS signing 0.0028 s, and PDF rendering represented the largest burden at 0.863 s (±43.27% of total process time). The main contribution is an offline PDF verification prototype integrating content forgery detection, QR Code substitution detection, and secret code validation in a single workflow. Production deployment requires Certificate Authority integration and secure Public Key Infrastructure-based key management.

Keywords : *Asymmetric cryptography; Document authentication; DSA; PDF; RSA.*

1. PENDAHULUAN

Transformasi digital menjadikan dokumen elektronik sebagai instrumen utama dalam administrasi pendidikan, bisnis, layanan publik, dan transaksi organisasi. *Portable Document Format* (PDF) banyak digunakan karena mampu mempertahankan tata letak dokumen lintas perangkat, tetapi kemudahan reproduksi dan distribusi digital juga meningkatkan risiko pemalsuan, penyisipan perubahan tanpa izin, serta penyangkalan pengirim. Dalam konteks keamanan informasi, dokumen digital tidak cukup hanya tersimpan secara elektronik; dokumen harus dapat diverifikasi integritasnya, ditelusuri asalnya, dan dibuktikan keterikatannya dengan penanda tangan yang sah. *Digital Signature Standard* (DSS) menyatakan bahwa tanda tangan digital digunakan untuk mendeteksi modifikasi tidak sah, mengautentikasi identitas penanda tangan, dan mendukung *non-repudiation* [1].

Tanda tangan digital berbeda dari gambar tanda tangan manual. Nilai tanda tangan digital bergantung pada isi dokumen, fungsi *hash*, dan kunci privat penanda tangan. Perubahan satu karakter atau bit pada dokumen akan menghasilkan *message digest* yang berbeda, sehingga proses verifikasi dapat menolak dokumen yang telah diubah. Prinsip ini sejalan dengan penelitian berbasis *hashing SHA*-256 pada sistem arsip elektronik, yang menunjukkan bahwa dokumen identik menghasilkan *hash* yang sama, sedangkan perubahan kecil menghasilkan

hash yang sepenuhnya berbeda [2]. Mekanisme *hash* juga menjadi fondasi berbagai penelitian tanda tangan digital berbasis RSA, ECDSA, Schnorr, dan DSA [3], [4], [5].

Digital Signature Algorithm (DSA) merupakan skema tanda tangan digital berbasis kriptografi asimetris yang bergantung pada kesulitan *Discrete Logarithm Problem*. DSA efektif untuk proses *signing* dan *verification*, tetapi kualitas keamanannya sangat bergantung pada parameter *per-message secret number* k yang harus unik dan tidak dapat diprediksi [6]. Penggunaan kata kunci yang lemah atau berulang dapat membuka peluang kompromi terhadap kunci privat. Selain itu, FIPS 186-5 tidak lagi menempatkan DSA sebagai algoritma untuk generasi tanda tangan digital baru, sehingga penggunaannya dalam penelitian ini perlu diposisikan sebagai eksperimen akademik dan pembanding konseptual, bukan sebagai klaim kepatuhan penuh terhadap standar industri terbaru [1], [7]. *Rivest–Shamir–Adleman* (RSA) adalah salah satu skema kriptografi kunci publik yang paling berpengaruh. Rivest, Shamir, dan Adleman memperkenalkan RSA sebagai metode yang memungkinkan kunci enkripsi dipublikasikan tanpa membuka kunci dekripsi, sekaligus menyediakan mekanisme tanda tangan yang dapat diverifikasi secara publik [8]. Kekuatan RSA bergantung pada kesulitan memfaktorkan bilangan komposit yang besar, tetapi biaya komputasinya meningkat seiring bertambahnya ukuran kunci. Beberapa studi menunjukkan bahwa RSA kuat untuk autentikasi dan tanda tangan digital, namun memiliki *overhead* yang lebih besar dibandingkan algoritma berbasis *hash* atau kurva elips dalam beberapa skenario [9], [10], [11].

State of the art dari penelitian ini menunjukkan dua arah utama dalam pengembangan tanda tangan digital. Pertama, penelitian implementatif menggabungkan fungsi *hash*, QR Code, dan algoritma asimetris untuk autentikasi dokumen elektronik, seperti SHA-256 dan RSA pada surat keterangan lulus, Keccak dan RSA pada tanda tangan digital, serta ECDSA dan QR Code untuk dokumen akademik. Meskipun demikian, penelitian-penelitian tersebut umumnya hanya memvalidasi keberhasilan fungsional tanpa mengukur profil performa antarkomponen secara rinci [3], [4], [12], [13]. Kedua, penelitian *hybrid* dan optimasi algoritma berupaya menyeimbangkan keamanan dengan performa, misalnya kombinasi RSA dan DSA, RSA dan AES, atau optimasi RSA menggunakan *Chinese Remainder Theorem* dan modifikasi eksponen [11], [14], [15], [16].

Penelitian terdahulu umumnya hanya mengandalkan satu algoritma tanda tangan utama (DSA atau RSA secara terpisah) tanpa lapisan verifikasi tambahan pada RSA-*Probabilistic Signature Scheme (*PSS*)* , sehingga ada gap pada model implementasi metode DSA dan RSA yang digunakan secara bersamaan untuk mendeteksi adanya pemalsuan tandatangan pada isi dokumen, penggantian QR Code, dan penggunaan *secret code* secara tidak sah dalam satu alur terintegrasi. Selain itu, profil waktu eksekusi antar komponen kriptografi dan non-kriptografi pada implementasi PDF berbasis desktop belum diukur secara komprehensif pada penelitian sebelumnya sehingga menambah gap untuk dilakukan penelitian selanjutnya.

Berdasarkan gap tersebut, penelitian ini bertujuan untuk: (1) mengimplementasikan skema tanda tangan digital DSA-RSA-PSS berbasis SHA-256 pada dokumen PDF menggunakan *Python*; (2) mengukur waktu eksekusi tiap komponen kriptografi (*hashing*, *signing DSA*, *signing RSA*-PSS) serta komponen non-kriptografi (*render* PDF, deteksi QR Code) pada variasi dokumen 5 hingga 20 halaman; dan (3) mengevaluasi ketahanan sistem terhadap empat skenario keamanan, yaitu dokumen asli, *secret code* salah, modifikasi isi PDF, dan modifikasi QR Code. Kebaruan penelitian terletak pada rancangan yang memosisikan DSA sebagai lapisan tanda tangan berbasis *hash*, RSA-PSS sebagai lapisan verifikasi tambahan terhadap paket tanda tangan, serta integrasi QR Code dan metadata PDF dalam aplikasi *desktop Python*. Artikel ini menyajikan formulasi matematis, arsitektur sistem, hasil implementasi aplikasi, grafik performa, serta analisis *trade-off* antara keamanan dan efisiensi komputasi.

