IMPLEMENTASI TANDA TANGAN DIGITAL DENGAN ALGORITMA EdDSA DAN SHA-256 UNTUK VERIFIKASI KEASLIAN DOKUMEN

TUGAS AKHIR

Diajukan Oleh :

SYARIFAH MAITSA ZAKIAH

NIM. 190705009

Mahasiswa Fakultas Sains dan Teknologi
Program Studi Teknologi Informasi

FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI AR-RANIRY
BANDA ACEH
2026 M/1447 H

LEMBAR PERSETUJUAN

IMPLEMENTASI TANDA TANGAN DIGITAL DENGAN ALGORITMA EdDSA DAN SHA-256 UNTUK VERIFIKASI KEASLIAN DOKUMEN

TUGAS AKHIR

Diajukan kepada Fakultas Sains dan Teknologi Universitas Islam Negeri (UIN) Ar-Raniry Banda Aceh Sebagai Salah Satu Beban Studi Memperoleh Gelar Sarjana (S1) dalam Ilmu/Prodi Teknologi Informasi

Oleh :

SYARIFAH MAITSA ZAKIAH

190705009

Mahasiswa Fakultas Sains dan Teknologi

Program Studi Teknologi Informasi

Disetujui Untuk Dimunaqasyahkan Oleh:

Pembimbing I,	Pembimbing II,
Malahayati, M.T.	**Dr. Hendri Ahmadian, S.Si.,M.I.M**
NIP. 198301272015032003	**NIP. 198301042014031002**

Mengetahui,
Ketua Program Studi Teknologi Informasi

Malahayati, M.T.
NIP. 198301272015032003

LEMBAR PENGESAHAN

IMPLEMENTASI TANDA TANGAN DIGITAL DENGAN ALGORITMA EdDSA DAN SHA-256 UNTUK VERIFIKASI KEASLIAN DOKUMEN

TUGAS AKHIR

Telah Diuji Oleh Panitia Ujian Munaqasyah Tugas Akhir
Fakultas Sains dan Teknologi UIN Ar-Raniry Banda Aceh dan Dinyatakan Lulus
Serta Diterima Sebagai Salah Satu Beban Studi Program Sarjana (S1)
Dalam Program Studi Teknologi Informasi

Pada Hari/Tanggal: Selasa, 10 Maret 2026
20 Ramadhan 1447 H

Di Darussalam Banda Aceh

Panitia Munaqasyah Tugas Akhir

Ketua,	Sekretaris,
Malahavati, M.T	**Dr. Hendri Ahmadian, S.Si.,M.I.M**
NIP. 198301272015032003	**NIP. 198301042014031002**
Penguji I,	Penguji II,
Khairan AR, M.Kom	**Aulia Syarif Aziz, S.Kom., M.Sc**
NIP. 198607042014031001	**NIP. 199305212022031001**

Mengetahui
Dekan Fakultas Sains dan teknologi
UIN Ar-Raniry Banda Aceh

Prof. Dr. Ir. Muhammad Dirhamsyah, M.T., IPU
NIP. 196210021988111001

LEMBAR PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama	: Syarifah Maitsa Zakiah
NIM	: 190705009
Program Studi	: Teknologi Informasi
Judul	: Implementasi Tanda Tangan Digital Dengan Algoritma EdDSA dan SHA-256 untuk Verifikasi Keaslian Dokumen

Dengan ini menyatakan bahwa dalam penulisan tugas akhir ini, saya:

1. Tidak menggunakan ide orang lain tanpa mampu mengembangkan dan mempertanggungjawabkan;
2. Tidak melakukan plagiasi terhadap naskah skripsi orang lain;
3. Tidak menggunakan skripsi orang lain tanpa menyebutkan sumber asli atau tanpa izin pemilik skripsi;
4. Tidak memanipulasi dan memalsukan data;
5. Mengerjakan sendiri skripsi ini dan mampu bertanggung jawab atas skripsi ini.

Bila dikemudian hari ada tuntutan dari pihak lain atas skripsi saya, dan telah melalui pembuktian yang dapat dipertanggungjawabkan dan ternyata memang dibuktikan bahwa saya telah melanggar pernyataan ini, maka saya siap dikenai sanksi berdasarkan aturan yang berlaku di Fakultas Sains dan Teknologi UIN Ar-Raniry Banda Aceh.

Demikian pernyataan ini saya buat dengan sesungguhnya dan tanpa paksaan dari pihak manapun.

Banda Aceh, 10 Maret 2026

Yang Menyatakan

Syarifah Maitsa Zakiah

ABSTRAK

Nama : Syarifah Maitsa Zakiah

NIM : 190705009

Program Studi : Teknologi Informasi

Judul : Implementasi Tanda Tangan Digital Dengan Algoritma EdDSA dan SHA-256 untuk Verifikasi Keaslian Dokumen

Tanggal Sidang : 10 Maret 2026

Tebal Skripsi : 58 halaman

Pembimbing I : Malahayati, M.T

Pembimbing II : Dr. Hendri Ahmadian, S.Si.,M.I.M

Perkembangan teknologi informasi telah meningkatkan penggunaan dokumen digital dalam berbagai aktivitas administrasi, sehingga diperlukan mekanisme keamanan yang mampu menjamin keaslian dan integritas dokumen. Penelitian ini bertujuan untuk merancang dan mengimplementasikan sistem tanda tangan digital pada dokumen sertifikat digital menggunakan algoritma Edwards-curve Digital Signature Algorithm (EdDSA) dan fungsi hash Secure Hash Algorithm 256 (SHA-256) yang dikonversi ke dalam bentuk QR Code sebagai media verifikasi. Metode penelitian dilakukan melalui beberapa tahapan, yaitu ekstraksi isi dokumen PDF, proses hashing menggunakan SHA-256 untuk menghasilkan nilai hash unik, pembuatan digital signature menggunakan EdDSA dengan pasangan kunci privat dan kunci publik, serta proses verifikasi dokumen melalui pemindaian QR Code. Dataset penelitian terdiri dari dokumen dengan tiga kategori ukuran, yaitu 100–500 KB, 500 KB–2 MB, dan 2–5 MB. Pengujian sistem dilakukan melalui tiga skenario utama, yaitu dokumen asli, dokumen yang dimodifikasi, serta ketidaksesuaian signature atau hash. Hasil pengujian menunjukkan bahwa sistem mampu memverifikasi dokumen dengan akurasi 100%, di mana dokumen asli menghasilkan status valid, sedangkan dokumen yang dimodifikasi atau memiliki ketidaksesuaian signature menghasilkan status invalid. Selain itu, analisis kinerja menunjukkan bahwa waktu proses meningkat secara linier seiring bertambahnya ukuran dokumen, namun seluruh proses tetap berada dalam rentang milidetik sehingga sistem dapat dikategorikan efisien dan responsif. Dengan demikian,

penerapan EdDSA dan SHA-256 berbasis QR Code terbukti efektif dalam menjaga autentikasi dan integritas dokumen digital.

Kata kunci: tanda tangan digital, EdDSA, SHA-256, QR Code, verifikasi dokumen.

ABSTRACT

Name	: Syarifah Maitsa Zakiah
NIM	: 190705009
Department	: Information Technology
Title	: Implementation of Digital Signatures Using EdDSA and SHA-256 Algorithms for Document Authenticity Verification
Date	: 10 March 2026
Number of Pages	: 58 pages
Supervisor I	: Malahayati, M.T
Supervisor II	: Dr. Hendri Ahmadian, S.Si.,M.I.M

The advancement of information technology has increased the use of digital documents in various administrative activities, creating a need for security mechanisms that can ensure the authenticity and integrity of documents. This research aims to design and implement a digital signature system for digital certificate documents using the Edwards-curve Digital Signature Algorithm (EdDSA) and the Secure Hash Algorithm 256 (SHA-256), which is converted into a QR Code for verification purposes. The research method consists of several stages, including extracting the content of PDF documents, performing a hashing process using SHA-256 to generate a unique hash value, creating a digital signature using EdDSA with a pair of private and public keys, and verifying the document through QR Code scanning. The dataset used in this study consists of documents in three size categories: 100–500 KB, 500 KB–2 MB, and 2–5 MB. System testing was conducted through three main scenarios: original documents, modified documents, and mismatched signatures or hash values. The results show that the system can verify documents with 100% accuracy, where original documents produce a valid status, while modified documents or those with mismatched signatures result in an invalid status. Furthermore, performance analysis indicates that processing time increases linearly with document size, but all processes remain within milliseconds, indicating that the system is efficient and responsive. Therefore, the implementation of EdDSA and SHA-256 with QR Code-based

verification is proven to be effective in maintaining the authenticity and integrity of digital documents.

Keywords: digital signature, EdDSA, SHA-256, QR Code, document verification.

KATA PENGANTAR

Puji dan syukur penulis panjatkan ke hadirat Allah SWT yang telah melimpahkan rahmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan tugas akhir yang berjudul "Implementasi **Tanda Tangan Digital Dengan Algoritma Eddsa Dan SHA-256 Untuk Verifikasi Keaslian Dokumen**". Shalawat dan salam senantiasa tercurah kepada Baginda Nabi Muhammad SAW, beserta keluarga dan para sahabat beliau yang telah membawa umat manusia dari zaman kebodohan menuju zaman yang penuh dengan ilmu pengetahuan.

Tugas akhir ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer (S.Kom) pada Program Studi Teknologi Informasi Fakultas Sains dan Teknologi Universitas Islam Negeri Ar-Raniry Banda Aceh. Dalam proses penyusunan tugas akhir ini, penulis telah mendapatkan banyak bantuan, bimbingan, arahan, dukungan serta doa dari berbagai pihak. Oleh karena itu, dengan segala kerendahan hati, penulis ingin menyampaikan ucapan terima kasih yang sebesar-besarnya kepada :

1. Kedua orang tua tercinta, Ibunda Nurliza Asni dan Ayahanda Said Jufri yang penulis hormati dan sayangi, yang telah memberikan kasih saying, doa, dukungan moril dan materil, serta pengorbanan yang tak pernah harap Kembali. Terima kasih atas segala jerih payah, air mata, dan doa yang tak pernah putus demi keberhasilan penulis. Semoga Allah SWT senantiasa melimpahkan rahmat, kesehatan, keberkahan, dan kebahagiaan kepada mereka di dunia dan akhirat.
2. Keluarga besar dan saudara-saudara penulis yang selalu memberikan semangat, dukungan, motivasi dalam setiap Langkah perjalanan pendidikan penulis.
3. Bapak Prof. Dr. Ir. Muhammad Dirhamsyah, M.T., IPU, selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Ar-Raniry Banda Aceh, yang telah memberikan dukungan dalam penyelesaian studi penulis.
4. Ibu Malahayati, M.T, selaku Ketua Program Studi Teknologi Informasi sekaligus Pembimbing I, yang telah meluangkan waktu, pikiran, dan tenaga

untuk membimbing penulis dengan penuh kesabaran dan memberikan arahan yang sangat berharga dalam penyusunan tugas akhir ini.

5. Bapak Dr. Hendri Ahmadian, S.Si., M.I.M, selaku Pembimbing II tugas akhir penulis, yang telah meluangkan waktu, pikiran, dan tenaga untuk memberikan masukan, bimbingan, dan arahan dalam penyusunan tugas akhir ini.
6. Bapak Khairan AR, M.Kom, selaku Sekretaris Program Studi Teknologi Informasi yang telah memberikan bimbingan dan arahannya dalam penyusunan tugas akhir ini.
7. Bapak/Ibu dosen Program Studi Teknologi Informasi yang telah memberikan ilmu pengetahuan dan pengalaman yang sangat bermanfaat selama penulis menempuh Pendidikan di Program Studi Teknologi Informasi.
8. Ibu Cut Rahmadiana, S.Si, selaku staf Program Studi Teknologi Informasi yang telah membantu penulis dalam pengurusan administrasi dan keperluan surat-menyurat selama proses penyelesaian tugas akhir ini.
9. Sahabat-sahabat terdekat yang tak bisa penulis sebut satu persatu namanya, yang telah banyak membantu, berbagi ilmu, dan memberikan semangat baik secara fisik maupun mental dalam penyelesaian tugas akhir ini. Semoga persahabatan kita tetap terjaga dan semoga kita semua bisa mencapai mimpinya masing-masing.
10. Semua pihak yang telah membantu baik pelajaran ataupun pengalaman dalam penyelesaian tugas akhir ini yang tidak dapat penulis sebutkan satu per satu. Semoga Allah SWT membalas semua kebaikan yang telah diberikan.
11. Terakhir, kepada Syarifah Maitsa Zakiah. Ya, diri saya sendiri. Apresiasi sebesar-besarnya untuk diri sendiri yang telah berjuang untuk menyelesaikan apa yang telah dimulai. Terima kasih telah bekerja keras dan bertahan sejauh ini. Untuk setiap malam yang dihabiskan dalam kelelahan, setiap pagi yang disambut dengan kekhawatiran, namun tetap dijalani dan berhasil dilalui. Teruslah belajar dan mensyukuri nikmat yang Allah SWT berikan. Tetap semangat untuk terus berusaha, berbahagialah dimanapun

kamu berada. Penulis berdoa agar langkah kecilmu selalu diperkuat, dikelilingi orang-orang baik dan hebat, serta mimpimu satu persatu akan terjawab. Aamiin.

Penulis menyadari sepenuhnya bahwa dalam peyusunan tugas akhir ini tidak sempurna, untuk itu dengan segala kerendahan hati Penulis menerima saran dan kritikan guna menyempurnakan penyusunan tugas akhir ini. Akhir kata, semoga tugas akhir ini dapat bermanfaat bagi penulis dan pembaca dan semoga dicatat sebagai sebuah amal kebaikan oleh Allah SWT.

Banda Aceh, 10 Februari 2026

Penulis

Syarifah Maitsa Zakiah

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB I

PENDAHULUAN

1.1. Latar Belakang

Di era digital yang semakin maju, penggunaan dokumen elektronik telah menjadi hal yang umum dalam berbagai sektor, termasuk pendidikan, pemerintahan, dan bisnis. Namun, dengan meningkatnya penggunaan dokumen digital, tantangan terkait keamanan dan keaslian dokumen juga semakin kompleks. Dokumen elektronik, seperti sertifikat, sering kali menjadi target pemalsuan dan manipulasi, yang dapat merugikan individu dan institusi (Ubaidillah & Murti, 2021). Kejadian ini menimbulkan kebutuhan mendesak akan sistem yang dapat menjamin integritas dan keaslian dokumen yang dipertukarkan secara elektronik. Menyikapi permasalahan yang terjadi, solusi yang bisa dilakukan ialah implementasi tanda tangan digital atau *digital signature* yang dapat menjaga keamanan dan integritas data atau dokumen.

Di Indenesia, implementasi tanda tangan digital telah mendapatkan pengakuan hukum sejak disahkannya Undang-Undang Informasi dan Transaksi Elekronik (UU ITE) pada tahun 2008 (Undang-Undang Nomor 11, 2008). Teknologi ini berfungsi untuk mengautentikasi dokumen digital dengan cara yang aman dan terverifikasi, memanfaatkan prinsip-prinsip kriptografi. Tanda tangan digital memungkinkan pengirim untuk menandatangani dokumen secara elektronik, sehingga penerima dapat memverifikasi keaslian dan integritas dokumen tersebut. Meskipun tanda tangan digital telah ada, sistem konvensional masih memiliki kelemahan, seperti ketergantungan pada otoritas pusat dan potensi serangan pihak ketiga (Makhdoom dkk, 2020). Hal ini menimbulkan keraguan terhadap keandalan dan keamanan sistem tanda tangan digital yang ada saat ini.