2. METODE PENELITIAN

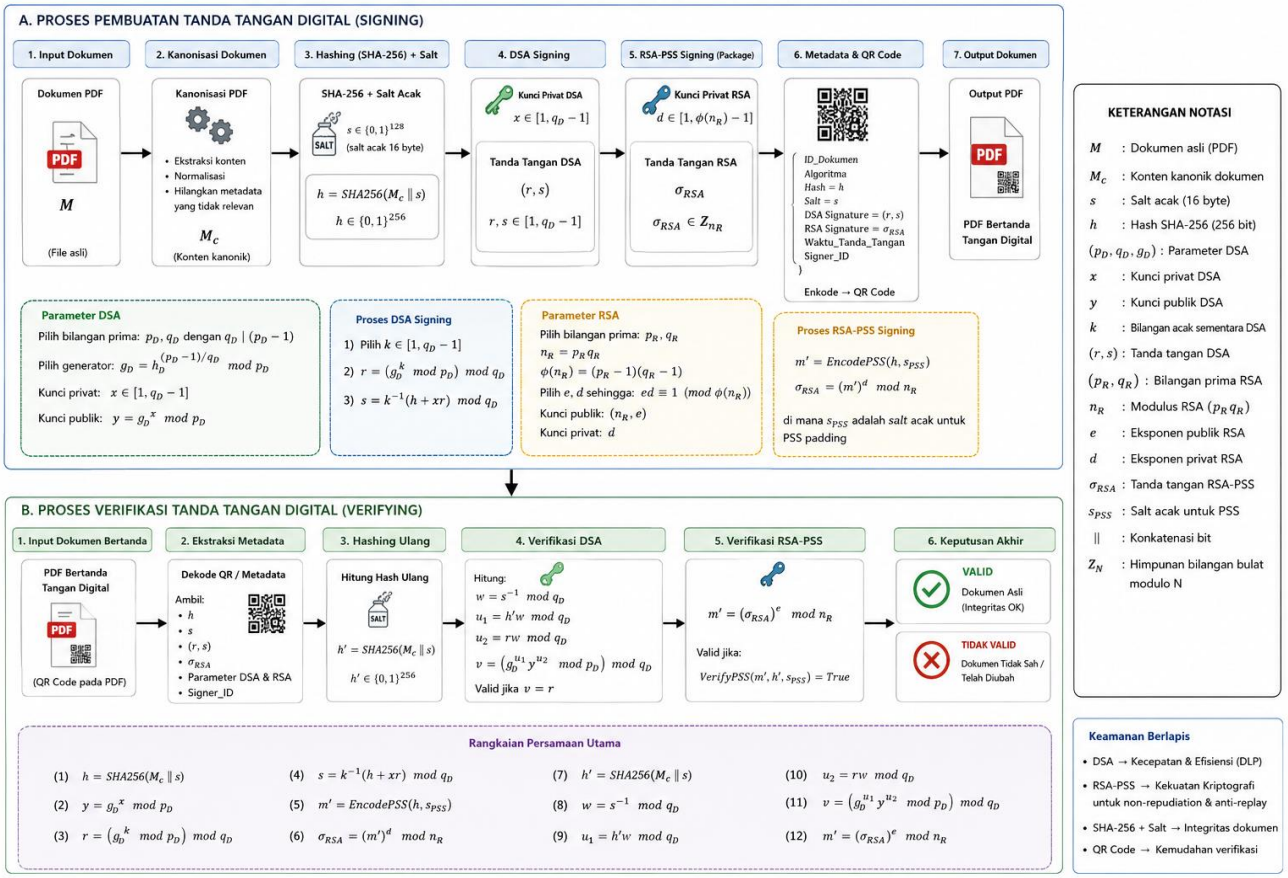
2.1 Desain Penelitian dan Lingkungan Implementasi

Penelitian ini menggunakan pendekatan eksperimen implementatif. Fokus penelitian bukan membuktikan skema kriptografi baru secara formal, melainkan mengimplementasikan dan mengevaluasi integrasi DSA, RSA-PSS, SHA-256, *metadata* PDF, QR Code, dan antarmuka *desktop* dalam satu aplikasi tanda tangan digital. Perangkat pengujian menggunakan laptop dengan prosesor Intel Core i3 generasi ke-11, RAM 8 GB, sistem operasi *Windows* 11, serta Python versi 3.8 atau lebih. Pemilihan ukuran kunci dalam penelitian ini mengacu pada NIST FIPS 186-5 [1], yang merekomendasikan parameter DSA dengan panjang modulus (L) minimal 2048-bit dan panjang subgrup (N) 224- atau 256-bit, serta RSA dengan ukuran kunci minimal 2048-bit untuk keamanan yang memadai. Pada penelitian ini, ukuran kunci yang digunakan adalah DSA 2048-bit/256-bit dan RSA-PSS 2048-bit, sesuai rekomendasi minimum NIST untuk lingkungan akademik dan pengujian prototipe. Pustaka utama yang digunakan adalah cryptography untuk operasi DSA/RSA, *hashlib* untuk *hashing SHA*-256, PyPDF2 untuk *metadata* PDF, ReportLab untuk *overlay QR* Code, pdf2image/Pillow untuk *preview*, pyzbar untuk deteksi QR, dan Tkinter untuk *Graphical User Interface* (GUI).

2.2 Arsitektur Sistem Implementasi DSA-RSA

Arsitektur sistem implementasi DSA-RSA mencakup dua alur kerja utama: penandatanganan (*signing*) dan verifikasi (*verifying*) dokumen PDF. Pada alur penandatanganan, dokumen PDF diproses melalui tahap kanonisasi konten, komputasi *hash* kriptografis, pembentukan tanda tangan digital berlapis menggunakan DSA dan RSA-PSS, kemudian dikemas bersama metadata ke dalam QR Code yang disisipkan pada dokumen keluaran. Pada alur verifikasi, QR Code pada dokumen dibaca dan diekstrak, kemudian divalidasi kesesuaian nilai *hash* dan keabsahan kedua lapisan tanda tangan, sehingga sistem dapat menetapkan status dokumen sebagai valid atau tidak valid. Pendekatan arsitektur berlapis ini memungkinkan autentikasi dokumen yang kuat sekaligus menjaga integritas secara menyeluruh.