Terdapat berbagai algoritma kriptografi yang digunakan dalam implementasi tanda tangan digital, diantaranya *Elliptic Curve Digital Signature Algorithm* (ECDSA), *Rivest Shamir Adleman* (RSA), dan *Edwards-curve Digital Signature Algorithm* (EdDSA). ECDSA adalah algoritma yang menggunakan konsep matematika kurva

eliptik untuk membentuk tanda tangan digital yang memberikaan keamanan yang tinggi dengan panjang kunci yang relative pendek. Adapun RSA merupakan algoritma kriptografi yang paling umum digunakan, menawarkan keamanan yang teruji namun memiliki kelemahan dalam hal kecepatan dan efisiensi, terutama pada kunci yang besar sehingga membutuhkan memori dan bandwidth yang besar. Sedangkan EdDSA, sebagai variasi lain dari ECDSA, dirancang untuk mengatasi beberapa kelemahan performa dan kemanan yang ada pada ECDSA, menawarkan kecepatan dan keamanan yang lebih baik serta lebih sederhana dan efisien (Suantara et al., 2024). Permasalahan yang dihadapi adalah kelemahan dari metode konvensional dalam menjamin keamanan dokumen digital. Penggunaan algoritma kriptografi seperti *Rivest Shamir Adleman* (RSA) dan *Elliptic Curve Digital Signature Algorithm* (ECDSA) terbukti memiliki kelemahan dari sisi performa dan keamanan terhadap serangan side-channel (Cinal & Sobolewski, 2025)

Beberapa penelitian sebelumnya yang mengintegrasikan algoritma *Rivest Shamir Adleman* (RSA), diantaranya pada penelitian Hutagalung dkk (2023) yang berjudul "*Keamanan Data Menggunakan Secure Hashing Algorithm (SHA)-256 Dan Rivest Shamir Adleman (RSA) Pada Digital Signature*" berfokus pada penerapan kriptografi dengan menggabungkan algoritma SHA-256 dan *Rivest Shamir Adleman* (RSA) untuk menjaga keaslian serta keamanan Surat Keterangan Lulus (SKL) di PKBM Hanuba Medan. Sistem yang dirancang berbasis web dengan fitur tanda tangan digital yang dikonversi menjadi kode QR sehingga memudahkan validasi dan verifikasi dokumen. Melalui SHA-256 untuk menghasilkan *massage digest* dan RSA untuk enkripsi-dekripsi tanda tangan digital. Sementara dalam penelitian Winanda dkk (2025) dengan judul "*Meningkatkan Keamanan Invoice Dengan Enkripsi QR-Code dan Digital Signature Berbasis RSA dan SHA-256*" yang mengungkapkan bahwa terbukti efektif dalam meningkatkan keamanan dan keaslian dokumen digital dengan sistem yang dikembangkan mampu memverifikasi integritas dokumen secara akurat tanpa menyebabkan perubahan ukuran *file* yang signifikan, dengan rata-rata peningkatan hanya 0,13%. Adapun pada penelitian Nurjaman (2024) yang berjudul

"*Penanda Tanganan Dokumen Digital Pada Sistem Penyimpanan File Menggunakan Kombinasi Algoritma SHA3-512 dan RSA Untuk Mempertahankan Keaslian Data Dokumen*" dengan hasil pengujian yang membuktikan bahwa proses validasi dan penandatanganan efektif, dengan tingkat keamanan bergantung pada panjang kunci privat serta ukuran *file* dokumen. Pada penelitian tersebut metode yang menggunakan algoritma SHA-256 dan RSA cukup aman hingga saat ini. Algoritma SHA-256 sendiri masih aman, namun kelemahan utama terletak pada RSA yang memiliki kecepatan rendah, membutuhkan kunci yang besar, kerentanan terhadap serangan kuantum, dan tidak efisien. Oleh karena itu, penelitian ini mengisi kekosongan dengan menghadirkan solusi yang lebih aman, efisien, dan praktis melalui algortitma EdDSA dan SHA-256 pada sistem tanda tangan digital berbasis QR *code*.

Berdasarkan permasalahan di atas, penelitian ini akan mengeksplorasi dan menganalisis tanda tangan digital menggunakan algoritma *Edwards-curve Digital Signature Algorithm* (EdDSA) dan SHA-256 berbasis QR *code* pada sertifikat digital. Penelitian ini diharapkan dapat memberikan kontribusi dalam pengembangan sistem yang lebih aman dan efisien dalam pengelolaan dokumen elektronik, serta memberikan wawasan bagi pengambil kebijakan dan praktisi di bidang teknologi informasi dan diharapkan dapat meningkatkan integritas dan kepercayaan dalam transaksi dokumen elektronik.

1.2. Rumusan Masalah

Dari latar belakang di atas maka dapat diambil beberapa rumusan masalah sebagai berikut:

1. Bagaimana merancang sistem tanda tangan digital untuk dokumen sertifikat digital menggunakan algoritma *Edwards-curve Digital Signature Algorithm* (EdDSA) dan SHA-256?
2. Bagaimana implementasi tanda tangan digital tersebut dalam kode QR serta evaluasi kinerja algoritma *Edwards-curve Digital Signature Algorithm* (EdDSA) dalam proses otentikasi dokumen digital berdasarkan waktu

pembuatan tanda tangan, waktu verifikasi, ukuran tanda tangan, dan status validitas verifikasinya?

1.3. Tujuan Penelitian

Berdasarkan rumusan masalah yang telah disusun, penelitian ini bertujuan untuk:

1. Merancang sistem tanda tangan digital untuk dokumen sertifikat digital menggunakan algoritma *Edwards-curve Digital Signature Algorithm* (EdDSA) dan SHA-256.
2. Menganalisis kinerja implementasi algoritma *Edwards-curve Digital Signature Algorithm* (EdDSA) yang dikonversi dalam format kode QR sebagai instrumen otentikasi dokumen sertifikat digital.

1.4. Batasan Penelitian

Dalam penelitian ini mempunyai beberapa batasan masalah, berikut batasan-batasan masalahnya:

1. Penelitian ini hanya menggunakan algoritma kriptografi SHA-256 sebagai fungsi *hash* dan *Edwards-curve Digital Signature Algorithm* (EdDSA).
2. Sistem yang dibangun hanya mencakup fitur perancangan tanda tangan digital, pencetakan sertifikat digital dengan kode QR, dan proses verifikasi dokumen.
3. Dokumen yang digunakan adalah sertifikat digital sebagai objek otentikasi dibatasi pada format PDF untuk menjaga konsistensi nilai *hash*.

1.5. Manfaat Penelitian

Adapun manfaat dalam penelitian ini :

1. Memberikan pemahaman tentang proses sistem tanda tangan digital untuk dokumen sertifikat digital menggunakan algoritma *Edwards-curve Digital Signature Algorithm* (EdDSA) dan SHA-256.

2. Memberikan kontribusi bagi pengembangan penelitian di bidang kriptografi, khususnya mengenai efisiensi algoritma *Edwards-curve Digital Signature Algorithm* (EdDSA) dalam skema penandatanganan dokumen digital

BAB II
TINJAUAN PUSTAKA

2.1. Penelitian Terdahulu

Guna mendukung penelitian ini, penulis melakukan kajian terhadap penelitian-penelitian terdahulu yang berfungsi untuk memberikan gambaran mengenai perkembangan penelitian dalam bidang yang relevan, sekaligus menjadi landasan teoritis bagi penelitian ini. Dengan demikian, penulis dapat mengidenifikasi relevansi, kontribusi, serta celah penelitian yang masih dapat dikembangkan.

Adapun penelitian dengan judul "*Keamanan Data Menggunkan Secure Hashing Algorithm (SHA)-256 Dan Rivest Shamir Adleman (RSA) Pada Digital Signature"* yang mengeksplorasi penggunaan tanda tangan digital menggunakan kombinasi RSA dan SHA-256 memanfaatkan QR *Code* untuk menjawab permasalahan pemalsuan dokumen, khususnya pada Surat Keterangan Lulus (SKL) di PKBM Hanuba Medan. Hasil dari penelitian ini menunjukan tanda tangan digital dapat diverifikasi dan hasil dekripsi indentik dengan hash dokumen elektronik, sehingga keaslian dokumen dapat dipastikan. Meskipun demikian,ukuran kunci yang digunakan RSA lebih besar dan relatif boros energi untuk proses enkripsi/dekripsi karena memerlukan perhitungan bilangan prima besar, serta belum dibandingkan dengan algoritma kriptografi modern lain seperti EdDSA atau SHA-3 yang berpotensi lebih optimal (Hutagalung et al., 2023)

Penelitian lainnya yang berjudul "*Implementasi Skema QR-Code dan Digital Signature Menggunakan Kombinasi Algoritma RSA dan AES Untuk Pengamanan Data Sertifikat Elektornik"*. Studi ini mengeksplorasi pengamanan sertifikat elektronik menggunakan *Rivest Shamir Adleman* (RSA) dan *Advenced Encryption Standard* (AES) sebagai algoritma kriptografi asimetris dan simetris serta implementasi fungsi hash SHA-3 untuk menjaga integritas dan keaslian data. Pada penelitian ini sistem yang dibangun memiliki performa enkripsi dan dekripsi yang cepat, serta nilai *avalanche effect* 40,61% yang membuktikan perubahan kode *chipertext* sangat acak sehingga

sertifikat dapat diverifikasi secara mudah melalui QR-*Code*. Namun, implementasi super enkripsi antara RSA dan AES menambah kompleksitas sistem yang berpontensi mengurangi efisiensi pada skala data yang lebih besar (Nuraeni et al., 2020)

Studi selanjutnya berjudul "*Pengamanan Data Teks Menggunakan Metode Digital Signature Algorithm Dan Advanced Encryption Standard (AES)*" yang berfokus pada penerapan DSA sebagai tanda tangan digital dan AES untuk melakukan enkripsi serta dekripsi file teks. Hasil penelitian menunjukkan file teks terenkripsi mengalami perubahan ukuran hingga 30% dan tidak dapat diakses sebelum proses dekripsi. Hasil dekripsi menunjukkan kesesuaian dengan file asli, sehingga integritas dan keaslian data valid. Adapun kelemahan dari penelitian ini, *Digital Signature Algorithm* (DSA) hanya mendukung proses autentikasi dan tidak dapat digunakan untuk enkripsi, sehingga sepenuhnya bergantung pada *Advanced Encryption Standard* (AES). Selain itu, penelitian ini belm mengkaji kinerja pada konteks *multi-user* maupun distribusi data jaringan (Alfani et al., 2024)

Dalam penelitian selanjutnya yang berjudul "*Implementasi Sistem Otentikasi Dokumen Berbasis Quick Response (QR) Code dan Digital Signature"*. Penelitian ini berfokus pada penggunaan *Secure Hash Algorithm-256* (SHA-256) sebagai fungsi *hash* dan *algoritma Rivest-Shamir-Adleman* (RSA) sebagai proses enkripsi/dekrpsi dan pembangkitan kunci public/kunci privat dengan memanfaatkan QR *code* sebagai media verifikasi dan tanda tangan digital sebagai jaminan keaslian dokumen. Pengujian sistem dilakukan terhadap 30 sertifikat, terdiri dari 15 sertifikat mencantumkan QR *code* yang dihasilkan oleh sistem sebagai *signature*-nya dan 15 sertifikat lainnya memuat *signature* QR *code* yang dibuat menggunakan QR *code generator* lain. Hasil dari penelitian ini menunjukkan bahwa implementasi sistem otentikasi dokumen berbasis QR *code* dan *digital signature* ini dapat memastikan keaslian dan integritas dokumen sehingga mencegah pemalsuan dokumen. Namun, sistem ini hanya terbatas pada implementasi berbasis QR *code* tanpa mengevaluasi efisiensi performa enkripsi maupun keamanan terhadap serangan tingkat lanjut (Lorien & Wellem, 2021)

Pada penelitian lain yang berjudul "*Implementasi Tanda Tangan Digital Menggunakan ECDSA (Studi Kasus: Jurnal Tipe File PDF)*" yang berfokus pada penerapan algoritma *Elliptic Curve Digital Signature Algorithm* (ECDSA) untuk mencegah duplikasi atau modifikasi dokumen ilmiah, khususnya jurnal dalam format PDF. Hasil penelitian yang di dapatkan bahwa sistem mampu mendeteksi perubahan sekecil apapun pada dokumen dan menjamin *authenticity*, *integrity*, dan *non-repudiation* suatu data, khususnya pada file PDF. Dalam penelitian ini pengujian hanya dilakukan pada sepuluh file jurnal, sehingga belum mencerminkan kinerja sistem dalam skala besar. Selain itu, penelitian ini hanya menggunakan SHA-1 sebagai fungsi *hash*, yang tidak aman untuk tanda tangan digital sehingga resiko manipulasi *file* karena nilai *hash* bisa dipalsukan (Arwah et al., 2024)

Penelitian selanjutnya dengan judul "*Implementasi Secure Hash Algorithm-256 dan Advanced Encryption Standard untuk Verifikasi Tanda tangan Digital*" yang mengkaji penggunaan kombinasi fungsi *hash* SHA-256 dan enkrpsi AES-256 untuk mengatasi masalah pemalsuan tanda tangan digital yang sering terjadi pada dokumen resmi. Sistem yang dibangun berbasis web yang dikembangkan mampu menghasilkan tanda tangan digital yang dapat diverifikasi dengan baik ditunjukkan dari hasil uji *usability* sebesar 93,71% sebagai tingkat kepuasan pengguna, menandakan bahwa sistem relevan dan efisien. Namun, pada penelitian ini sistem belum diuji secara menyeluruh terhadap serangan keamanan modern dan juga belum melakukan perbandingan dengan algoritma anda tangan digital berbasis kunci publik lainnya (Nadimsyah et al., 2024)

Dalam studi lain yang berjudul "*Academic Document Authentication using Elliptic Curve Digital Signature Algorithm and QR Code*" yang mengeksplorasi penerapan *Elliptic Curve Digital Signature Algorithm* (ECDSA) yang disimpan dalam QR *code* menjaid solusi untuk menjaga keaslian dan integritas dokumen akademik. Pemanfaatan ECDSA menggunakan kurva prime256v1, hash SHA-256, dan QR *Code* berhasil mengeimplementasikan sistem otentikasi dokumen akademik yang mampu menerima QR *Code* asli dan menvalidasi keaslian tanda tangan digital. Namun, dengan

tidak adanya sertifikasi *encoder/decoder* QR *Code,* validasi masih dapat berhasil meskipun QR *Code* dihasilkan olehe generator lain, selama pesan dan tanda tangan tidak di ubah. Hal ini menjadi celah keamanan yang signifikan serta belum ada pembahasan terkait ketahanan terhadap serangan kriptografi modern (Wellem et al., 2022)

Pada penelitian yang berjudul "*Secure Files Using Secure Hash Algorithm and Elliptical Curve Cryptography*" yang menjawab permasalahan keamanan file pada era digital yang sangat bergantung pada penyimpanan *cloud* dan sistem berbagi data menggunakan kombinasi *Secure Hash Algorithm* (SHA-256) untuk verifikasi integritas data dan penerapan *Elliptic Curve Cryptography* (ECC) untuk enkripsi asimetris sebagai solusi yang tepat untuk menjaga kerahasiaan , integritas, serta verifikasi file dengan beban komputasi yang relatif ringan. Hasil penelitian ditunjukkan bahwa kombinasi SHA-256 dan ECC mampu memberikan keamanan tingkat tinggi dan dapat melakukan enkripsi dan dekripsi file dengan waktu yang relative singkat dan ruang penyimpanan lebih kecil dibandingkan dengan RSA. Meski begitu, tidak ada analisis mendalam mengenai manajemen kunci pada sistem berbasis ECC, yang merupakan salah satu isu kritis dalam implementasi praktis. Dalam penelitian ini meskipun sistem efisien, penggunaan ECC masih memiliki potensi kerentanan jika kurva yang digunakan tidak aman (Prashanth et al., 2025)

Penelitian selanjutnya dengan judul "*Meningkatkan Keamanan Invoice Dengan Enkripsi QR Code Dan Digital Signature Berbasis RSA Dan SHA-256"* yang berfokus pada penerapan *digital signature* berbasis RSA dan SHA-256 serta diintegrasi dalam QR *code* menjadi solusi yang dapat menjamin keaslian, integritas, dan keamanan dokumen. Hasil pengujian yang dilakukan pada tujuh dokumen menunjukkan bahwa dokumen yang telah diberi *digital signature* dapat diverifikasi keasliannya, memastikan bahwa dokumen yang diterima adalah file asli tanpa modifikasi. Perubahan ukuran file setelah penambahan *digital signature* juga sangat kecil, sebesar 0,13%. Namun dalam penelitian ini, tidak ada analisis terkait efisiensi performa enkripsi-dekripsi ketika jumlah dokumen meningkat secara signifikan. Mengandalkan

RSA dengan panjang kunci 2048-bit, walaupun masih aman, lebih lambat dibanding algoritma modern seperti EdDSA (Winanda et al., 2025)

Dari berbagai penelitian yang telah dilakukan, penulis menyimpulkan bahwa tanda tangan digital dengan dukungan algoritma kriptografi modern dan integrasi QR *Code* terbukti efektif dalam menjaga keaslian serta integritas dokumen digital. Namun, kelemahan umum yang ditemui ialah keterbatasan skala pengujian, penggunaan algoritma klasik yang relatif kurang efisien, serta belum adanya analisis mendalam terhadap serangan kriptografi kontemporer. Adapun celah penelitian yang masih terbuka yakni, pemanfaatan algoritma modern seperti EdDSA untuk meningkatkan efisiensi dan keamanan dokumen digital serta pengujian sistem pada skala besar dengan puluhan dokumen dan *multi-user*. Dengan demikian, penelitian selanjutnya dapat diarahkan untuk menutup celah tersebut melalui pengembangan sistem autentikasi dokumen digital yang lebih efisien, aman, dan cepat.

2.2. Kajian Teori

2.2.1. Kriptografi

Kriptografi modern merupakan disiplin ilmu algoritma matematika yang kompleks untuk mengamankan data melalui proses enkripsi dan dekripsi, dengan tujuan utama menjafa kerahasiaan, integritas, otentikasi, serta non-repudiation informasi (Amalya et al., 2023). Dalam kriptografi, terdapat dua proses, yaitu enkripsi yang mengubah *plaintext* menjadi *ciphertext* yang tidak terbaca dan dekripsi mengembalikan *ciphertext* menjadi *plaintext*. *Plaintext* berarti pesan asli, sedangkan hasil penyandian disebut *ciphertext* (Azhari et al., 2022). Pada era digital, kebutuhan akan keamanan data meningkat secara signifikan seiring dengan pesatnya perkembangan teknologi informasi dan jumlah data yang terus bertambah. Sehingga menuntut penerapan metode kriptografi yang lebih efisien, adaptif, dan tahan terhadap berbagai serangan siber yang semakin canggih, termasuk juga ancaman dari teknologi computer kuantum yang dapat memecah algoritma kriptografi klasik (Amalya et al., 2023).

2.2.2 Algoritma Kriptografi Simetris dan Asimetris

Algoritma simetris merupakan kunci yang sama dalam proses enkripsi dan dekripsi sehingga algoritma ini sering disebut dengan algoritma kunci tunggal. Kunci dalam algoritma ini bersifat rahasia atau *private key* (Meko, 2023). Dua algoritma yang paling banyak digunakan adalah *Advanced Encryption Standard* (AES) dan *Blowfish.* Sedangkan algoritma asimetris adalah algoritma yang menggunakan kunci yang berbeda untuk proses enkripsi dan dekripsi. Keuntungan utama kunci enkripsi asimetris ialah memiliki enkripsi yang kuat dan sulit diprediksi oleh peretas (Saputro et al., 2020) Metode ini meningkatkan keamanan dalam pertukaran kunci dan autentikasi, dengan algoritma yang sangat sering digunakan seperti *Rivest Shamir Adleman* (RSA), *Diffie-Hellman, XTR, Elliptic Curve Cryptography* (ECC), dan *Edward-curve Digital Signature Algorithm* (EdDSA) yang menawarkan tingkat keamanan tinggi dan efisiensi pada aplikasi modern, termasuk komunikasi dan tanda tangan digital (Saputro et al., 2020)

2.2.3 *Digital Signature Algorithm* (DSA)

Tanda tangan digital atau disebut *digital signature* merupakan Teknik kriptografi dasar yang menjadi bagian dari teknologi keamanan digital yang dirancang untuk menjamin integritas dan autentikasi dokumen. Algoritma ini menggunakan *hashing* seperti SHA-1 untuk menghasilkan ringkasan pesan untuk setiap data, sehingga memastikan bahwa data tidak dapat diubah (Saepulrohman & Ismangil, 2021). Fungsi *digital signature* sebagai *marking* pada data yang sebenarnya (tidak ada yang berubah) alhasil keaslian dari dokumen dapat divalidasi. Adapun cara kerja *digital signature* dengan mengubah dokumen digital menjadi *massage digest*, sebuah keluaran fungsi dari *hash* biasanya berbentuk simbol dan angka matematis yang kemudian di-enkripsi menggunakan algoritma *asimetris,* hasil dari enkripsi tersebut yang menjadi tanda tangan dari dokumen tersebut (Fachrul et al., 2022)

2.2.4 *Edwards-curve Digital Signature Algorithm* (EdDSA)

Edwards-curve Digital Signature Algorithm (EdDSA) merupakan implementasi modern dari skeman tanda tangan Schnoor yang menfaatkan struktur grup yang dibentuk oleh titik-titik pada kurva *Edwards* terpilin (*twisted edwards curve*). Berbeda dengan persyaratan skema Schnorr yang mengharuskan grup memiliki ordo prima, kurva Edwards memiliki jumlah titik yang selalu merupakan kelipatan empat. Akibatnya, terdapat subgroup kecil dengan nord 4 para parameter Ed448 dan ordo 8 pada parameter Ed25519, yang merupakan set parameter standar dalam implementasi EdDSA. Kondisi ini memiliki implikasi penting terhadap desain dan keamanan algortma tanda tangan digital tersebut (Cinal & Sobolewski, 2025). Studi terbaru menunjukkan bahwa meskipun standar EdDSA cukup spesifik mengenai validitas tanda tangan, kurangnya kejelasan menyebabkan ketidakkonsistenan pada Pustaka kriptografi utama seperti OpenSSL dan CIRCL (Cinal & Sobolewski, 2025)

2.2.5 *Hashing*

Hashing merupakan teknik yang digunakan untuk menyimpan, menambahkan, menghapus, dan mencari data dengan cara memanfaatkan alamat kunci yang dihasilkan melalui fungsi tertentu. *Hashing* bertujuan untuk memetakan elemen data ke indeks dalam table *hash* , yang berupa array dengan ukuran tetap dan berisi data atau kunci yang berkaitan. Fungsi hashing yang ideal harus cepat dihitung dan mampu meminimalkan terjadinya *collision,* kondisi dimana dua atau lebih kunci dipetakan ke lokasi yang sama. Untuk mengatasi *collision,* dapat menggunakan strategi *Close Hashing* (Opening Address) dan *Open Hashing* (Chaining), dengan berbagai modifikasi fungsi untuk meningkatkan efisiensi dan akurasi (Syahrir & Fatimatuzzahra, 2020). SHA merupakan fungsi hash satu arah yang diterbitkan oleh National Instituteof Standards and Technology sebagai Federal Information Processing Standard tahun 1993 yang sebelumnya dikenal sebagai SHA-0, pada generasi berikutnya, SHA-1 dirilis tahun 1994 yang merupakan perbaikn dari algoritma SHA-0. Tahun 2002, algoritma ini merilis empat versi tambahan lainnya: SHA_224, SHA-256, SHA-384, dan SHA-512 (FIPS 180-3, 2008)

2.2.6 SHA-256

Secure Hash Algorithm 256-bit (SHA-256) merupakan salah satu algoritma *hashing* yang digunakan untuk mengamankan data dengan menghasilkan nilai *hash* unik sepanjang 256-bit. Cara kerja SHA-256 dengan mengubah input data menjadi string karakter tetap (*massage digest)* dengan panjang kunci 256 bit (Winanda et al., 2025). SHA-256 masuk kedalam bagian dari keluarga SHA-2, dimana fungsi hash kriptografi yang banyak digunakan dan dikenal karena ketahanan serta fitur keamanannya. Algoritma ini dikembangkan oleh *National Institute of Standards and Technology* (NIST) dan telah menjadi standar dalam berbagai aplikasi, termasuk tanda tangan digital, hashing kata sandi, dan teknologi blockchain seperti bitcoin (Franck et al., 2024)

2.2.7 Python

Python adalah bahasa pemograman multifungsi dengan level tinggi yang dikembangkan oleh Guido van Rossum pada tahun 1991. Bahasa pemograman ini dikenal karena sintaksnya yang *open source* dan mudah dipahami. Dengan mendukung beberapa paradigma pemograman, termasuk objek, procedural, dan fungsional. Bahasa python sangat fleksibel digunakan dalam berbagai jenis aplikasi (Dyer & Chauhan, 2022). Perkembangan python dalam decade terakhir menjadikannya bahasa utama dalam penggunaan *data science, machine learning, artificial intelligence,* karena kemampuannya mengintegrasikan kemudahan sintaks dengan performa komputasi yang efisien (Raschka et al., 2020)

2.2.8 Sertifikat Digital

Sertifikat digital adala kredensial elektronik yang dikeluarkan oleh *Certificate Authority* (CA) untuk mengikat identitas pemilik dengan sepasan kunci kriptografi publik dan privat. Dalam kerangka *Public Key Infrastructure* (PKI), sertifikat digital memiliki fungsi utama yaitu: autentikasi identitas entitas yang berkomunikasi; integritas data melalui verifikasi tanda tangan digital; *non-repudiasi* sehingga penandatangan tidak dapat menyangkal keterlibatan; dan kerahasiaan dengan

mendukung enkripsi komunikasi. Dengan begitu, PKI menyediakan kerangka formal untuk penerbitan, distribusi, validasi, dan pencabutan sertifikat sesuai standar internasional seperti X.509 (Raschka et al., 2020) Dalam konteks keamanan dokumen, khususnya Pdf, sertifikat digital digunakan untuk menerapkan tanda tangan digital yang menjamin keaslian dokumen dan integritas konten. Sertifikat digital juga memungkinkan enkripsi dokumen sehingga hanya penerima yang sah dapat membukanya. Selain itu, sertifikat mendukung standar keamanan internasional seperti *PDF Advanced Electronic Signatures* (PAdES) yang memastikan kepatuhan hukum dan interoperabilitas lintas platform. Dengan penerapan sertifikat digital, dokumen PDF dapat digunakan sebagai bukti hukum yang sah karena memuat aspek identitas penandatangan, waktu tanda tangan, dan perlindungan dari modifikasi tidak sah (Adobe, 2021)

2.3 Kerangka Berpikir

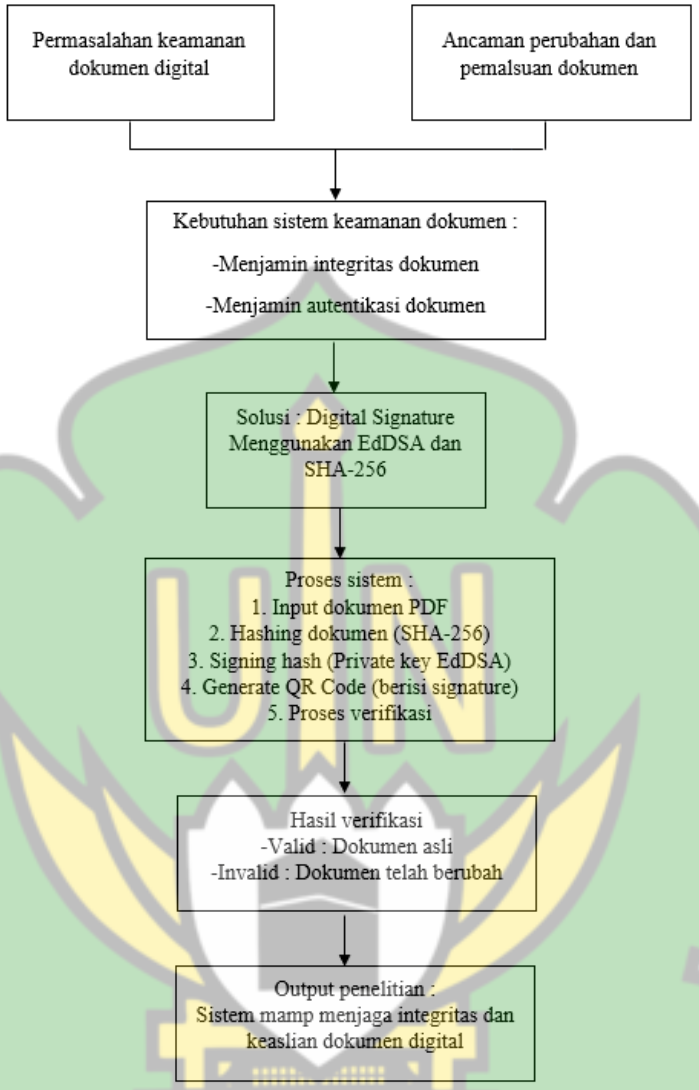
Perkembangan teknologi digital yang semakin pesat membawa konsekuensi terhadap meningkatnya kebutuhan keamanan informasi, khususnya dalam konteks autentikasi dan validasi dokumen elektronik. Fenomena ini diperkuat dengan banyaknya kasus pemalsuan dokumen digital yang menimbulkan permasalahan serius dalam dunia pendidikan, pemerintahan, dan sektor bisnis. Oleh karena itu, diperlukan suatu mekanisme keamanan yang mampu menjamin keaslian (*authenticity*), integritas (*integrity*), dan mencegah penyangkalan (*non-repudiation*) pada dokumen digital.

Untuk mengatasi permasalahan tersebut, penelitian ini mengusulkan penerapan tanda tangan digital (*digital signature*) dengan memanfaatkan algoritma *Edwards-curve Digital Signature Algorithm* (EdDSA) yang dikombinasikan dengan fungsi hash *Secure Hash Algorithm 256* (SHA-256). Algoritma SHA-256 digunakan untuk menghasilkan nilai hash unik dari dokumen, sedangkan EdDSA digunakan untuk melakukan proses penandatanganan digital terhadap nilai hash tersebut menggunakan private key. Dalam proses sistem, dokumen PDF terlebih dahulu diinput ke dalam sistem, kemudian dilakukan proses hashing menggunakan SHA-256 untuk

menghasilkan nilai hash dokumen. Nilai hash tersebut selanjutnya ditandatangani menggunakan *private key* pada algoritma EdDSA untuk menghasilkan *digital signature*. *Signature* yang dihasilkan kemudian dikonversi ke dalam bentuk *QR Code* sehingga memudahkan proses penyimpanan dan verifikasi dokumen.

Tahap selanjutnya adalah proses verifikasi dokumen dengan memindai *QR Code* dan melakukan pengecekan menggunakan *public key*. Hasil verifikasi akan menunjukkan status dokumen, yaitu valid apabila dokumen masih asli dan tidak mengalami perubahan, atau invalid apabila dokumen telah dimodifikasi. Dengan demikian, sistem yang dirancang dalam penelitian ini diharapkan mampu menjaga integritas serta memastikan keaslian dokumen digital secara efektif.

Hasil penelitian kemudian dibandingkan dengan penelitian terdahulu untuk menunjukkan kontribusi serta keunggulan metode yang diusulkan. Berdasarkan uraian diatas, maka kerangka berpikir dalam penelitian ini dapat digambarkan pada gambar 2.3.



Gambar 2.3. Bagan Kerangka Berpikir

BAB III
METODE PENELITIAN

3.1 Jenis Penelitian

Penelitian ini menggunakan metode kuantitatif dengan pendekatan eksperimen. Penelitian kuantitatif digunakan karena fokus pada pengukuran kinerja algoritma EdDSA dan SHA-256 pada dokumen PDF berdasarkan data numerik, seperti waktu proses tanda tangan (signing), waktu verifikasi, ukuran tanda tangan, dan tingkat keberhasilan verifikasi. Metode eksperimen dipilih untuk menguji efektivitas algoritma secara langsung pada dokumen digital (Creswell & Creswell, 2022).

3.2 Waktu dan Lokasi Penelitian

Penelitian dilaksanakan selama dua bulan, pada bulan Januari – Februari 2025 dimulai dari persiapan hingga analisis data. Penelitian dilaksanakan di laboratorium komputer dengan perangkat keras berupa laptop Lenovo Thinkpad T490s dengan layer 14 inci, system operasi Windows 11 Pro, prosessor Intel® Core™ i5-8265U CPU @ 1.60GHz, 1800 MHz, 4 Core(s), 8 Logical Processor(s) dan RAM 16 GB, serta perangkat lunak berupa Python 3.10 dengan pustaka Libsodium/OpenSSL.

2.3 Objek Penelitian

Objek penelitian yang akan dilakukan adalah implementasi *Edwards-curve Digital Signature Algorithm* (EdDSA) yang dimodifikasi menggunakan pra-hash SHA-256 untuk menghasilkan tanda tangan digital pada dokumen PDF. Studi kasus difokuskan pada dokumen sertifikat akademik atau sertifikat pelatihan yang membutuhkan verifikasi keaslian.