Jurnal Informatika
E-ISSN: 2528-0090
P-ISSN: 2252-9160
Vol. 15, No. 1, Juni 2026, Page. 15~28
https://ejournal.unidayan.ac.id/index.php/JIU
DOI: https://doi.org/10.55340/jiu.v15i1.2684



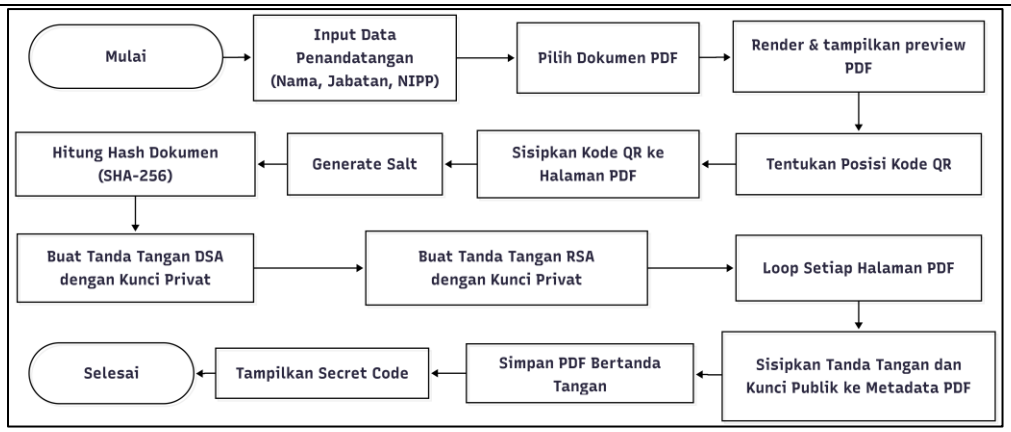
Gambar 1. Arsitektur sistem implementasi DSA-RSA untuk penandatanganan dan verifikasi dokumen PDF

2.3 Formulasi Sistem Sesuai Alur Implementasi

Formulasi matematis dalam penelitian ini dikonstruksi secara sekuensial sesuai dengan *pipeline* implementasi sistem yang sesungguhnya, meliputi: ekstraksi dan *parsing* dokumen PDF, normalisasi representasi konten, komputasi nilai *hash* kriptografis, pembentukan tanda tangan digital menggunakan algoritma DSA, enkapsulasi paket tanda tangan, penerapan lapisan tanda tangan sekunder melalui migrasi skema RSA-Classic ke RSA-PSS (*Probabilistic Signature Scheme*), penyematan QR Code beserta metadata verifikasi, serta eksekusi prosedur verifikasi di sisi penerima. Pendekatan formulasi berbasis alur implementasi ini diterapkan untuk memastikan konsistensi antara representasi matematis dan eksekusi sistem nyata, sekaligus menghilangkan ambiguitas terminologis antara operasi enkripsi RSA konvensional dan mekanisme tanda tangan digital RSA-PSS. [1], [8], [11], [13].

Alur keseluruhan proses penandatanganan (*signing*) dan verifikasi digambarkan secara prosedural pada Gambar 2 untuk memperjelas urutan eksekusi sistem, mulai dari *input* data PDF hingga penyimpanan metadata dokumen.

Jurnal Informatika Vol. 15, No. 1, Juni 2026, Page. 15~28
E-ISSN: 2528-0090 https://ejournal.unidayan.ac.id/index.php/JIU
P-ISSN: 2252-9160 DOI: https://doi.org/10.55340/jiu.v15i1.2684



Gambar 2. Diagram Alir proses penandatanganan PDF menggunakan algoritma DSA dan RSA mulai dari *input* data hingga penyimpanan metadata dokumen

a. Proses pembuatan tanda tangan digital

Dokumen PDF dimasukkan ke dalam format normalisasi konten kanonik agar objek dinamis yang muncul setelah penandatanganan, seperti QR Code dan metadata, tidak ikut mengubah nilai verifikasi *hash*. Sistem kemudian menghasilkan *salt* acak ρ berukuran 16 byte dan menghitung *digest* SHA-256 sebagaimana pada (1).

$$h = \text{SHA-256}(M_c \parallel \rho),\ \rho \in \{0,1\}^{128} \quad (1)$$

Kunci publik DSA dibentuk dari parameter domain pD, qD, dan gD serta kunci privat xD. Relasi kunci publik DSA ditunjukkan pada (2).

$$y_D = {g_D}^{x_D} \bmod p_D \quad (2)$$

Pada tahap *signing DSA*, sistem memilih bilangan acak *per-message* kD yang unik dan bersifat rahasia. Nilai rD dan sD dihitung menggunakan (3) dan (4). *Digest* h terlebih dahulu dikonversi menjadi bilangan bulat z = int(h) mod qD agar kompatibel dengan operasi modular DSA.

$$r_D = \left({g_D}^{k_D} \bmod p_D\right) \bmod q_D \quad (3)$$

$$s_D = {k_D}^{-1}(z + x_D r_D) \bmod q_D,\ z = \text{int}(h) \bmod q_D \quad (4)$$

Lapisan RSA digunakan sebagai tanda tangan tambahan berbasis RSA-PSS pada paket tanda tangan, bukan untuk mengenkripsi dokumen. Paket P memuat *hash* dokumen, *salt*, tanda tangan DSA, kunci publik DSA, identitas penanda tangan, *timestamp*, serta *digest secret code*. *Digest* paket dan *encoding* PSS dirumuskan dalam (5).

$$EM = \text{EncodePSS}(\text{SHA-256}(P), \rho_R) \quad (5)$$

Tanda tangan RSA-PSS terhadap paket P dihasilkan dengan eksponen privat dR dan modulus nR, sebagaimana pada (6).

$$\sigma_{RSA} = EM^{d_R} \bmod n_R \quad (6)$$

b. Proses verifikasi tanda tangan digital

Jurnal Informatika Vol. 15, No. 1, Juni 2026, Page. 15~28
E-ISSN: 2528-0090 https://ejournal.unidayan.ac.id/index.php/JIU
P-ISSN: 2252-9160 DOI: https://doi.org/10.55340/jiu.v15i1.2684

Pada proses verifikasi, sistem mengekstraksi metadata dan QR Code, membentuk kembali konten kanonik dari dokumen yang diterima, lalu menghitung ulang *digest* dokumen menggunakan *salt* yang sama. Proses ini dinyatakan pada (7).

$$h_v = \text{SHA-256}(M_{cv} \parallel \rho),\ z_v = \text{int}(h_v)\ mod\ q_D \tag{7}$$

Verifikasi DSA dilakukan dengan menghitung invers modular dari sD, dua nilai antara u1 dan u2, serta nilai pembanding vD. Rangkaian verifikasi DSA dinyatakan dalam (8) sampai (11).

$$w = {s_D}^{-1}\ mod\ q_D \tag{8}$$

$$u_1 = z_v w\ mod\ q_D \tag{9}$$

$$u_2 = r_D w\ mod\ q_D \tag{10}$$

$$v_D = ({g_D}^{u_1} {y_D}^{u_2}\ mod\ p_D)\ mod\ q_D, valid\ jika\ v_D = r_D \tag{11}$$

Lapisan RSA-PSS diverifikasi menggunakan kunci publik RSA. Secara konseptual, pembukaan tanda tangan RSA dinyatakan pada (12), kemudian hasilnya divalidasi terhadap *encoding* PSS dan *digest* paket P.

$$EM_v = {\sigma_{RSA}}^{e_R}\ mod\ n_R, valid\ jika\ \text{VerifyPSS}(EM_v, \text{SHA-256}(P), \rho_R) = 1 \tag{12}$$

Perlu dicatat bahwa pada implementasi prototipe ini, verifikasi kriptografis belum sepenuhnya mencakup dua aspek berikut: (1) pengukuran *avalanche effect* pada tingkat bit yang mengkuantifikasi persentase bit *digest* yang berubah akibat modifikasi satu karakter pada dokumen, dan (2) validasi kunci publik melalui rantai sertifikat *Certificate Authority* (CA) yang mengaitkan identitas penandatangan dengan kunci publik secara resmi. Keterbatasan ini disebabkan oleh cakupan penelitian yang difokuskan pada validasi fungsionalitas dan profil performa sistem prototipe, bukan pada kepatuhan penuh terhadap infrastruktur kunci publik produksi. Implementasi *Key Public Infrastructure* penuh dan pengukuran *avalanche effect* yang direkomendasikan sebagai tahapan pengembangan berikutnya.