3.4 Populasi dan Sampel

- **Populasi**: seluruh dokumen digital dalam format PDF yang dapat ditandatangani secara elektronik.

- **Sampel**: 60 dokumen PDF yang dipilih dengan metode purposive sampling dengan kriteria dokumen berstruktur kompleks berdasarkan ukuran dokumen. Kategori tersebut terdiri dari 20 dokumen berukuran kecil (100–500 KB), 20 dokumen berukuran sedang (500 KB–2 MB), dan 20 dokumen berukuran besar (2–5 MB). Pendekatan ini dipilih agar eksperimen dapat mengukur variasi performa berdasarkan ukuran dokumen (Etikan & Bala, 2017).

3.5 Variabel Penelitian

Variabel penelitian merupakan suatu atribut atau sifat atau nilai dari orang, obyek, atau kegiatan yang mempunyai variasi tertentu yang ditetapkan oleh peneliti untuk dipelajari dan ditarik kesimpulan (Sugiyono, 2020). Dalam penelitian ini, variable independen yang digunakan adalah metode algoritma tanda tangan digital, yaitu implementasi EdDSA dan algoritma hash SHA-256. Variabel dependen meliputi beberapa aspek, antara lain: waktu proses, yang diukur berdasarkan lama eksekusi proses *signing* dan *verification* dalam satu milidetik; keaslian dan integritas dokumen, yang diukur melalui status valid atau invalid dari tanda tanda tangan digital; ukuran, yang ditentukan dari proses setelah dimasukkan kedalam tanda tangan digital; serta kompatibilitas PDF, yang ditentukan dari kemampuan dokumen untuk tetap dibuka setelah ditandatangani. Hal ini digambarkan dalam bentuk tabel 3.1 hubungan antar variable independent-dependen berikut ini.

Tabel 3.1. Hubungan antar Variabel Independen-dependen

Jenis Variabel	Nama Variabel	Indikator
Independen	Metode tanda tangan digital	EdDSA + SHA-256
Dependen	Waktu proses	Lama waktu *signing* & verifikasi (ms)
Dependen	Keaslian & integritas dokumen	Status valid/invalid tanda tanga digital
Dependen	Ukuran dokumen	Dokumen PDF setelah tanda tangan digital (byte)
Dependen	Kompatibilitas PDF	PDF dapat dibuka/tidak setelah ditandatangai

3.6 Teknik Pengumpulan Data

Pengumpulan data dilakukan dengan dua cara:

1. Eksperimen langsung: mengimplementasikan EdDSA-SHA256 untuk menandatangani dan memverifikasi dokumen PDF.
2. Pengukuran kinerja: hasil eksperimen dicatat dalam bentuk:
   - Waktu pembuatan tanda tangan digital (ms).
   - Waktu verifikasi tanda tangan digital (ms).
   - Ukuran tanda tangan digital (byte).
   - Status verifikasi (valid/invalid).

3.7 Rencana Penelitian

Rencana pengujian dilakukan berdasarkan ukuran dokumen dan jumlah iterasi. Setiap skenario dijalankan minimal 5 kali untuk menjaga reliabilitas data (Brendel et al., 2021). Rencana pengujian pada penelitian ini disusun untuk mengevaluasi kinerja algoritma *Edwards-curve Digital Signature Algorithm* (EdDSA) dan SHA-256 dalam proses penandatanganan dan verifikasi dokumen digital berformat PDF.

Pengujian dilakukan terhadap tiga kategori ukuran dokumen, yaitu dokumen kecil (100 − 500 KB)**,** dokumen sedang (500 KB – 2 MB)**,** dan dokumen besar (2 − 5 MB)**.** Klasifikasi ini bertujuan untuk mengetahui pengaruh ukuran *file* terhadap performa algoritma dalam menghasilkan tanda tangan digital serta melakukan proses verifikasi keaslian dokumen. Tabel rencana pengujian terhadap algoritma EdDSA dengan SHA-256 dapat dilihat pada table 3.2 berikut ini.

Tabel 3.2. Rencana Pengujian Algoritma EdDSA-SHA256

No	Ukuran Dokumen PDF	Jumlah Iterasi	Parameter yang Diukur	Keterangan
1	100 − 500 KB (kecil)	5 kali	Signing time, Verification time, Signature size, and Validity	Dokumen sertifikat teks singkat
2	500 KB – 2 MB (sedang)	5 kali	Signing time, Verification time, Signature size, and Validity	Dokumen dengan teks + gambar
3	2 − 5 MB (besar)	5 kali	Signing time, Verification time, Signature size, and Validity	Dokumen dengan elemen grafik kompleks

Berdasarkan pada tabel 3.2 setiap kategori ukuran dokumen diuji sebanyak lima kali (5 iterasi). Pengulangan dilakukan untuk memperoleh nilai rata-rata yang lebih akurat serta mengurangi kemungkinan bias akibat kondisi sistem atau variasi waktu proses selama eksekusi program. Dari hasil pengujian berulang tersebut, diperoleh data empiris yang dapat digunakan untuk analisis performa secara kuantitatif.

Parameter utama yang diukur dalam pengujian meliputi empat aspek penting, yaitu:

1. *Signing time*, yaitu waktu yang dibutuhkan untuk melakukan proses penandatanganan digital terhadap dokumen menggunakan algoritma EdDSA–SHA256.

2. *Verification time*, yaitu waktu yang diperlukan untuk memverifikasi keaslian tanda tangan digital dan memastikan integritas dokumen.
3. *Signature size*, yaitu ukuran tanda tangan digital yang dihasilkan dalam satuan byte, sebagai indikator efisiensi penyimpanan data hasil tanda tangan.
4. *Validity*, yaitu status validitas tanda tangan yang menunjukkan apakah hasil verifikasi dinyatakan valid (sah) atau invalid (tidak sah).

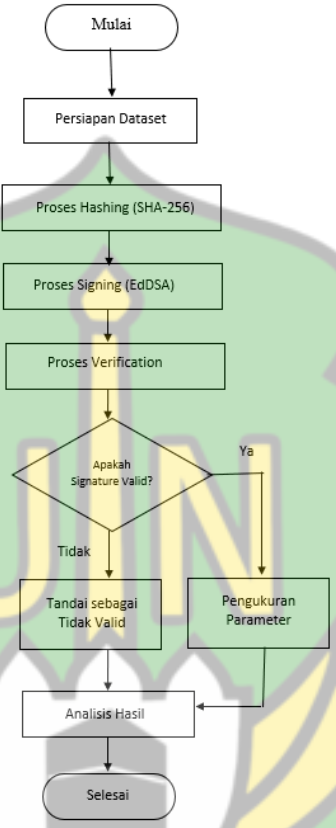
Untuk dokumen berukuran kecil (100 − 500 KB), pengujian dilakukan pada dokumen sertifikat yang hanya berisi teks singkat. Untuk dokumen berukuran sedang (500 KB – 2 MB), pengujian mencakup dokumen dengan kombinasi antara teks dan gambar, seperti sertifikat yang memuat logo, tanda tangan digital bergambar, atau cap instansi. Sementara untuk dokumen berukuran besar (2 – 5 MB), pengujian difokuskan pada dokumen dengan elemen grafis kompleks, seperti diagram, tabel besar, atau ilustrasi beresolusi tinggi.

Rencana pengujian ini diharapkan dapat memberikan gambaran menyeluruh mengenai hubungan antara ukuran dokumen dengan performa algoritma EdDSA–SHA-256, baik dari segi kecepatan, efisiensi ukuran tanda tangan, maupun keakuratan hasil verifikasi. Hasil pengujian akan menjadi dasar dalam penarikan kesimpulan terhadap efektivitas implementasi algoritma ini pada sistem verifikasi keaslian sertifikat digital berbasis file PDF.

3.8 Desain Penelitian

Desain eksperimen disusun untuk memberikan alur sistematis dalam pengujian algoritma EdDSA dan SHA-256 terhadap berbagai ukuran dokumen digital. Tujuan utama dari eksperimen ini adalah untuk mengukur kinerja algoritma dalam menghasilkan dan memverifikasi tanda tangan digital secara efisien, aman, dan akurat. Pada tahap persiapan dataset, dilakukan proses pengumpulan dan klasifikasi dokumen PDF ke dalam tiga kategori ukuran — kecil, sedang, dan besar. Setiap kategori berisi beberapa sampel sertifikat digital dengan kompleksitas konten yang berbeda untuk

merepresentasikan kondisi nyata di lingkungan penggunaan dokumen digital. Berikut ini digambarkan pada gambar 3.1. dalam bentuk flowchart.



Gambar 3.1 Flowchart Desain Penelitian Algoritma EdDSA dan SHA−256

Tahap selanjutnya adalah proses hashing, di mana isi dokumen diekstraksi dan diubah menjadi nilai hash unik menggunakan fungsi hash SHA-256. Nilai hash ini berfungsi sebagai sidik jari digital yang akan digunakan dalam proses penandatanganan. Setelah nilai hash diperoleh, dilakukan proses signing dengan algoritma EdDSA menggunakan pasangan kunci privat dan publik. Proses ini menghasilkan signature file yang merupakan tanda tangan digital dari dokumen tersebut. Tahap berikutnya adalah proses verifikasi, di mana sistem menggunakan kunci publik untuk memverifikasi tanda tangan digital. Hasil verifikasi menunjukkan apakah dokumen masih otentik (*valid*) atau telah mengalami perubahan (*invalid*).

Kemudian dilakukan pengukuran parameter utama yang meliputi waktu tanda tangan (*signing time*), waktu verifikasi (*verification time*), ukuran tanda tangan (*signature size*), dan validitas tanda tangan (*validity*). Semua parameter ini dicatat dalam bentuk data kuantitatif untuk dianalisis lebih lanjut.

Tahap akhir adalah analisis hasil, di mana seluruh data hasil pengujian diolah untuk mengevaluasi performa algoritma EdDSA–SHA256. Analisis ini bertujuan untuk menilai pengaruh ukuran dokumen terhadap kecepatan proses, efisiensi ukuran tanda tangan, serta tingkat keandalan sistem dalam memverifikasi keaslian dokumen digital. Melalui desain eksperimen ini, penelitian diharapkan dapat memberikan bukti empiris mengenai efektivitas algoritma EdDSA–SHA256 dalam penerapan tanda tangan digital, khususnya pada verifikasi keaslian sertifikat digital berbasis file PDF.

3.9 Prosedur Penelitian

3.9.1 Persiapan Dataset dan Import Library

Pada tahap ini merupakan langkah awal dalam proses implementasi tanda tangan digital menggunakan algoritma EdDSA dengan kombinasi fungsi *hash* SHA-256 dengan menyiapkan N file PDF sertifikat digital dengan N = 60 dengan kriteria dokumen berdasarkan ukuran, yaitu 20 dokumen (100 - 500 KB), 20 dokumen (500 KB – 2 MB), dan 20 dokumen (2 – 5 MB). Langkah ini mempersiapkan seluruh library yang digunakan untuk proses tanda tangan, hashing, dan pengolahan data. Dalam konteks penelitian, modul `ed25519`, `hashes` dari *library* `cryptography` Python untuk membuat dan memverifikasi tanda tangan digital pada dokumen yang dapat dilihat pada gambar 3.2.

```
2   from cryptography.hazmat.primitives.asymmetric import ed25519
3   from cryptography.hazmat.primitives import hashes
4   from PyPDF2 import PdfReader
5   import hashlib, json, os, pandas as pd, qrcode, time
```

Gambar 3.2. Modul Import Library

Untuk menjalankan proses *hashing*, modul `hashlib,json,os, pandas,qrcode, time` untuk menghasilkan nilai hash SHA-256 yang akan diformat dalam bentuk JSON sehingga dataset sertifikat digital dapat dikelola ke file csv agar mudah dianalisis.

3.9.2 Membaca file PDF dan Ekstraksi Isi

Sebelum dilakukan proses *hashing*, dokumen sertifikat digital dalam format PDF terlebih dahulu diekstraksi untuk memperoleh data tekstual yang akan dijadikan represenasi digital dari isi dokumen tersebut. Ekstraksi dilakukan dengan memanfaatkan pustaka `PyPDF2`, yaitu modul Python yang berfungsi untuk membaca dan memanipulasi file PDF. Proses ini bertujuan untuk memastikan bahwa sistem hanya mengolah data yang benar-benar terdapat di dalam dokumen, sehingga integritas data dapat dijaga sebelum dilakukan proses verifikasi atau pembuatan tanda tangan digital.

```
def extract_text_from_pdf(pdf_path):
    reader = PdfReader(pdf_path)
    text = ""
    for page in reader.pages:
        text += page.extract_text()
    return text
```

Gambar 3.3. Modul Ekstraksi file PDF

Ditunjukkan pada gambar 3.3 bahwa fungsi ini diberi nama `extract_text_from_pdf()`, dengan parameter `pdf_path` yang menunjukkan lokasi file PDF yang akan diproses. Pada baris pertama, objek `PdfReader` digunakan untuk membuka dokumen PDF dan membaca struktur isinya. Selanjutnya, variabel `text` diinisialisasi sebagai string kosong untuk menampung hasil ekstraksi teks. Proses iterasi dilakukan terhadap setiap halaman dokumen melalui perulangan `for page in reader.pages:`. Pada setiap iterasi, metode `extract_text()` dipanggil untuk mengambil isi teks dari halaman tersebut. Seluruh teks yang diperoleh kemudian digabungkan ke dalam satu variabel menggunakan operator penambahan string (`+=`).

Hasil akhir dari proses ini dikembalikan dalam bentuk *string* teks penuh melalui perintah `return text`. Nilai inilah yang nantinya akan digunakan sebagai input untuk tahap *hashing* menggunakan algoritma SHA-256, sehingga setiap perubahan isi pada file PDF akan menghasilkan nilai hash yang berbeda.

3.9.3 Proses Hashing dengan SHA-256

Setelah teks berhasil diekstraksi dari dokumen PDF pada tahap sebelumnya, proses berikutnya adalah membentuk nilai hash digital menggunakan algoritma Secure Hash Algorithm 256-bit (SHA-256)**.** Tahapan ini berfungsi untuk menghasilkan identitas unik (digital fingerprint) dari isi dokumen, yang digunakan sebagai dasar dalam proses verifikasi dan penandatanganan digital menggunakan algoritma EdDSA.

Untuk setiap dokumen (file PDF), M: Hitung digest dokumen menggunakan SHA-256 dengan persamaan (1) :

$$h_m = \mathrm{H}_{256}(M) \qquad (1)$$

dimana :

h_m = Nilai hash (*message digest*) yang dihasilkan dari proses hashing
H_{256} = Fungsi hash dengan *output* sepanjang 256 bit
M = Data masuka (*message*) yang akan diproses

Pada persamaan ini menunjukkan bahwa nilai hash h_m diperoleh dengan menerapkan fungsi hash kriptografis SHA-256 terhadap pesan atau data masukan M. Fungsi hash SHA-256 merupakan algoritma yang mengubah data masukan dengan ukuran yang bervariasi menjadi nilai keluaran dengan panjang tetap sebesar 256 bit. Dalam konteks penelitian ini, dokumen PDF yang berperan sebagai data masukan (M) diproses menggunakan algoritma SHA-256 untuk menghasilkan nilai hash (h_m). Nilai hash tersebut berfungsi sebagai representasi digital yang unik dari isi dokumen.

Karakteristik utama dari fungsi hash kriptografis adalah sifatnya yang sensitif terhadap perubahan data (*avalanche effect*), di mana perubahan sekecil apa pun pada data masukan akan menghasilkan nilai hash yang berbeda secara signifikan. Pilihan SHA-256 pada level dokumen tidak bertentangan dengan penggunaan SHA-512 internal Ed25519 — SHA-512 tetap dipakai untuk operasi internal tanda tangan. Gunakan *canonicalization* (mis. strip metadata, normalisasi whitespace) pada M sebelum *hashing*, agar proses verifikasi tidak terganggu oleh metadata yang berubah. Proses hashing dilakukan dengan memanfaatkan modul `hashlib` dari pustaka standar Python, yang telah mendukung berbagai algoritma hash kriptografis, termasuk SHA-256. Adapun modul implementasi yang digunakan dapat ditunjukkan pada gambar 3.4 dalam potongan kode dibawah ini.

```python
import hashlib

def generate_sha256_hash(text):
    # Konversi teks menjadi byte
    text_bytes = text.encode('utf-8')

    # Membuat objek hash SHA-256
    hash_object = hashlib.sha256(text_bytes)

    # Menghasilkan nilai hash dalam format heksadesimal
    hash_hex = hash_object.hexdigest()

    return hash_hex
```

Gambar 3.4. Modul Hashing dengan SHA-256

Fungsi dengan nama `generate_sha256_hash()` yang menerima satu parameter, yaitu *text*. Parameter ini berisi teks hasil ekstraksi dari file PDF yang akan diubah menjadi nilai *hash* dan mengubah data teks menjadi bentuk *byte* menggunakan encoding UTF-8, karena algoritma hash hanya dapat memproses data dalam format biner. Proses hashing bertujuan untuk mengubah data teks hasil ekstraksi menjadi rangkaian karakter acak dengan panjang tetap (256 bit), yang merepresentasikan isi dokumen secara unik. Apabila terdapat perubahan sekecil apa pun pada dokumen (misalnya satu huruf atau spasi), maka hasil hash akan berubah sepenuhnya.

3.9.4 Pembuatan kunci EdDSA

Tahapan ini merupakan bagian dari proses pembuatan pasangan kunci (*key pair generation*) dalam algoritma tanda tangan digital EdDSA (Ed25519). Pasangan kunci ini terdiri dari kunci privat (*private key*) dan kunci publik (*public key*). Kunci privat berfungsi untuk membuat tanda tangan digital, sedangkan kunci publik digunakan untuk memverifikasi keaslian tanda tangan. Pada tahap ini, sistem menghasilkan sepasang kunci berupa *private key* (α) dan *public key* (A) berdasarkan kurva eliptik Edwards. Adapun persamaan pembentukan kunci :

$$A=[\alpha]G \tag{2}$$

dimana:

α = kunci privat, yaitu bilangan bulat acak yang dipilih dari himpunan bilangan modulo orde kurva eliptik ℓ

G = titik generator pada kurva eliptik yang telah ditentukan sebelumnya

A = kunci publik hasil perkalian skalar antara α dan G

Persamaan ini menunjukkan proses pembangkitan kunci publik pada algoritma tanda tangan digital berbasis kurva eliptik, seperti EdDSA. Dalam proses ini, sebuah bilangan acak α dipilih sebagai kunci privat yang bersifat rahasia dan hanya diketahui oleh pemilik sistem. Selanjutnya, kunci publik A dihasilkan melalui operasi perkalian skalar antara kunci privat α dan titik generator G pada kurva eliptik. Operasi perkalian skalar tersebut dilakukan dengan menjumlahkan titik generator G secara berulang sebanyak α kali sesuai dengan aturan aritmetika pada kurva eliptik. Hasil dari operasi tersebut adalah sebuah titik baru pada kurva eliptik yang dinyatakan sebagai kunci publik A. Kunci publik ini dapat dibagikan kepada pihak lain dan digunakan dalam proses verifikasi tanda tangan digital, sedangkan kunci privat α tetap dijaga kerahasiaannya oleh pemilik sistem.

Keamanan mekanisme ini didasarkan pada tingkat kesulitan komputasi dari permasalahan *Elliptic Curve Discrete Logarithm Problem* (ECDLP), yaitu kesulitan

dalam menentukan nilai α apabila hanya diketahui titik generator G dan kunci publik A. Oleh karena itu, hubungan matematis antara α dan A tidak dapat dengan mudah dibalikkan, sehingga menjamin keamanan sistem kriptografi yang digunakan. Dengan demikian, pembangkitan kunci ini menjadi dasar keamanan sistem tanda tangan digital karena keabsahan seluruh proses kriptografi bergantung pada keamanan pasangan kunci tersebut.

```
private_key = ed25519.Ed25519PrivateKey.generate()
public_key = private_key.public_key()
```

Gambar 3.5. Modul Sintaks Pembuatan kunci EdDSA

Metode `.generate()` yang ditunjukkan pada gambar 3.5 akan menghasilkan sebuah kunci privat acak yang hanya diketahui oleh pihak pembuat tanda tangan. Kunci privat inilah yang digunakan untuk mengenkripsi atau membuat tanda tangan digital pada dokumen. Dalam konteks kriptografi asimetris, keamanan sistem sangat bergantung pada kerahasiaan kunci privat ini — jika bocor, maka pihak lain dapat memalsukan tanda tangan. Dengan menurunkan modul *public_key* dari kunci privat yang telah dibuat sebelumnya. Kunci publik ini bersifat terbuka dan dapat disebarkan secara bebas kepada pihak lain. Fungsinya adalah untuk memverifikasi tanda tangan digital yang dihasilkan menggunakan kunci privat yang berpasangan dengannya.

3.9.5 Penandatanganan Digital

Tahap ini merupakan inti dari penelitian karena berhubungan langsung dengan proses autentikasi digital dokumen sertifikat. Setelah dokumen diubah menjadi nilai hash melalui algoritma SHA-256, langkah selanjutnya adalah membuat tanda tangan digital menggunakan algoritma EdDSA (Edwards-curve Digital Signature Algorithm) yang diimplementasikan melalui *Ed25519*.

Dalam beberapa implementasi, proses penandatanganan dapat menggunakan nilai hash pesan h_m yang sebelumnya dihasilkan melalui fungsi hash kriptografis. Proses ini menghasilkan tanda tangan digital yang dinotasikan sebagai:

$$\sigma=(R,s) \tag{3}$$

Dimana komponen *R* merupakan titik pada kurva eliptik yang dihasilkan dari perkalian skalar antara nilai *nonce r* dengan titik generator G. Nilai ini berfungsi sebagai bagian awal dari tanda tangan digital yang berkaitan dengan proses pembangkitan nilai acak deterministik. Sementara itu, komponen *s* merupakan nilai skalar yang dihitung berdasarkan kombinasi antara nilai *nonce*, nilai hash pesan, serta kunci privat yang digunakan dalam proses penandatanganan. Nilai ini berfungsi untuk mengikat pesan yang ditandatangani dengan identitas pemilik kunci privat. Dengan ukuran total 64 byte, di mana pasangan nilai tersebut akan digunakan dalam proses verifikasi tanda tangan digital. Adapun tahapan pembentukan tanda tangan digital sebagai berikut.

1. Perhitungan Nilai Hash dari Seed

   Langkah awal dengan menghitung nilai hash dari *seed* menggunakan fungsi hash SHA-512, yang dinyatakan dalam persamaan (4) :

$$h = H_{512}(\text{seed}) = h_0 \| h_1 \tag{4}$$

   Dimana :

   H_{512} : fungsi hash SHA-512 dengan keluaran sepanjang 512 bit

   h : hasil hashing dari *seed*

   h_0 dan h_1 : merupakan dua bagian dari hasil hash

   Pada tahap ini, bagian h_1 digunakan sebagai *prefix* untuk membentuk nilai *nonce* secara deterministik. Penggunaan *nonce* deterministik bertujuan untuk meningkatkan keamanan dengan menghindari penggunaan bilangan acak yang berpotensi menyebabkan kerentanan jika dihasilkan secara tidak aman.

2. Pembentukan *nonce* deterministik (*r*), setelah memperoleh nilai *prefix* h_1 nilai *nonce* deterministik *r* dapat dihitung menggunakan persamaan (5) :

$$r = \text{LE\_to\_int}\,(H_{512}(h_1 \,||\, h_m)) \bmod \ell \qquad (5)$$

Dimana :

h_1 : bagian hash dari *seed* yang digunakan sebagai *prefix*

h_m : nilai hash dari pesan atau dokumen

H_{512} : fungsi hash SHA-512

ℓ : orde dari grup pada kurva eliptik yang digunakan

LE_to_int : proses konversi dari representasi *little-endian* ke bilangan bulat

Proses ini menghasilkan nilai *nonce* deterministik *r* yang akan digunakan pada tahap berikutnya dalam pembentukan komponen tanda tangan digital. Penggunaan *nonce* deterministik memastikan bahwa nilai *nonce* dihasilkan secara konsisten dari kunci privat dan pesan, sehingga mengurangi risiko kebocoran kunci privat akibat penggunaan *nonce* yang lemah atau berulang.

Berikut potongan fungsi penandatanganan digital dokumen dapat dilihat pada gambar 3.6.

```
def sign_hash(private_key, hash_value):
    signature = private_key.sign(hash_value.encode('utf-8'))
    return signature.hex()
```

Gambar 3.6. Modul Penandatanganan Dokumen

Modul sintaks `signature = private_key.sign(hash_bytes)` merupakan proses utama pembuatan tanda tangan digital. Metode `.sign()` akan menghasilkan data kriptografis unik berdasarkan nilai hash dan kunci privat, sehingga hasil tanda tangan hanya dapat diverifikasi menggunakan kunci publik yang sesuai dan mengubah nilai *hash* ke dalam format *byte* karena algoritma kriptografi memerlukan input biner, bukan teks biasa dengan modul `hash_bytes = hash_value.encode('utf-8')`.

3.9.6 Verifikasi Dokumen

Tahap ini merupakan bagian akhir dalam sistem tanda tangan digital menggunakan algoritma EdDSA (Ed25519) dan SHA-256, yang berfungsi untuk memverifikasi keaslian dan integritas dokumen**.** Verifikasi tanda tangan digital dilakukan untuk memastikan bahwa:

1. Dokumen benar-benar ditandatangani oleh pemilik kunci privat yang sah (authenticity); dan
2. Isi dokumen tidak mengalami perubahan setelah ditandatangani (integrity).

Proses verifikasi dilakukan menggunakan kunci publik (*public key*) yang telah dihasilkan pada tahap sebelumnya. Dalam sistem tanda tangan digital asimetris, kunci publik tidak bersifat rahasia dan dapat dibagikan kepada pihak penerima dokumen untuk memastikan validitas tanda tangan tanpa mengungkapkan kunci privat. Berikut dapat dilihat pada gambar 3.7 pada implementasi fungsi Python untuk proses verifikasi tanda tangan digital menggunakan *library cryptography*.

```
def verify_signature(public_key, hash_value, signature_hex):
    signature = bytes.fromhex(signature_hex)
    try:
        public_key.verify(signature, hash_value.encode('utf-8'))
        return True
    except Exception:
        return False
```

Gambar 3.7. Modul Verifikasi Dokumen

Fungsi `hash_bytes = hash_value.encode('utf-8')` Mengubah nilai hash dari format string menjadi format *byte*, karena algoritma kriptografi bekerja pada data biner, bukan teks biasa. Langkah ini memastikan format data kompatibel dengan fungsi verifikasi dari *library* Ed25519.

Inti dari proses verifikasi terdapat pada fungsi `.verify()` yang digunakan untuk memeriksa apakah tanda tangan digital (*signature*) benar-benar dihasilkan oleh kunci privat yang berpasangan dengan kunci publik yang digunakan. Jika proses

verifikasi berhasil ditandai dengan dokumen belum diubah sejak ditandatangani dan tanda tangan digital valid serta berasal dari pemilik kunci privat yang sah. Jika terjadi kesalahan (misalnya tanda tangan tidak cocok atau dokumen diubah), program akan menangkap *exception* dan menampilkan pesan bahwa verifikasi gagal. Hal ini menunjukkan bahwa dokumen telah dimodifikasi, atau tanda tangan tidak dibuat menggunakan kunci privat yang sesuai.

3.9.7 Analisis dan Penyimpanan Hasil

Ini merupakan bagian dari tahap pengujian sistem secara komputasional dalam penelitian ini. Tujuannya adalah untuk mengotomatisasi proses pengujian tanda tangan digital pada sejumlah dokumen sertifikat digital berformat PDF. Tahap ini melibatkan beberapa proses utama yang dijalankan secara berurutan untuk setiap file di dalam folder *dataset*, yaitu:

1. Ekstraksi teks dari dokumen PDF.
2. Pembuatan nilai hash (SHA-256).
3. Proses tanda tangan digital menggunakan EdDSA (Ed25519).
4. Verifikasi tanda tangan digital menggunakan kunci publik.
5. Penyimpanan metadata hasil pengujian.
6. Perekaman waktu eksekusi (execution time) untuk menilai efisiensi algoritma.

Hasil pengujian disimpan dalam bentuk DataFrame Pandas, kemudian diekspor menjadi file CSV (hasil_pengujian.csv) agar dapat digunakan untuk analisis performa dan validasi hasil. Pendekatan ini termasuk ke dalam metode eksperimen komputasional yang bersifat kuantitatif, karena melibatkan pengukuran waktu eksekusi dan tingkat keberhasilan verifikasi. Dalam tahap akhir prosedur penelitian, mengeksekusi seluruh proses untuk semua file PDF dalam dataset, kemudian menyimpan hasilnya dalam bentuk CSV, potongan fungsi kode dapat dilihat pada gambar 3.8.

```
results = []
for file in os.listdir("dataset"):
    if file.endswith(".pdf"):
        text = extract_text_from_pdf(os.path.join("dataset", file))
        h = hash_document(text)
        start = time.time()
        s = sign_hash(private_key, h)
        verify_result = verify_signature(public_key, h, s)
        elapsed = (time.time() - start) * 1000  # dalam milidetik
        save_metadata(file, s, h)
        results.append({"file": file, "hash": h, "verified": verify_result, "time_ms": elapsed})

df = pd.DataFrame(results)
df.to_csv("hasil_pengujian.csv", index=False)
```

Gambar 3.8. Modul Analisis dan Penyimpanan Tanda Tangan Digital

3.10 Teknik Analisis Data

Data yang diperoleh dianalisis dengan statistik deskriptif kuantitatif, yaitu menghitung rata-rata, minimum, maksimum, dan standar deviasi dari parameter yang diukur. Hasil pengujian kemudian dibandingkan antar sampel dokumen untuk mengetahui pengaruh ukuran file terhadap performa algoritma. Analisis juga disajikan dalam bentuk tabel dan grafik agar lebih mudah dipahami (Field, 2021). Analisis data dalam penelitian ini dilakukan secara kuantitatif, bertujuan untuk mengukur dan mengevaluasi kinerja algoritma EdDSA–SHA256 berdasarkan hasil pengujian terhadap 60 dokumen PDF dengan ukuran yang berbeda. Analisis dilakukan melalui dua pendekatan utama, yaitu analisis statistik deskriptif dan analisis inferensial, disertai evaluasi terhadap validitas hasil tanda tangan digital.

3.10.1 Statistik Deskriptif

Menurut (Sugiyono, 2020) Statistik deskriptif merupakan cabang ilmu statistik yang berfungsi untuk menganalisis serta menyajikan data dengan cara menggambarkan atau mendeskripsikan karakteristik data yang diperoleh sebagaimana adanya, tanpa melakukan penarikan kesimpulan atau generalisasi terhadap populasi secara keseluruhan. Analisis deskriptif digunakan untuk memberikan gambaran umum mengenai distribusi dan karakteristik data hasil pengukuran. Parameter statistik yang dihitung mencakup:

1. Mean (rata-rata), untuk mengetahui nilai tengah dari waktu penandatanganan (*signing time*), waktu verifikasi (*verification time*), dan ukuran tanda tangan (*signature size*). Untuk menghitung rata-rata menggunakan persamaan (6) :

$$\bar{X} = \frac{\sum_{i=1}^{n} X_i}{n} \quad (6)$$

Dimana :

$\bar{X}$ = nilai rata-rata (mean)

X_i = data ke-i

n = jumlah total data (sampel)

Nilai rata-rata menunjukkan kecenderungan sentral dari data, yakni nilai yang paling mewakili keseluruhan hasil pengamatan. Dalam konteks penelitian ini, mean digunakan untuk mengetahui rata-rata waktu kesekusi proses tanda tangan digital EdDSA maupun proses verifikasi SHA-256 terhadap sejumlah dokumen PDF.