Keputusan akhir sistem tidak ditambahkan sebagai persamaan baru agar jumlah persamaan inti tetap konsisten dengan skema. Secara logis, dokumen dinyatakan valid apabila empat kondisi terpenuhi secara simultan: *hash* dokumen hasil verifikasi sama dengan *hash* referensi, tanda tangan DSA valid, tanda tangan RSA-PSS valid, dan *digest secret code* yang dimasukkan pengguna sesuai dengan *digest secret code* pada metadata. Dengan susunan ini, DSA berperan dalam pembentukan tanda tangan awal terhadap *digest* dokumen, RSA-PSS berperan sebagai lapisan tanda tangan tambahan terhadap paket DSA dan metadata verifikasi, QR Code berfungsi sebagai kanal identifikasi dan validasi cepat, sedangkan SHA-256 menjadi basis pemeriksaan integritas dokumen. Urutan tersebut selaras dengan prinsip tanda tangan digital berbasis *hash*, kunci privat untuk *signing*, dan kunci publik untuk *verification* sebagaimana dijelaskan dalam standar *Digital Signature Standard* [1], implementasi RSA [11], [13], [15], serta implementasi DSA pada pemeriksaan integritas dokumen [7], [17], [18].

2.4 Parameter Pengujian

Pengujian dilakukan pada dokumen PDF dengan variasi jumlah halaman 5, 10, 15, dan 20 halaman. Parameter yang diukur meliputi waktu *render* PDF, waktu *hashing SHA*-256, waktu *signing DSA*, waktu *signing RSA*-PSS, dan waktu deteksi QR Code. Pengujian fungsional dilakukan dengan *black-box testing* pada fitur pemilihan PDF, *preview dokumen*, penempatan QR Code, penandatanganan, verifikasi *secret code*, verifikasi dokumen asli, serta verifikasi dokumen yang dimodifikasi. Untuk pengembangan lanjutan, parameter yang

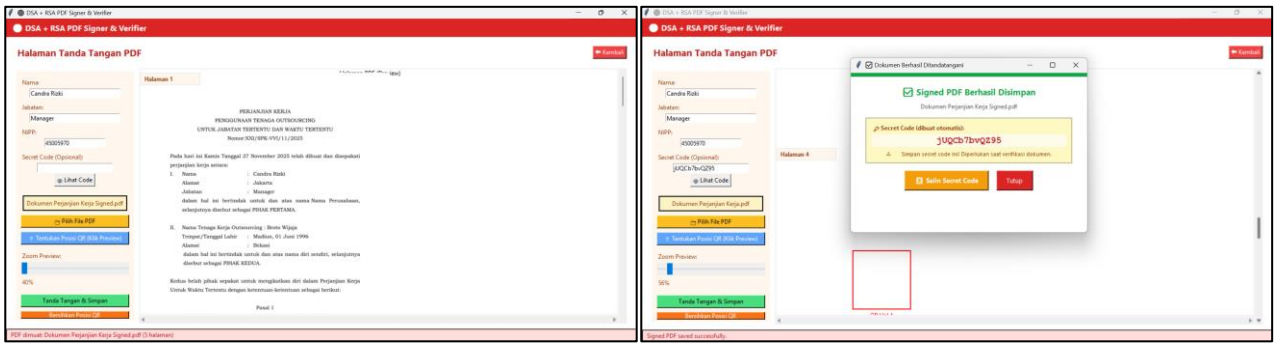
Jurnal Informatika
E-ISSN: 2528-0090
P-ISSN: 2252-9160
Vol. 15, No. 1, Juni 2026, Page. 15~28
https://ejournal.unidayan.ac.id/index.php/JIU
DOI: https://doi.org/10.55340/jiu.v15i1.2684

disarankan meliputi ukuran file 1 MB, 5 MB, 10 MB, 25 MB, dan 50 MB serta variasi kunci 1024-bit dan 2048-bit agar pengaruh ukuran file dan panjang kunci dapat dipetakan secara lebih ketat.

3. HASIL DAN PEMBAHASAN

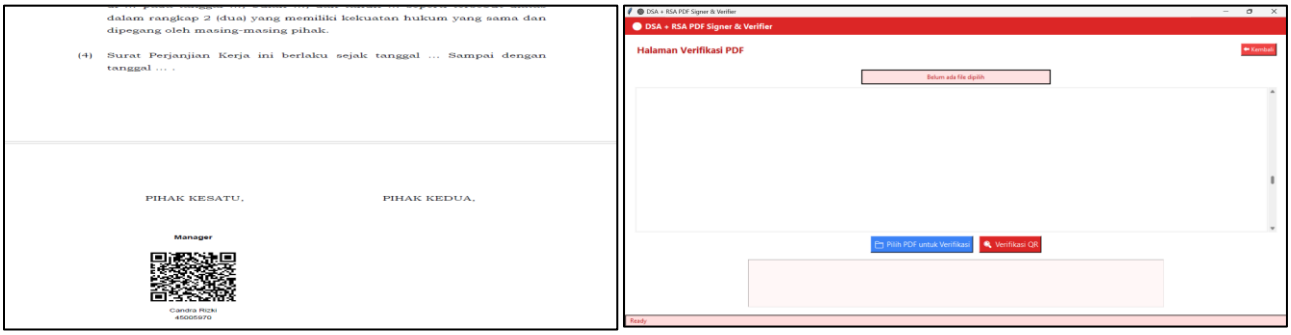
3.1 Hasil Implementasi Aplikasi

Aplikasi berhasil diimplementasikan sebagai aplikasi *desktop* berbasis Python. Halaman utama menyediakan dua fungsi inti, yaitu penandatanganan dan verifikasi PDF. Antarmuka ini dirancang untuk meminimalkan kompleksitas kriptografi bagi pengguna akhir. Halaman penandatanganan menyediakan formulir identitas penandatangan, pemilihan PDF, pratinjau dokumen, penentuan posisi kode QR, serta tombol eksekusi tanda tangan. Uji implementasi menunjukkan bahwa sistem mampu memuat dokumen dan menempatkan QR Code sesuai koordinat yang dipilih pengguna sebagaimana ditunjukkan pada gambar 3.