2. Median, adalah nilai tengah dari data yang telah diurutkan dari yang terkecil hingga terbesar. Berikut persamaan untuk menghitung median :

$$\text{Median} = \begin{cases} X_{\frac{n+1}{2}}, & \text{Jika } n \text{ ganjil} \\ \frac{X_{\frac{n}{2}} + X_{\frac{n}{2}+1}}{2}, & \text{Jika } n \text{ genap} \end{cases} \quad (7)$$

3. Modus, untuk menunjukkan nilai yang paling sering muncul pada hasil pengujian. Berikut persamaan untuk menghitung modus :

$$\text{Modus} = X_i \text{ dengan frekuensi kemunculan tertinggi} \quad (8)$$

Dalam konteks sistem tanda tangan digital, modus dapat menunjukkan nilai waktu atau ukuran tanda tangan yang paling sering muncul, yang berarti kondisi tersebut merupakan performa paling umum dari sistem dalam lingkungan pengujian yang stabil (Sugiyono, 2020)

4. Standar deviasi (SD), adalah akar dari varians yang menunjukkan tingkat variasi atau penyimpangan data dari nilai rata-rata dalam satuan yang sama dengan data aslinya. Rumus menghitung standar deviasi dapat dilihat pada persamaan (9) :

$$S = \sqrt{\frac{\sum_{i=1}^{n} (X_i - \bar{X})2}{n-1}} \quad (9)$$

Keterangan :

$\bar{X}$ = nilai rata-rata (mean)

X_i = data ke-i

n = jumlah total data (sampel)

Dalam konteks sistem tanda tangan digital, standar deviasi waktu penandatanganan yang kecil menandakan bahwa algoritma EdDSA memberikan hasil yang konsisten dan stabil pada setiap uji coba, terlepas dari ukuran atau isi dokumen PDF.

BAB IV
HASIL DAN PEMBAHASAN

4.1 Persiapan Dataset

Persiapan dataset merupakan tahapan awal yang penting dalam pengujian kinerja *Edwards-curve Digital Signature Algorithm* (EdDSA) pada sistem tanda tangan digital. Dataset yang digunakan harus mampu merepresentasikan karakteristik dokumen digital yang digunakan agar hasil pengujian dapat menggambarkan performa sistem secara objektif dan realistis. Dataset pada penelitian ini berupa dokumen digital berformat PDF. Format PDF dipilih karena merupakan standar dokumen elektronik yang paling banyak digunakan dalam pertukaran dokumen resmi serta memiliki struktur data yang stabil, sehingga sesuai untuk diterapkan pada sistem tanda tangan digital berbasis kriptografi kunci publik.

Dokumen dataset disusun dengan variasi ukuran *file* yang dapat dilihat pada tabel 4.1 dan dikelompokkan ke dalam tiga kategori, yaitu, sedang, besar. Pengelompokan ini bertujuan untuk menganalisis pengaruh terhadap waktu proses *hashing*, waktu proses penandatanganan (*signing*), ukuran dokumen, dan verifikasi tanda tangan digital.

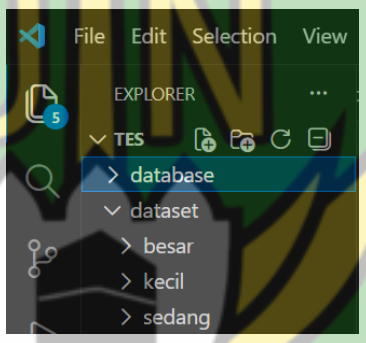
Tabel 4.1. Klasifikasi Dataset

Kategori Ukuran	Rentang Ukuran File	Jumlah File	Format	Keterangan
Kecil	100 – 500 KB	20 file	PDF	Dokumen teks sederhana
Sedang	500 KB – 2 MB	20 file	PDF	Dokumen teks dan gambar
Besar	2 – 5 MB	20 file	PDF	Dokumen kompleks berisi grafik dan gambar
Total		**60 file**	**PDF**	**Pengujian dataset**

Berdasarkan Dataset pengujian pada tabel 4.1 yang digunakan dalam penelitian ini terdiri dari 60 dokumen digital berformat PDF yang diklasifikasikan berdasarkan

ukuran *file* dalam tiga kategori, yaitu dokumen berukuran kecil, sedang, dan besar. Sebagaimana ditunjukkan pada tabel 4.1, masing-masing kategori terdiri atas 20 dokumen dengan rentang ukuran yang berbeda. Dokumen berukuran kecil berada pada rentang 100-500 KB dan umumnya berisi teks sederhana. Dokumen berukuran sedang memiliki rentang ukuran antara 500 KB hingga 2 MB yang berisi kombinasi teks dan gambar. Sementara dokumen berukuran besar berada pada rentang 2-5 MB dan berisi elemen data yang lebih kompleks seperti grafik resolusi tinggi dan gambar.

Pembagian dataset ke dalam kategori ini yang dapat dilihat pada gambar 4.1 memungkinkan analisis yang lebih komprehensif terhadap kinerja sistem, khususnya dalam mengamati kecendrungan peningkatan waktu proses seiring dengan adanya perubahan ukuran dokumen.



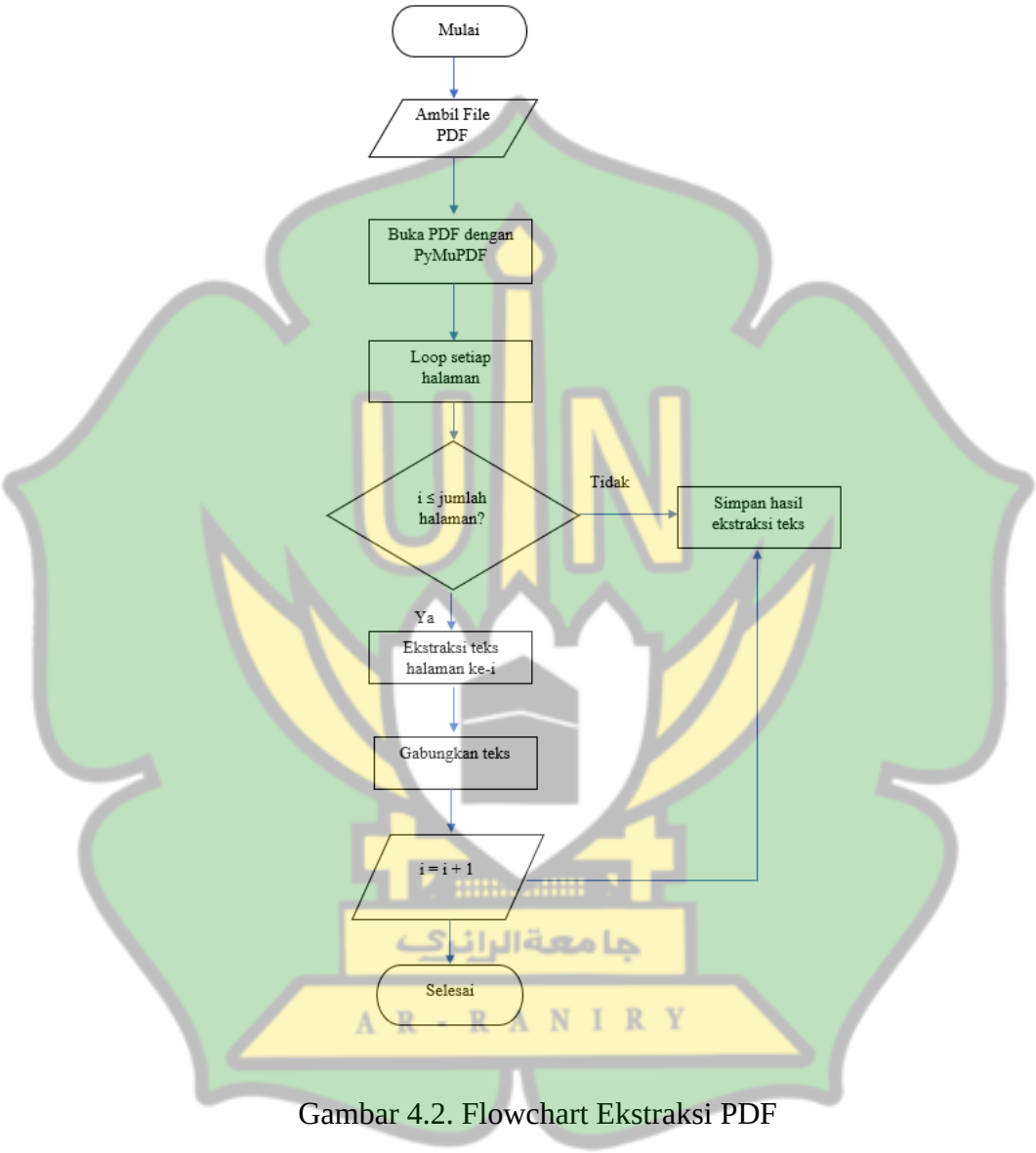
Gambar 4.1 Direktori dataset dan database pengujian

Dari gambar 4.1 menunjukkan direktori dataset yang disusun ke dalam subfolder kecil, sedang , dan besar berdasrkan ukuran dokumen. Pengelompokan ini bertujuan untuk mempermudah pengelolaan data, otomasi pengujian, serta analisis kinerja algoritma EdDSA terhadap variasi ukuran dokumen. Dengan demikian, hasil pengujian diharapkan dapat memberikan gambaran objektif mengenai skalabilitas dan efisiensi algoritma EdDSA dalam menangani variasi beban data.

4.2 Ekstraksi Dokumen

Tujuan dari ekstraksi dokumen ialah untuk mengambil konten teks dari setiap *file* PDF sebagai bahan utama dalam proses *hashing* menggunakan algoritma SHA-256. Hasil ekstraksi ini digunakan sebagai input pada proses penandatanganan digital

menggunakan algoritma EdDSA. Flowchart tahapan ekstraksi dokumen dilakukan menggunakan bahasa pemograman Ptython dengan bantuan *library* PyMuPDF yang dapat dilihat pada Gambar 4.2.



Gambar 4.2. Flowchart Ekstraksi PDF

Flowchart pada gambar 4.2 menunjukkan proses ekstraksi teks dari dokumen PDF menggunakan pustaka PyMuPDF. Proses diawali dengan inisialisasi sistem dan pengambilan file PDF sebagai data masukan. Selanjutnya, sistem membuka dokumen menggunakan PyMuPDF untuk membaca struktur dokumen serta mengakses setiap

halaman yang tersedia. Setelah dokumen berhasil dibuka, sistem melakukan proses iterasi untuk memproses seluruh halaman. Pada setiap iterasi, sistem memeriksa apakah indeks halaman iii masih berada dalam batas jumlah halaman dokumen. Jika kondisi terpenuhi, sistem mengekstraksi teks dari halaman ke-iii dan menggabungkannya dengan hasil ekstraksi dari halaman sebelumnya.

Setelah proses ekstraksi pada suatu halaman selesai, nilai indeks halaman ditambah satu (i=i+1)(i = i + 1)(i=i+1) untuk melanjutkan proses pada halaman berikutnya. Proses ini berlangsung hingga seluruh halaman selesai diproses. Ketika indeks halaman melebihi jumlah halaman dokumen, proses iterasi dihentikan dan hasil akhir ekstraksi teks dari seluruh dokumen disimpan..

```
import os
import fitz  # PyMuPDF

import time
import logging
from tqdm import tqdm
```

Gambar 4.3. Source Code Library PyMuPDF

Dari gambar 4.3 dapat dilihat bahwa pada tahap pengujian sistem, digunakan *library* os untuk manajemen dokumen dan direktori, fitz (PyMuPDF) untuk ekstraksi isi dokumen PDF, serta tqdm untuk menampilkan progress pengujian. Kombinasi *library* ini memungkinkan proses ekstraksi, pengujian, dan pencatatan data dilakukan secara otomatis, sistematis, dan terkontrol. Pada tahap implementasi sistem, dilakukan penentuan direktori dasar sebagai acuan pengelolaan *file* menggunakan fungsi os.path.abspath(_file_) yang berfungsi untuk mengatur lokasi direktori *input* dataset dan direktori *output* hasil pemrosesan. Potongan fungsi dapat dilihat pada gambar 4.4.

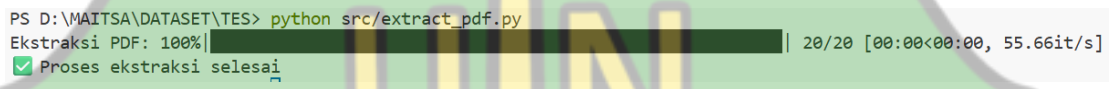
```python
BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))

INPUT_FOLDER = os.path.join(BASE_DIR, "dataset", "kecil")
OUTPUT_TEXT = os.path.join(BASE_DIR, "output", "text")
OUTPUT_LOG = os.path.join(BASE_DIR, "output", "log")

os.makedirs(OUTPUT_TEXT, exist_ok=True)
os.makedirs(OUTPUT_LOG, exist_ok=True)
```

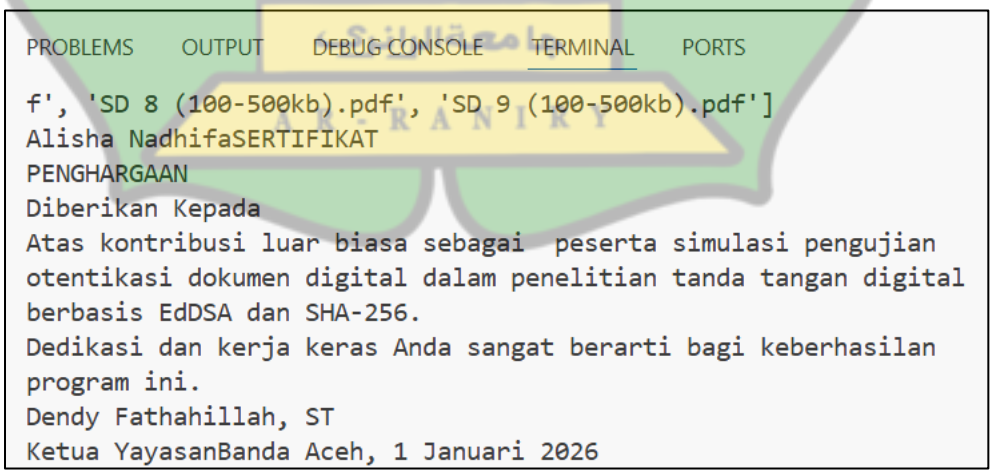
Gambar 4.4. Potongan Source Code Penyimpanan Direktori

Hasil eksekusi modul ekstraksi dokumen PDF yang dijalankan melalui terminal menggunakan perintah python src/exctract_pdf.py dapat ditunjukkan pada Gambar 4.5 berikut ini.

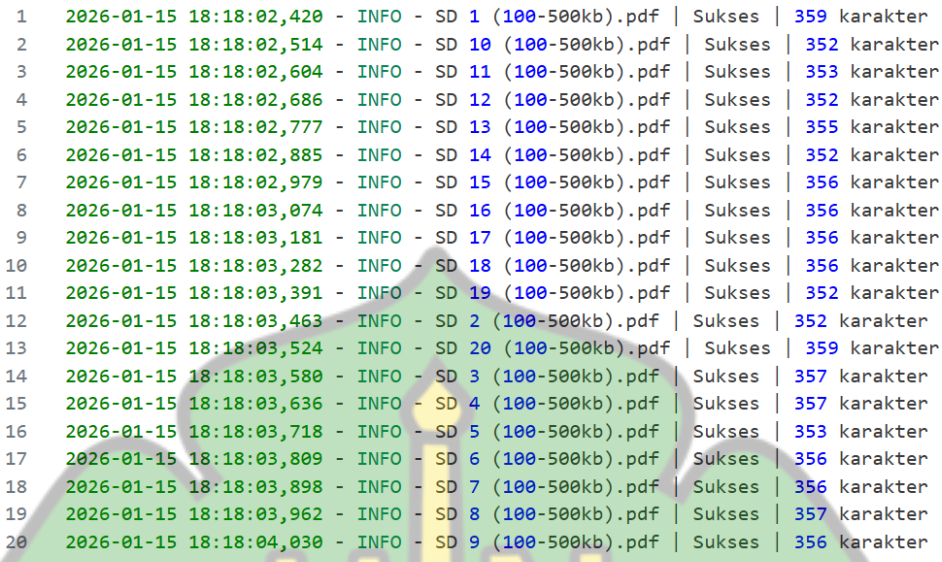


Gambar 4.5. Hasil Eksekusi Modul Ekstraksi Dokumen

Proses ekstraksi dilakukan per batch terhadap 20 dokumen PDF dalam dataset, yang ditandai dengan indicator progress mencapai 100%. Seluruh dokumen berhasil diproses tanpa terjadi kesalahan, dengan kecepatan pemrosesan rata-rata sebesar 55,66 *file* per detik. Proses ekstraksi berhasil dinyatakan berhasil dibuktikan dengan terbacanya isi salah satu dokumen PDF yang dapat ditunjukkan pada Gambar 4.6 dibawah ini.