Gambar 3. Tampilan halaman penandatanganan dan notifikasi keberhasilan penandatanganan dan penyisipan kode QR pada dokumen PDF

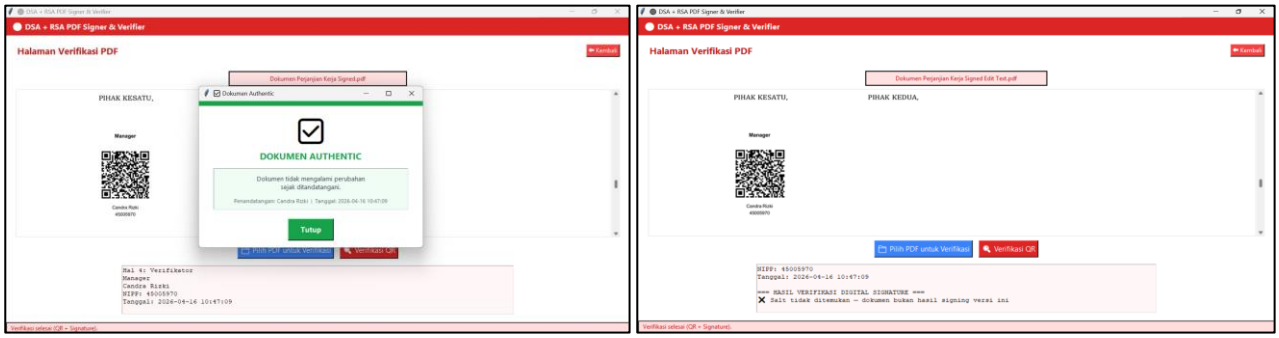
Setelah proses *signing* selesai, QR Code ditempatkan pada halaman PDF dan metadata tanda tangan disimpan bersama informasi kriptografis. Gambar 4 menampilkan contoh dokumen yang telah memuat QR Code sebagai penanda visual dan media bantu verifikasi dan menampilkan halaman verifikasi PDF sebelum proses validasi QR dan *signature package*.



Gambar 4. Dokumen PDF hasil penandatanganan dan halaman verifikasi PDF sebelum proses validasi QR dan *signature package*.

Halaman verifikasi memungkinkan pengguna memilih dokumen PDF yang bertanda tangan, membaca QR Code, memasukkan *secret code*, serta menampilkan status verifikasi. Hasil implementasi menunjukkan dua keluaran utama, yaitu dokumen autentik ketika *hash* dan data validasi sesuai, serta verifikasi gagal ketika *secret*

code salah atau dokumen telah berubah. Contoh keluaran verifikasi ditampilkan pada gambar 5 yang menunjukkan hasil verifikasi berhasil, dan menunjukkan hasil verifikasi gagal pada dokumen.



Gambar 5. Hasil verifikasi berhasil pada dokumen dan hasil verifikasi gagal pada dokumen

3.2 Pengujian Fungsional

Pengujian fungsional dilakukan menggunakan pendekatan *black-box testing*. Tujuannya adalah memastikan setiap fungsi utama menghasilkan keluaran sesuai dengan rancangan tanpa menilai struktur kode internal. Ringkasan hasil pengujian ditunjukkan pada Tabel 1.

Tabel 1. Ringkasan hasil pengujian fungsional aplikasi

No	Rancangan proses	Hasil yang diharapkan	Hasil pengujian	Status
1	Halaman utama tampil dan navigasi aktif	Dua tombol utama dan *status bar* tampil	Sesuai harapan	Valid
2	Pemilihan dan *render* file PDF	PDF dimuat dan *preview* tampil	Sesuai harapan	Valid
3	Penentuan posisi QR Code	Koordinat QR tersimpan dan *overlay* terbentuk	Sesuai harapan	Valid
4	*Hashing*, DSA *signing*, dan RSA-PSS layer	*Signature package* tersimpan pada metadata PDF	Sesuai harapan	Valid
5	Verifikasi dokumen asli	Dokumen dinyatakan *authentic*	Sesuai harapan	Valid
6	Verifikasi *secret code* salah atau dokumen berubah	Sistem menolak verifikasi	Sesuai harapan	Valid

3.3 Pengujian Performa Komputasi

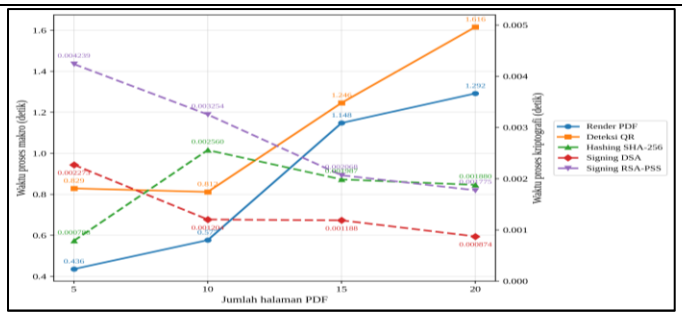
Pengujian performa menunjukkan bahwa komponen kriptografi memiliki waktu eksekusi yang sangat kecil dibandingkan dengan *render* PDF dan deteksi QR Code. Hasil pengukuran waktu ditunjukkan pada Tabel 2.

Tabel 2. Waktu eksekusi berdasarkan jumlah halaman PDF

Jumlah halaman	*Render* PDF (s)	*Hashing* (s)	*Signing DSA* (s)	*Signing RSA*-PSS (s)	Deteksi QR (s)
5	0,436000	0,000788	0,002273	0,004239	0,829000
10	0,577000	0,002560	0,001204	0,003254	0,812000
15	1,148000	0,001987	0,001188	0,002068	1,246000
20	1,292000	0,001880	0,000874	0,001775	1,616000

Gambar 6 memperlihatkan bahwa waktu *render* dan deteksi QR Code meningkat seiring bertambahnya jumlah halaman. Sebaliknya, waktu *hashing* dan *signing* tetap berada pada skala milidetik. Artinya, *overhead* kriptografis DSA-RSA bukan faktor dominan dalam performa total aplikasi.

Jurnal Informatika
E-ISSN: 2528-0090
P-ISSN: 2252-9160
Vol. 15, No. 1, Juni 2026, Page. 15~28
https://ejournal.unidayan.ac.id/index.php/JIU
DOI: https://doi.org/10.55340/jiu.v15i1.2684



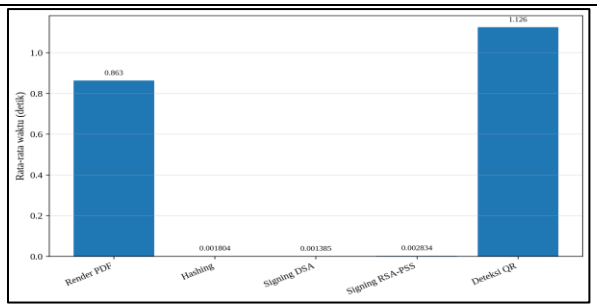
Gambar 6. Profil waktu eksekusi berdasarkan jumlah halaman PDF

Gambar 6 menunjukkan bahwa peningkatan waktu eksekusi pada jumlah halaman PDF berpengaruh lebih besar terhadap proses *render* dokumen dan deteksi QR Code dibandingkan dengan proses kriptografi. Waktu *render* PDF meningkat dari 0,436 detik pada dokumen 5 halaman menjadi 1,292 detik pada dokumen 20 halaman, sedangkan waktu deteksi QR meningkat dari 0,829 detik menjadi 1,616 detik. Kenaikan tersebut wajar karena semakin banyak halaman yang diproses, semakin besar pula kebutuhan sistem untuk membaca, menampilkan, dan memindai elemen visual dalam dokumen. Sebaliknya, proses *Hashing SHA*-256, *Signing DSA*, dan *Signing RSA*-PSS menunjukkan waktu eksekusi yang sangat singkat, yaitu pada orde milidetik hingga submilidetik. Temuan ini mengindikasikan bahwa *bottleneck* utama sistem tidak terletak pada algoritma kriptografi, melainkan pada pemrosesan dokumen PDF dan deteksi kode QR. Dengan demikian, kombinasi DSA dan RSA-PSS dapat diterapkan secara efisien karena mampu menambah lapisan keamanan tanpa meningkatkan waktu komputasi secara signifikan.

Tabel 3. Rata-Rata Waktu Eksekusi dan Kontribusi Tiap Komponen Sistem DSA–RSA

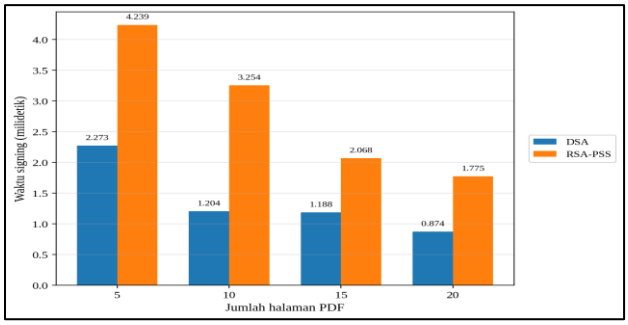
Komponen	Rata-rata Waktu	Kontribusi terhadap Total
Render PDF	0,863 detik	±43,27%
Hashing SHA-256	0,001804 detik	±0,09%
Signing DSA	0,001385 detik	±0,07%
Signing RSA-PSS	0,002834 detik	±0,14%
Deteksi QR	1,126 detik	±56,43%

Berdasarkan grafik rata-rata waktu kontribusi tiap komponen, proses deteksi QR memiliki waktu eksekusi tertinggi, yaitu 1,126 detik atau sekitar 56,43% dari total waktu proses rata-rata. Komponen berikutnya adalah *render* PDF dengan rata-rata 0,863 detik atau sekitar 43,27%. Sementara itu, komponen kriptografi menunjukkan waktu eksekusi yang sangat rendah, yaitu 0,001804 detik untuk *hashing SHA*-256, 0,001385 detik untuk *signing DSA*, dan 0,002834 detik untuk *signing RSA*-PSS. Secara keseluruhan, proses kriptografi hanya berkontribusi sekitar 0,30% terhadap total waktu eksekusi, sedangkan *render* PDF dan deteksi QR berkontribusi sekitar 99,70%. Temuan ini menunjukkan bahwa penambahan lapisan keamanan melalui kombinasi DSA dan RSA-PSS tidak memberikan beban komputasi yang signifikan terhadap sistem[15], [17]. Dengan demikian, optimasi lanjutan sebaiknya difokuskan pada peningkatan efisiensi proses *render* PDF dan deteksi QR Code, bukan pada algoritma kriptografi inti.



Gambar 7. Rata-rata kontribusi waktu untuk tiap komponen pemrosesan.

Perbandingan khusus antara *signing DSA* dan RSA-PSS ditunjukkan pada Gambar 7. RSA-PSS lebih lambat daripada DSA pada seluruh skenario, tetapi selisih absolutnya tetap kecil. Rata-rata tambahan waktu RSA-PSS sekitar 0,0028 detik, sehingga lapisan keamanan kedua tidak menimbulkan *overhead* yang besar pada dokumen uji hingga 20 halaman. Hasil ini konsisten dengan literatur yang menyatakan bahwa RSA lebih berat dibandingkan beberapa skema lain, tetapi masih layak digunakan ketika ukuran dokumen dan frekuensi *signing* berada pada skala aplikasi administratif [9], [10].



Gambar 8. Perbandingan waktu *signing DSA* dan RSA-PSS.

3.4 Pengujian Skenario Keamanan

Pengujian keamanan dilakukan melalui 4 skenario: dokumen asli, *secret code* salah, modifikasi isi PDF, dan modifikasi QR Code. Tujuan pengujian adalah mengevaluasi kemampuan sistem dalam membedakan dokumen yang valid dan yang tidak valid. Ringkasan hasil pengujian ditunjukkan pada Tabel 4.

Table 4. Ringkasan uji skenario keamanan

Skenario	Ekspektasi	Hasil implementasi	Interpretasi
Dokumen asli dan *secret code* benar	Valid	Dokumen *authentic*	*Hash*, QR, dan metadata sesuai
Secret code salah	Gagal	Verifikasi ditolak	Kontrol akses verifikasi bekerja
Isi dokumen dimodifikasi	Gagal	*Hash* tidak sesuai	Integritas dokumen terdeteksi berubah
QR Code dimodifikasi/rusak	Gagal	Data QR tidak tervalidasi	Penanda visual tidak cocok dengan paket verifikasi

Hasil tersebut menunjukkan bahwa properti *avalanche* pada *hash* kriptografis berperan penting dalam mendeteksi perubahan. Pada tingkat konseptual, perubahan kecil pada dokumen menyebabkan *digest* baru yang tidak sama dengan *digest* referensi, sehingga proses verifikasi gagal. Namun, penelitian ini belum menghitung *avalanche effect* secara *bit-level*, misalnya persentase *bit digest* yang berubah setelah satu karakter pada dokumen dimodifikasi. Pengukuran tersebut disarankan sebagai pengujian lanjutan agar ketahanan *hash* dapat dinilai secara kuantitatif, bukan hanya berdasarkan status valid atau tidak valid.