Gambar 4.6. Hasil Isi Dokumen PDF



Gambar 4.7. Hasil Log Dokumen PDF

Berdasarkan gambar yang ditunjukkan pada Gambar 4.7, hasil ekstraksi menunjukkan bahwa seluruh *file* pdf dapat diekstraksi dengan baik dengan tingkat keberhasilan mencapai 100%. Seluruh konten teks berhasil diambil tanpa kehilangan karakter penting, sehingga dapat digunakan sebagai *input* hashing.

4.3. Pengujian dan Analisis Kinerja Algoritma

4.3.1. Hasil Pengujian Hashing

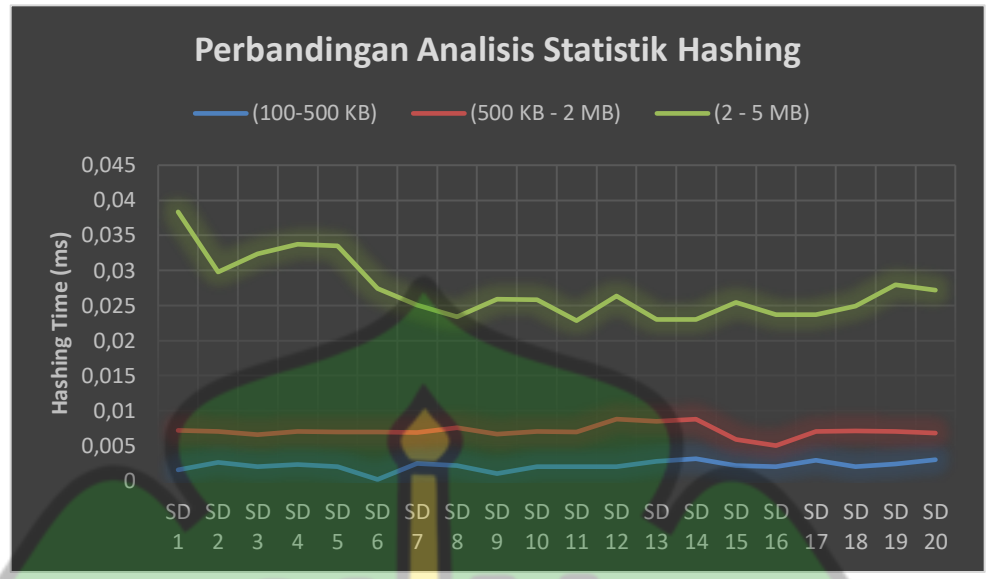
Pengujian ini bertujuan untuk menganalisis pengaruh ukuran dokumen terhadap waktu proses hashing serta mengamati karakteristik nilai hash yang dihasilkan. Berdasarkan hasil pengujian terhadap enam dokumen PDF dengan variasi ukuran antara 385 KB hingga 3953 KB menggunakan algoritma SHA-256, diperoleh bahwa seluruh dokumen menghasilkan nilai hash dengan panjang tetap sebesar 256 bit (32 bytes). Berbeda dengan panjang nilai hash, waktu proses hashing menunjukkan adanya variasi yang dipengaruhi oleh ukuran dokumen. Semakin besar ukuran *file*, semakin lama waktu yang dibutuhkan untuk menyelesaikan proses hashing.

Tabel 4.2. Hasil Pengujian Hashing

No	File	Hash Value	File Size	Hashing Time (ms)
1	SD 12 (100-500kb).pdf	2358c9de98…e0fc922242ae9e8c	385 KB	0.001009
2	SD 13 (100-500kb).pdf	1285593c…4a2753efd01fa107d	446 KB	0.000998
3	SD 2 (500KB-2MB).pdf	449434c964d2c…c4a9e5b2121	1112 KB	0.100376
4	SD 3 (500KB-2MB).pdf	19a2e285c7be…60b7c242df5	1111 KB	0.080564
5	SD 10 (2-5 MB).pdf	809b4099a…c3dcfb0b03d23e72	3953 KB	0.025861
6	SD 11 (2-5 MB).pdf	be8834d5fb…0ec2d337b9ab8fa0	2985 KB	0.022843

Berdasarkan hasil pengujian pada tabel 4.2, Temuan ini menegaskan karakteristik fundamental SHA-256 yang memiliki *fixed output length*, sehingga ukuran data masukan tidak memengaruhi panjang nilai hash yang dihasilkan. Dari sisi waktu proses hashing, dokumen berukuran kecil (100-500 KB) memiliki waktu hashing yang relatif sangat cepat, yaitu sekitar 0,000998–0,001009 milidetik. Pada kategori ukuran sedang (500 KB – 2 MB), waktu hashing cenderung meningkat, yaitu 0,080564–0,100376 milidetik. Sementara itu, pada dokumen berukuran besar (2-5 MB), waktu hashing tercatat sebesar 0,022843–0,025861 milidetik.

Untuk menganalisis pengaruh ukuran dokumen terhadap waktu hashing, dilakukan perhitungan statistik deskriptif pada tiga kategori ukuran file. Ditunjukkan dalam grafik yang menyajikan analisis waktu proses *hashing* terhadap 20 sampel data (SD 1 hingga SD 20) yang dikategorikan ke dalam tiga kelompok ukuran berkas berbeda. Data visual grafik ditunjukkan pada gambar 4.8.



Gambar 4.8. Grafik Analisis Statistik Hashing

Berdasarkan grafik tersebut, hasil analisis waktu proses (*hashing time*) dapat dikategorikan sebagai berikut:

- **Data Berukuran Kecil (100–500 KB):** Menunjukkan performa paling stabil dengan rata-rata waktu proses di bawah 0,005 ms. Hal ini mengindikasikan beban komputasi yang sangat rendah pada kapasitas memori minimal.
- **Data Berukuran Sedang (500 KB – 2 MB):** Terjadi peningkatan durasi pemrosesan yang konsisten pada rentang 0,005 ms hingga 0,01 ms, namun tetap mempertahankan grafik yang linear tanpa fluktuasi tajam.
- **Data Berukuran Besar (2–5 MB):** Memerlukan waktu komputasi tertinggi dengan fluktuasi yang lebih dinamis. Titik tertinggi tercatat pada SD 1 sebesar ≈ 0,038 ms, sebelum kemudian stabil pada kisaran 0,023 ms hingga 0,028 ms.

Ini membuktikan bahwa sistem mampu menangani peningkatan beban data secara responsif, meski beban komputasi menjadi lebih sensitif terhadap variasi konten pada berkas di atas 2 MB.

Adapun parameter yang dianalisis meliputi jumlah sampel, nilai minimum, maksimum, rentang (modus), rata-rata (mean), median, dan standar deviasi. Analisis ini bertujuan untuk memberikan gambaran kuantitatif mengenai kecenderungan waktu komputasi serta tingkat kestabilan sistem pada setiap kategori ukuran dokumen. Berikut ini ditunjukkan pada tabel 4.3 yang menyajikan ringkasan statistik deskriptif dari seluruh hasil pengujian sebagai dasar untuk mengevaluasi tren kenaikan waktu hashing dan konsistensi performa sistem.

Parameter	**100–500 KB**	**500 KB–2 MB**	**2–5 MB**
Jumlah Sampel (n)	20	20	20
Minimum (ms)	0,000195	0,005020	0,022843
Maksimum (ms)	0,003152	0,008800	0,038344
Rentang (modus) (ms)	0,002957	0,003780	0,015501
Rata-rata (Mean) (ms)	0,002138	0,007197	0,026857
Median (ms)	0,002040	0,007029	0,025700
Standar Deviasi (ms)	0,000708	0,000990	0,004340

Tabel. 4.3. Analisis Statistik Waktu Hashing

Terjadi peningkatan waktu *hashing* yang signifikan seiring bertambahnya ukuran dokumen :

- 100–500 KB → ± 0,002 milidetik
- 500 KB–2 MB → ± 0,007 milidetik
- 2–5 MB → ± 0,027 milidetik

Artinya :

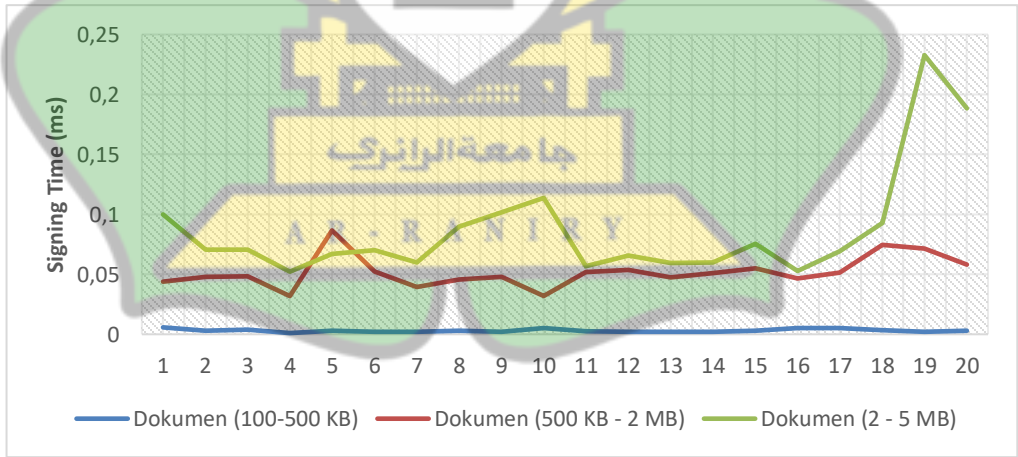
- Dokumen ukuran 500 KB – 2 MB ≈ 3,3 kali lebih lambat dari dokumen ukuran 100 – 500 KB
- Dokumen ukuran 2 – 5 MB ≈ 3,7 kali lebih lambat dari dokumen ukuran 500 KB – 2 MB

- Dokumen ukuran 2 – 5 MB ≈ 12 kali lebih lambat dari dokumen ukuran 100 – 500 KB

Hal ini menunjukkan hubungan linier positif antara ukuran dokumen dan waktu hashing. bahwa waktu hashing meningkat secara konsisten seiring bertambahnya ukuran dokumen. Hal ini sesuai dengan kompleksitas algoritma hashing yang bergantung pada panjang input data (O(n)). Meskipun terjadi peningkatan waktu, seluruh proses hashing masih berada dalam rentang milidetik hingga puluhan milidetik, sehingga sistem dapat dikategorikan efisien dan layak digunakan untuk pengamanan dokumen digital.

4.3.2. Hasil Pengujian Signing

Pengujian signing dilakukan untuk mengukur kinerja proses penandatanganan digital terhadap dokumen PDF yang telah melalui tahap hashing menggunakan algoritma SHA-256 dan memastikan tanda tangan digital berhasil dihasilkan dan dapat diverifikasi. Proses penandatanganan dilakukan menggunakan algoritma EdDSA dengan skema Ed25519. Berikut grafik analisis waktu *signing* berdasarkan tiga kategori ukuran dokumen yang ditunjukkan pada gambar 4.9



Gambar 4.9. Grafik Analisis Waktu Signing Berdasarkan Ukuran Dokumen

Berdasarkan pada gambar 4.9 grafik menunjukkan bahwa, waktu signing berada dalam rentang milidetik dan relatif stabil pada seluruh kategori ukuran dokumen. Secara umum diperoleh karakteristik sebagai berikut:

- Waktu signing tidak meningkat secara signifikan seiring bertambahnya ukuran file.
- Variasi waktu antar dokumen relatif kecil.
- Tidak ditemukan perbedaan ekstrem seperti pada proses hashing.

Hal ini terjadi karena proses signing tidak dilakukan langsung terhadap seluruh isi dokumen, melainkan terhadap nilai hash yang memiliki panjang tetap (256-bit). Dengan demikian, kompleksitas waktu signing lebih bergantung pada operasi kriptografi kurva eliptik dibandingkan ukuran file.

Berikut analisis statistik deskriptif hasil pengujian *signing* EdDSA berdasarkan tiga kategori ukuran dokumen yang diuji, dapat dilihat pada tabel 4.4 sebagai berikut.

Tabel 4.4. Analisis Statistik Pengujian Signing

Parameter	100–500 KB	500 KB–2 MB	2–5 MB
Jumlah Sampel	20	20	20
Mean (Rata-rata)	0,003030 ms	0,051822 ms	0,087460 ms
Median	0,002991 ms	0,049514 ms	0,070282 ms
Modus	0,002 ms	0,048 ms	0,070 ms
Standar Deviasi	0,001320 ms	0,013183 ms	0,046059 ms

Berdasarkan hasil analisis statistik deskriptif terhadap waktu proses digital signing menggunakan algoritma EdDSA, terlihat adanya peningkatan waktu proses seiring bertambahnya ukuran dokumen, yakni :

- **Rata-rata (mean):** Menunjukkan bahwa dokumen berukuran 100–500 KB memiliki waktu signing paling cepat yaitu sekitar 0,003 milidetik, sedangkan

dokumen 500 KB–2 MB membutuhkan waktu rata-rata sekitar 0,051 milidetik, dan dokumen 2–5 MB sekitar 0,087 milidetik.

- **Median:** Ketiga kategori memiliki nilai yang relatif dekat dengan rata-rata, yang menunjukkan bahwa distribusi data relatif stabil dan tidak terlalu dipengaruhi oleh nilai ekstrem.
- **Modus:** Menunjukkan waktu yang paling sering muncul pada masing-masing kategori pengujian. Nilai ini juga memperlihatkan kecenderungan waktu signing meningkat seiring bertambahnya ukuran file.
- **Standar deviasi:** Dokumen berukuran kecil memiliki variasi waktu yang sangat rendah (0,001320 milidetik), sedangkan pada dokumen yang lebih besar variasinya meningkat hingga 0,046059 milidetik, yang menandakan bahwa waktu signing menjadi lebih bervariasi pada ukuran dokumen yang lebih besar.

Secara keseluruhan, hasil analisis statistik deskriptif menunjukkan bahwa baik proses hashing maupun signing mengalami peningkatan waktu komputasi seiring dengan bertambahnya ukuran dokumen. Namun demikian, nilai standar deviasi pada setiap kategori relatif kecil dibandingkan dengan nilai rata-ratanya, yang menunjukkan bahwa performa sistem dalam melakukan hashing dan penandatanganan digital cenderung stabil dan konsisten pada berbagai ukuran dokumen. Temuan ini menunjukkan bahwa kombinasi algoritma SHA-256 dan EdDSA mampu memberikan kinerja yang efisien dalam menjaga integritas dan autentikasi dokumen digital.

4.3.3. Hasil Pengujian Verifikasi Dokumen

Pengujian verifikasi dokumen dilakukan untuk memastikan bahwa tanda tangan digital yang dihasilkan dapat diverifikasi dengan benar menggunakan kunci publik yang sesuai. Proses verifikasi dilakukan dengan membandingkan nilai hash dokumen yang dihitung kembali menggunakan algoritma SHA-256 dengan nilai hash yang telah ditandatangani menggunakan algoritma EdDSA. Pengujian mencakup tiga scenario utama : dokumen asli, dokumen yang dimodifikasi, dan signature/hash tidak sesuai.

Skenario 1 : Dokumen Asli

Berdasarkan hasil pengujian, dokumen yang masih dalam kondisi asli berhasil diverifikasi oleh sistem dengan status **valid**, yang menunjukkan bahwa nilai hash dokumen sesuai dengan signature yang tersimpan pada kode QR. Hal ini membuktikan bahwa integritas dan keaslian dokumen tetap terjaga karena tidak terdapat perubahan pada isi dokumen setelah proses penandatanganan digital.

Tabel 4.5. Pengujian Dokumen Asli

No	QR Code	Nama File	Status Dokumen	Status QR Code	Status Verifikasi	Ket
1		SD 1 (100-500 kb).pdf	Asli	Asli	Valid	Signature sesuai dengan hash dokumen
2		SD 1 (500KB-2MB).pdf	Asli	Asli	Valid	Hash dan signature cocok
3		SD 1 (2-5MB).pdf	Asli	Asli	Valid	Dokumen sesuai signature

Pada table 4.5 tersebut menyajikan hasil pengujian verifikasi dokumen menggunakan *QR Code* pada beberapa dokumen PDF dengan variasi ukuran *file*. Pengujian dilakukan terhadap tiga kategori ukuran dokumen, yaitu kategori kecil (100–500 KB), kategori sedang (500 KB–2 MB), dan kategori besar (2–5 MB). Setiap dokumen yang diuji merupakan dokumen asli yang telah ditandatangani secara digital.