3.5 Analisis Keamanan dan *Trade-off*

Skema *hybrid* DSA dengan RSA-PSS memiliki dua keuntungan utama. Pertama, DSA memberikan mekanisme tanda tangan digital berbasis *Discrete Logarithm Problem* (DLP). Secara matematis, kekuatan DSA bertumpu pada kesulitan komputasional menyelesaikan DLP: diberikan nilai yD = gD^xD mod pD, tidak ada algoritma polinomial yang diketahui mampu menghitung kunci privat xD dari yD, gD, dan pD secara efisien ketika parameter domain dipilih sesuai rekomendasi NIST FIPS 186-5 [1]. Kedua, RSA-PSS menambahkan lapisan tanda tangan probabilistik yang mengurangi risiko determinisme pada keluaran dan memperkuat validasi paket tanda tangan. Kekuatan RSA-PSS bertumpu pada kesulitan faktorisasi bilangan komposit besar: diberikan modulus nR = pR × qR (hasil perkalian dua bilangan prima besar), tidak ada algoritma yang diketahui mampu memfaktorkan nR dalam waktu polinomial. Dengan ukuran kunci 2048-bit, waktu faktorisasi menggunakan GNFS diperkirakan melampaui kapasitas komputasi yang tersedia saat ini, sehingga kunci privat dR tetap aman selama pR dan qR dijaga kerahasiaannya [8]. Dalam skenario implementasi, penyerang tidak cukup hanya mengganti QR Code atau metadata karena sistem tetap menghitung ulang *hash* dokumen dan memeriksa kecocokannya dengan *signature package* [4]. Pendekatan ini lebih kuat dibandingkan sistem yang hanya menyimpan QR Code tanpa validasi kriptografis atau hanya menggunakan *hash* tanpa autentikasi kunci privat. Dibandingkan dengan penelitian berbasis ECDSA dan QR Code [4] yang mengandalkan satu lapisan tanda tangan, skema DSA-RSA-PSS dalam penelitian ini menambahkan lapisan verifikasi kedua, meskipun dengan konsekuensi bertambahnya kompleksitas manajemen kunci. Perlu dicatat bahwa perbandingan langsung terhadap performa skema ECDSA atau EdDSA belum dilakukan dalam penelitian ini, sehingga klaim keunggulan efisiensi relatif terhadap skema lain belum dapat ditegaskan [2], [11].

Kompensasi utama dari pendekatan ini adalah kompleksitas manajemen kunci dan bertambahnya langkah-langkah verifikasi. RSA-PSS menambah waktu *signing*, meskipun pada hasil pengujian *overhead* absolutnya kecil. Risiko lain berada pada penyimpanan *private key* [13]. Jika *private key* disimpan sebagai file lokal tanpa perlindungan tambahan, maka keamanan sistem bergantung pada keamanan perangkat pengguna. Oleh karena itu, pada implementasi produksi, *private key* sebaiknya dikelola menggunakan *key store* terenkripsi, *Certificate Authority*, *Hardware Security Module*, atau *Cloud Key Management Service*. Rekomendasi ini sejalan dengan penekanan FIPS 186-5 bahwa keamanan sistem tanda tangan digital sangat bergantung pada kerahasiaan *private key* [1], [19].

Dari perspektif standar, penggunaan DSA perlu dinyatakan secara hati-hati. Karena FIPS 186-5 tidak lagi menyetujui DSA untuk pembuatan tanda tangan baru, penelitian ini lebih tepat diposisikan sebagai prototipe akademik dan evaluasi terhadap dokumen PDF, bukan sebagai sistem tanda tangan digital yang ditujukan langsung untuk kepatuhan terhadap standar modern. Untuk pengembangan jangka panjang, DSA dapat dibandingkan atau diganti dengan ECDSA/EdDSA, sedangkan RSA-PSS tetap dapat digunakan sebagai pembanding klasik. Studi terbaru menunjukkan bahwa ECDSA, SM2 berbasis *timestamp*, dan skema post-quantum mulai menjadi arah penting dalam tanda tangan digital modern [4], [7], [20].

Secara praktis, hasil implementasi menunjukkan bahwa kendala performa utama bukan pada DSA atau RSA-PSS, melainkan pada proses *render* PDF dan deteksi QR Code. Hal ini berarti optimasi aplikasi dapat dilakukan dengan memisahkan proses kriptografi dan proses visual. Misalnya, sistem dapat menyimpan indeks QR, menerapkan *lazy rendering* pada halaman PDF, atau melakukan verifikasi metadata terlebih dahulu sebelum melakukan deteksi QR. Strategi ini penting karena dalam lingkungan administrasi, pengguna lebih merasakan waktu tunggu antarmuka daripada waktu komputasi tanda tangan yang berlangsung dalam skala milidetik.

4. KESIMPULAN

Penelitian ini berhasil mengimplementasikan skema tanda tangan digital berbasis kombinasi Digital Signature Algorithm (DSA) dan RSA Probabilistic Signature Scheme (RSA-PSS) dengan SHA-256 sebagai fungsi *hash* pada dokumen PDF menggunakan Python. Pengujian fungsional dengan pendekatan *black-box testing* menunjukkan bahwa seluruh fitur utama sistem berjalan sesuai spesifikasi. Pengujian skenario keamanan terhadap empat kondisi, yaitu dokumen asli, *secret code* tidak valid, modifikasi isi PDF, dan modifikasi QR Code, membuktikan bahwa sistem secara konsisten mampu membedakan dokumen autentik dari dokumen yang telah dimanipulasi. Analisis performa komputasi menunjukkan bahwa komponen kriptografi memiliki *overhead* yang sangat rendah, dengan rata-rata waktu *hashing SHA*-256 sebesar 0,0018 detik, *signing DSA* sebesar 0,0014 detik, dan *signing RSA*-PSS sebesar 0,0028 detik. *Bottleneck* utama sistem terletak pada proses deteksi QR Code dan *render* PDF, dengan kontribusi masing-masing sebesar 56,43% (rata-rata 1,1258 detik) dan 43,27% (rata-rata 0,8633 detik) dari total waktu eksekusi. Temuan ini mengindikasikan bahwa penerapan skema kriptografi DSA-RSA-PSS tidak menimbulkan beban komputasi yang signifikan pada konteks prototipe yang diuji. Kontribusi utama penelitian ini terletak pada implementasi dan pengukuran sistem, yaitu integrasi dua lapisan tanda tangan digital dalam satu alur PDF desktop beserta profil performa antarkomponen yang terukur, bukan pada inovasi algoritma kriptografi. Penerapan sistem ini pada lingkungan produksi memerlukan penguatan infrastruktur, termasuk implementasi *Certificate Authority* (CA) dan *Public Key Infrastructure* (PKI) dan manajemen kunci yang aman, sebelum dapat dinyatakan siap untuk deployment skala lebih luas. Penelitian selanjutnya perlu memperkuat evaluasi kriptografis melalui pengukuran avalanche effect SHA-256, pengujian variasi panjang kunci DSA dan RSA, serta perbandingan skema DSA-RSA-PSS dengan ECDSA dan EdDSA dari sisi keamanan dan efisiensi. Pengujian performa juga perlu diperluas pada dokumen PDF berukuran 1–50 MB dengan tingkat kompleksitas yang beragam agar hasil lebih representatif untuk lingkungan produksi.