Hasil verifikasi menunjukkan bahwa seluruh dokumen memiliki status dokumen asli dan status *QR Code* asli, sehingga proses verifikasi menghasilkan status *valid*. Hal ini menunjukkan bahwa nilai hash dokumen yang dihasilkan sesuai dengan tanda tangan digital (*signature*) yang tersimpan dalam sistem. Dengan demikian, dapat disimpulkan bahwa dokumen tidak mengalami perubahan atau manipulasi setelah proses penandatanganan digital dilakukan.

Skenario 2 : Dokumen Yang Dimodifikasi

Berdasarkan hasil pengujian, dokumen yang telah dimodifikasi setelah proses penandatanganan menghasilkan status verifikasi **invalid**, meskipun kode QR masih dalam kondisi asli. Perubahan isi dokumen menyebabkan nilai hash yang dihasilkan berbeda dari nilai hash saat proses penandatanganan. Hal ini menunjukkan bahwa sistem mampu mendeteksi perubahan pada dokumen sehingga integritas dokumen tidak lagi terjamin. Dapat dilihat pada tabel 4.6

Tabel 4.6. Pengujian Dokumen Yang Dimodifikasi

No	QR Code	Nama File	Status Dokumen	Status QR Code	Status Verifikasi	Ket
1		SD 1 (100-500 kb).pdf	Diubah	Asli	Invalid	Isi dokumen dimodifikasi
2		SD 1 (500KB-2MB).pdf	Dokumen Lain	Asli	Invalid	QR tidak sesuai dokumen

Tabel 4.6 menampilkan hasil pengujian verifikasi pada dokumen yang mengalami perubahan atau tidak sesuai dengan *QR Code*. Pengujian ini bertujuan untuk mengetahui kemampuan sistem dalam mendeteksi perubahan isi dokumen dan ketidaksesuaian dengan *QR Code*. Pada pengujian pertama, dokumen berukuran 100–500 KB yang telah diubah tetapi menggunakan *QR Code* asli menghasilkan status **invalid** karena nilai hash dokumen tidak lagi sesuai dengan tanda tangan digital. Pada pengujian kedua, dokumen yang diverifikasi berbeda dengan dokumen yang terhubung dengan QR Code, sehingga hasil verifikasi juga menunjukkan status **invalid**.

Skenario 3 : Signature/Hash Tidak Sesuai

Berdasarkan hasil pengujian, dokumen yang mengalami perubahan pada isi maupun pada kode QR menghasilkan status verifikasi **invalid**. Perubahan tersebut menyebabkan nilai hash**,** public key, atau informasi pada kode QR tidak lagi sesuai dengan data asli saat proses penandatanganan. Hal ini menunjukkan bahwa sistem mampu mendeteksi ketidaksesuaian data sehingga keaslian dan integritas dokumen tidak dapat diverifikasi. Dapat dilihat pada tabel 4.7 dibawah ini.

Tabel 4.7. Pengujian Signature/Hash Yang Tidak Sesuai

No	QR Code	Nama File	Status Dokumen	Status QR Code	Status Verifikasi	Ket
1		SD 1 (100-500 kb).pdf	Diubah	Asli	Invalid	Signature berubah
2		SD 1 (500KB-2MB).pdf	Diubah	Diubah	Invalid	*Public key* dan kode QR berubah

Pada table 4.7 tersebut menunjukkan hasil pengujian verifikasi ketika dokumen dan *QR Code* mengalami perubahan. Pada pengujian pertama, dokumen yang telah diubah namun masih menggunakan *QR Code* asli menghasilkan status verifikasi **invalid** karena nilai *signature* berubah. Pada pengujian kedua, baik dokumen maupun *QR Code* telah diubah sehingga hasil verifikasi juga menunjukkan status **invalid** akibat perubahan pada *public key* dan kode QR. Hasil ini menunjukkan bahwa sistem mampu mendeteksi perubahan pada dokumen maupun *QR Code*.

4.4. Analisis dan Pembahasan

Hasil pengujian sistem pada penelitian ini menunjukkan kesesuaian dengan konsep yang telah dijelaskan pada landasan teori mengenai kriptografi dan keamanan dokumen digital. Proses hashing yang menggunakan algoritma SHA-256 menghasilkan nilai hash dengan panjang tetap sebesar 256 bit pada setiap dokumen yang diuji. Hal ini sesuai dengan teori fungsi hash kriptografis yang menyatakan bahwa panjang output hash bersifat tetap (*fixed length*) meskipun ukuran data masukan berbeda.

Selain itu, hasil pengujian menunjukkan bahwa waktu proses hashing meningkat seiring dengan bertambahnya ukuran dokumen. Hal ini sejalan dengan teori bahwa proses hashing harus membaca seluruh isi data untuk menghasilkan nilai hash, sehingga semakin besar ukuran dokumen maka semakin banyak data yang harus diproses oleh sistem. Pada tahap penandatanganan digital, sistem menggunakan algoritma EdDSA untuk menghasilkan digital signature dari nilai hash dokumen. Berdasarkan hasil pengujian, seluruh dokumen berhasil menghasilkan signature dengan ukuran yang konsisten. Hal ini sesuai dengan konsep *digital signature* yang menyatakan bahwa tanda tangan digital digunakan untuk menjamin autentikasi dan integritas dokumen.

Selanjutnya pada proses verifikasi dokumen, sistem memanfaatkan *QR Code* sebagai media penyimpanan informasi verifikasi. *QR Code* berfungsi untuk menyimpan data yang berkaitan dengan proses validasi dokumen sehingga pengguna

dapat melakukan verifikasi dengan cepat melalui proses pemindaian. Hasil pengujian menunjukkan bahwa dokumen yang tidak mengalami perubahan menghasilkan status valid, sedangkan dokumen yang telah dimodifikasi menghasilkan status invalid. Kondisi ini sesuai dengan teori integritas data pada sistem tanda tangan digital yang menyatakan bahwa perubahan sekecil apa pun pada dokumen akan menghasilkan nilai hash yang berbeda sehingga proses verifikasi gagal.

Dengan demikian, hasil pengujian yang diperoleh pada penelitian ini mendukung konsep teoritis yang telah dijelaskan pada landasan teori, khususnya terkait dengan fungsi hash, digital signature, dan mekanisme verifikasi dokumen digital. Namun, penelitian ini tidak dimaksudkan sebagai solusi akhir untuk implementasi berkendara nyata, melainkan sebagai tahap pengembangan dasar yang siap dikembangkan lebih lanjut melalui pengujian lapangan dan integrasi indikator tambahan.

BAB V
PENUTUP

5.1 Kesimpulan

Berdasarkan hasil penelitian sistem tanda tangan digital menggunakan algoritma *Edwards-curve Digital Signature Algorithm* (EdDSA) dan SHA-256 dengan kode QR pada dokumen sertifkat PDF, maka dapat ditarik kesimpulan sebagai berikut:

1. Efektivitas Integrasi Sistem**:** Sistem berhasil mengintegrasikan algoritma EdDSA dan SHA-256 untuk menjamin keamanan sertifikat digital. Penggunaan kode QR sebagai media penyimpanan tanda tangan digital terbukti efektif dalam memfasilitasi proses verifikasi dokumen secara praktis tanpa mengubah struktur fisik dokumen asli secara signifikan.
2. Kinerja dan Validitas sistem Verifikasi**:** Pengujian menunjukkan bahwa algoritma EdDSA memiliki performa yang sangat efisien dalam proses otentikasi. Rata-rata waktu penandatanganan (*signing time*) untuk dokumen kecil (100–500 KB) adalah 0,003 detik, dokumen sedang (500 KB–2 MB) adalah 0,051 detik, dan dokumen besar (2–5 MB) adalah 0,087 detik. Meskipun waktu proses meningkat secara linier seiring bertambahnya ukuran dokumen, seluruh proses tetap berada pada rentang milidetik yang sangat responsif. Selain itu, sistem mampu mendeteksi keaslian dokumen dengan baik, di mana dokumen asli menghasilkan status *valid*, sedangkan dokumen yang dimodifikasi atau tidak sesuai dengan kode QR menghasilkan status *invalid*, sehingga integritas dan keaslian dokumen dapat terjaga.

5.2 Saran

Untuk pengembangan dan penyempurnaan sistem lebih lanjut, disarankan beberapa hal berikut:

1. Pengembangan Platform. Disarankan untuk mengembangkan sistem ini ke dalam platform *mobile* (Android/iOS) guna meningkatkan aksesibilitas verifikasi dokumen di lapangan secara *real-time*.
2. Pengujian Ketahanan Kriptografi. Perlu dilakukan pengujian lebih lanjut mengenai ketahanan algoritma EdDSA terhadap serangan kriptografi spesifik, seperti serangan *side-channel* yang lebih kompleks atau simulasi serangan komputer kuantum untuk memastikan keamanan jangka panjang.
3. Optimalisasi Kecepatan. Untuk dokumen berukuran sangat besar (di atas 10 MB), disarankan melakukan optimalisasi pada metode ekstraksi teks agar waktu *hashing* tetap minimal.

DAFTAR PUSTAKA

Adobe. (2021). *Securing PDFs with certificates*. Adobe Systems. https://helpx.adobe.com/id_id/acrobat/using/securing-pdfs-certificates.html

Alfani, M. R., Furqan, M., & Nasution, Y. R. (2024). Pengamanan Data Teks Menggunakan Metode Digital Signature Algorithm (Dsa) Dan Advanced Encryption Standard (Aes). *Journal of Science and Social Research*, *4307*(1), 301–306. http://jurnal.goretanpena.com/index.php/JSSR

Amalya, N., Silalahi, S. M. S., Fatricia Nasution, D., Sari, M., & Gunawan, I. (2023). Kriptografi dan Penerapannya Dalam Sistem Keamanan Data. *Jurnal Media Informatika*, *4*(2), 90–93. https://doi.org/10.55338/jumin.v4i2.428

Arwah, N., Aminuddin, A., & Arifianto, S. (2024). Implementasi Tanda Tangan Digital menggunakan ECDSA (Studi Kasus: Jurnal Tipe File pdf). *Jurnal Repositor*, *3*(3), 321–330. https://doi.org/10.22219/repositor.v3i3.31069

Azhari, M., Mulyana, D. I., Perwitosari, F. J., & Ali, F. (2022). Implementasi Pengamanan Data pada Dokumen Menggunakan Algoritma Kriptografi Advanced Encryption Standard (AES). *Jurnal Pendidikan Sains Dan Komputer*, *2*(01), 163–171. https://doi.org/10.47709/jpsk.v2i01.1390

Cinal, A., & Sobolewski, O. (2025). On cofactored verification of EdDSA signatures. *International Journal of Electronics and Telecommunications*, *71*(2), 453–461. https://doi.org/10.24425/ijet.2025.153592

Fachrul, M., Sutardi, Tajidun, L. ., & Bahtiar Aksara, L. . (2022). *PENERAPAN KONSEP DIGITAL SIGNATURE TERHADAP VERIFIKASI KEASLIAN DOKUMEN TRANSKRIP NILAI MAHASISWA MENGGUNAKAN ENKRIPSI RIVEST SHAMIR ADLEMAN*.

FIPS 180-3. (2008). *Archived Publication*. *180–3*(October 2008). http://csrc.nist.gov/publications/PubsFIPS.html#fips180-4.

Franck, L. D., Ginja, G. A., Carmo, J. P., Afonso, J. A., & Luppe, M. (2024). Custom

ASIC Design for SHA-256 Using Open-Source Tools. *Computers*, *13*(1), 1–16. https://doi.org/10.3390/computers13010009

Hutagalung, J., Ramadhan, P. S., & Sihombing, S. J. (2023). Keamanan Data Menggunakan Secure Hashing Algorithm (SHA)-256 dan Rivest Shamir Adleman (RSA) pada Digital Signature. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, *10*(6), 1213–1222. https://doi.org/10.25126/jtiik.1067319

Hanyfa Sari, N. (2024). Penerapan Teknik Digital Signature Dalam Pengamanan Piagam Penghargaan Menggunakan Algoritma SHA-1 dan RSA. *Management of Information System Journal*, *2*(3), 55–67. https://doi.org/10.47065/mis.v2i3.1465

Lorien, A., & Wellem, T. (2021). Implementasi Sistem Otentikasi Dokumen Berbasis Quick Response (QR) Code dan Digital Signature. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, *5*(4), 663–671. https://doi.org/10.29207/resti.v5i4.3316

Meko, D. A. (2023). Jurnal Teknologi Terpadu Perbandingan Algoritma DES , AES , IDEA Dan Blowfish dalam Enkripsi dan Dekripsi Data Donzilio Antonio Meko Program Studi Teknik Informatika , STIMIK Kupang Jurnal Teknologi Terpadu. *Jurnal Teknologi Terpadu*, *4*(1), 8–15.

Nadimsyah, A. Z., Ezar, M., & Rivan, A. (2024). *Implementasi Secure Hash Algorithm-256 dan Advanced Encryption Standard Untuk Verifikasi Tanda Tangan Digital*. *6*(1), 240–250. https://doi.org/10.47065/josh.v6i1.5942

Nuraeni, F., Handoko Agustin, Y., Kurniadi, D., & Dewi Ariyanti, I. (2020). Implementasi Skema QR-Code dan Digital Signature menggunakan Kombinasi Algoritma RSA dan AES untuk Pengamanan Data Sertifikat Elektronik. *Seminar Nasional Teknologi Informasi Komunikasi Dan Industri*, *12*(6), 43–52.

Nurjaman, A. rizal. (2024). Penanda Tanganan Dokumen Digital Pada Sistem Penyimpanan File Menggunakan Kombinasi Algoritma Sha3-512 Dan Rsa Untuk

Mempertahankan Keaslian Data Dokumen. *Jurnal Ilmiah Teknologi Infomasi Terapan*, *10*(3), 119–129. https://doi.org/10.33197/jitter.vol10.iss3.2024.2200

Prashanth, B., Naveen, C., Sreeja, B., & Naveena, G. (2025). Secure Files Using Secure Hash Algorithm and Elliptical Curve Cryptography. *International Journal on Science and Technology*, *16*(2), 2–6. https://doi.org/10.71097/ijsat.v16.i2.5555

Raschka, S., Patterson, J., & Nolet, C. (2020). Machine learning in python: Main developments and technology trends in data science, machine learning, and artificial intelligence. *Information (Switzerland)*, *11*(4). https://doi.org/10.3390/info11040193

Saepulrohman, A., & Ismangil, A. (2021). Data integrity and security of digital signatures on electronic systems using the digital signature algorithm (DSA). *International Journal of Electronics and Communications Systems*, *1*(1), 11–15. https://doi.org/10.24042/ijecs.v1i1.7923

Saputro, T. H., Hidayati, N. H., & H. Ujianto, E. I. (2020). Survei Tentang Algoritma Kriptografi Asimetris. *Jurnal Informatika Polinema*, *6*(2), 67–72. https://doi.org/10.33795/jip.v6i2.345

Suantara, I. P. Y., Satwika, I. P., & Fredlina, K. Q. (2024). Perbandingan Kinerja Waktu Algoritma ECDSA, EdDSA, RSA, Dan Implementasinya Pada Sistem Multi-Signature Dokumen Pdf. *Jurnal Teknik Informatika Dan Sistem Informasi*, *11*(1), 85–98. http://jurnal.mdp.ac.id

Sugiyono. (2020). *Metodologi Penelitian Kuantitatif, Kualitatif dan R & D*.

Syahrir, M., & Fatimatuzzahra, F. (2020). Association Rule Integrasi Pendekatan Metode Custom Hashing dan Data Partitioning untuk Mempercepat Proses Pencarian Frekuensi Item-set pada Algoritma Apriori. *MATRIK : Jurnal Manajemen, Teknik Informatika Dan Rekayasa Komputer*, *20*(1), 149–158. https://doi.org/10.30812/matrik.v20i1.833

Wellem, T., Nataliani, Y., & Iriani, A. (2022). Academic Document Authentication

using Elliptic Curve Digital Signature Algorithm and QR Code. *International Journal on Informatics Visualization*, *6*(3), 667–675. https://doi.org/10.30630/joiv.6.2.872

Winanda, Y. A., Fatimah, T., Aditya, A., & Ushud, A. (2025). Meningkatkan Keamanan Invoice Dengan Enkripsi Qr- Code Dan Digital Signature Berbasis Rsa Dan Sha-256 Improving Invoice Security With Qr-Code Encryption And Digital Signature Based On Rsa Dan Sha-256. *22*(1), 62–69.