KONFLIK KEPENTINGAN

Penulis menyatakan bahwa tidak terdapat konflik kepentingan dalam penyusunan artikel ini. Hasil penelitian berupa data analisis dan aplikasi yang telah dibuat merupakan kajian dan karya ilmiah penulis sendiri.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada pihak-pihak yang telah banyak memberikan arahan dalam penelitian ini, khususnya kepada Dosen Pembimbing dan Program Studi Teknik Informatika Universitas Pelita Bangsa atas dukungan selama proses penelitian.

DAFTAR PUSTAKA

[1] National Institute of Standards and Technology (US), “Digital Signature Standard (DSS),” National Institute of Standards and Technology (U.S.), Washington, D.C., NIST FIPS 186-5, Feb. 2023. doi: 10.6028/NIST.FIPS.186-5.

[2] J. K. Dalimunthe and A. Ikhwan, “Implementasi Sistem Informasi Pengelolaan Arsip Elektronik dengan Algoritma *Hashing* untuk Integritas Data,” vol. 11, no. 1, 2026.CESS (Journal of Computer Engineering, System and Science), 11(1), pp. 41-50. doi: 10.24114/cess.v11i1.70979.

[3] R. A. Azdy, “Tanda tangan Digital Menggunakan Algoritme Keccak dan RSA,” *Jurnal Nasional Teknik Elektro dan Teknologi Informasi (JNTETI)*, vol. 5, no. 3, Sep. 2016, doi: 10.22146/jnteti.v5i3.255.

[4] T. Wellem, Y. Nataliani, and A. Iriani, "Academic Document *Authentic*ation using Elliptic Curve Digital Signature Algorithm and QR Code," *JOIV : Int. J. Inform. Visualisation, vol. 6, no. 3, p. 667, Sep. 2022, doi: 10.30630/joiv.* 6.2.872.

[5] L. J. Pangaribuan, C. I. Cahyadi, J. Banjarnahor, B. Barus, and B. N. Siahaan, "Strategi Otentikasi Dokumen Pada Email Menggunakan Digital Signature dengan Algoritma Schnorr".KLIK: Kajian Ilmiah Informatika dan Komputer, 3(4), pp. 384-392.

[6] F. Lalem, A. Laouid, M. Kara, M. Al-Khalidi, and A. Eleyan, "A Novel Digital Signature Scheme for Advanced Asymmetric Encryption Techniques," *Applied Sciences*, vol. 13, no. 8, p. 5172, Apr. 2023, doi: 10.3390/app13085172.

[7] A. Saepulrohman and A. Ismangil, "Data integrity and security of digital signatures on electronic systems using the digital signature algorithm (DSA)," *IJECS*, vol. 1, no. 1, pp. 11–15, Jun. 2021, doi: 10.24042/ijecs.v1i1.7923.

[8] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Commun. ACM*, vol. 21, no. 2, pp. 120–126, Feb. 1978, doi: 10.1145/359340.359342.

[9] M. Ramadhoni and H. Santoso, "Performance Comparison between Signature Cryptography: A Case Study on SNAP Indonesia," *SinkrOn*, vol. 8, no. 4, pp. 2327–2335, Oct. 2023, doi: 10.33395/sinkron.v8i4.12819.

[10] P. Natho, S. Boonmee, and N. Khongkrairoek, "Evaluation of Digital Signatures Using RSA and HMAC Algorithms," *Eng. Technol. Appl. Sci. Res.*, vol. 15, no. 6, pp. 29509–29514, Dec. 2025, doi: 10.48084/etasr. 14149.

[11] K. Somsuk, "The special algorithm based on RSA cryptography for *signing* and *verifying* digital signature," *Heliyon*, vol. 11, no. 4, p. e42481, Feb. 2025, doi: 10.1016/j.heliyon.2025.e42481.

[12] J. Hutagalung, P. S. Ramadhan, and S. J. Sihombing, "Keamanan Data Menggunakan Secure *Hashing* Algorithm (SHA)-256 dan Rivest Shamir Adleman (RSA) pada Digital Signature," *JTIIK*, vol. 10, no. 6, pp. 1213–1222, Dec. 2023, doi: 10.25126/jtiik.1067319.

[13] Y. Anshori, A. Y. Erwin Dodu, and D. M. P. Wedananta, "Implementasi Algoritma Kriptografi Rivest-Shamir-Adleman (RSA) pada Tanda Tangan Digital," *tc*, vol. 18, no. 2, pp. 110–121, May 2019, doi: 10.33633/tc.v18i2.2166.

[14] F. J. Aufa, Endroyono, and A. Affandi, "Security System Analysis in Combination Method: RSA Encryption and Digital Signature Algorithm," in *2018 4th International Conference on Science and Technology (ICST)*, Yogyakarta: IEEE, Aug. 2018, pp. 1–5. doi: 10.1109/ICSTC.2018.8528584.

[15] M. A. Sadikin and R. W. Wardhani, "Implementation of RSA 2048-bit and AES 256-bit with Digital Signature For Secure Electronic Health Record Application".CommIT (Communication & Information Technology) Journal, 10(2), pp. 63-69. doi: 10.21512/commit.v10i2.1549.

[16] K. Somsuk, "The development of *signing* and *verification* methods for high-speed digital signatures on electronic official documents by using RSA cryptography," *Cogent Engineering*, vol. 11, no. 1, p. 2432513, Dec. 2024, doi: 10.1080/23311916.2024.2432513.

[17] A. Fitri, I. Abdullah, T. Ramada, R. Saputri, J. Saharani, and M. F. Gultom, "Implementasi Algoritma Asimetris DSA (*Digital Signature Algorithm*) dalam Aplikasi Sederhana untuk Tanda Tangan Digital," vol. 1, no. 1, 2025.JIKUM: Jurnal Ilmu Komputer, 1(1), pp. 22-26. doi: 10.62671/jikum.v1i1.39.

[18] D. Arisandi, M. B. Yusuf, and S. Sukri, "Pemeriksaan Integritas Dokumen Dengan Digital Signature Algorithm," *JOISIE*, vol. 4, no. 1, p. 1, Jun. 2020, doi: 10.35145/joisie.v4i1.508.

[19] E. A. Adeniyi, P. B. Falola, M. S. Maashi, M. Aljebreen, and S. Bharany, "Secure Sensitive Data Sharing Using RSA and ElGamal Cryptographic Algorithms with *Hash* Functions," *Information*, vol. 13, no. 10, p. 442, Sep. 2022, doi: 10.3390/info13100442.

[20] Z. Cao, B. Gao, X. Xiong, and Z. Liu, "An Improved SM2 Digital Signature Algorithm with High-Precision *Timestamp*s for Trusted Metrological Data," *Sensors*, vol. 25, no. 16, p. 4920, Aug. 2025, doi: 10.3390/s25164920.