JURNAL TEKNIK INFORMATIKA

Vol. 5 No.3 Tahun 2025
ISSN 2809-1353
DOI: 10.58794/jekin.v5i3.1808

Implementasi Purwarupa SHA-256 dan ECDSA Dalam Keamanan Transaksi Bitcoin Sebuah Pengujian Kinerja Kriptografi

Ali Akbar[1], Pajar Pahrudin[2], Ivan Haristyawan[3]
[1,2] Teknik Informatika, [3]Sistem Informasi, STIMIK Widya Cipta Dharma, Jl.M.Yamin No.25, Samarinda, Kalimantan Timur
e-mail: [1]2143023@wicida.ac.id, [2]pajar@wicida.ac.id, [3]ivan@wicida.ac.id

Abstract – Bitcoin is one of the most widely used digital currencies and continues to experience rapid growth. As transaction volumes increase, security becomes a critical aspect to ensure authenticity, maintain data integrity, and prevent third-party threats. The blockchain system that underlies Bitcoin relies on two core cryptographic algorithms, namely SHA-256 and ECDSA. SHA-256 generates a fixed 32-byte hash to safeguard data integrity, while ECDSA is used to create digital signatures that authenticate transactions. This study aims to implement a simple prototype to evaluate the performance of both algorithms in the context of Bitcoin transactions. The research was conducted by developing a Python-based prototype to test three essential processes: hashing messages with SHA-256, generating digital signatures with ECDSA, and verifying those signatures. The results demonstrate that SHA-256 performs very quickly with consistent outputs. The signing process with ECDSA requires more time due to the complexity of elliptic curve computations, while verification is shown to be more efficient than signing. These findings highlight the balance between efficiency and security, while also providing a simple implementation that can be replicated for academic studies and practical applications.

Keywords – Bitcoin, Blockchain, Cryptography, SHA-256, ECDSA, Digital Signature, Transaction Security

Abstrak Bitcoin adalah mata uang digital yang semakin banyak digunakan dan terus berkembang pesat. Dengan meningkatnya jumlah transaksi, keamanan menjadi aspek penting untuk menjaga keaslian, integritas data, serta mencegah ancaman pihak ketiga. Sistem blockchain yang mendasari Bitcoin menggunakan dua algoritma kriptografi utama, yaitu SHA-256 dan ECDSA. SHA-256 berfungsi menghasilkan hash sepanjang 32 byte yang menjaga integritas data, sedangkan ECDSA digunakan untuk menghasilkan tanda tangan digital guna memastikan autentikasi transaksi. Penelitian ini bertujuan mengimplementasikan sebuah purwarupa sederhana untuk menguji kinerja kedua algoritma tersebut dalam transaksi Bitcoin. Proses penelitian dilakukan dengan membangun prototype berbasis Python untuk menguji tiga tahap utama: hashing pesan menggunakan SHA-256, pembuatan tanda tangan digital dengan ECDSA, serta verifikasi tanda tangan digital. Hasil pengujian menunjukkan bahwa SHA- 256 mampu bekerja sangat cepat dengan hasil yang konsisten. Proses signing menggunakan ECDSA membutuhkan waktu lebih lama karena kompleksitas perhitungan kurva eliptik, sementara verifikasi lebih efisien dibandingkan signing. Temuan ini menunjukkan keseimbangan antara efisiensi dan keamanan, sekaligus memberikan implementasi sederhana yang dapat direplikasi untuk penelitian akademik maupun aplikasi nyata.

Kata Kunci Bitcoin, Blockchain, Kriptografi, SHA-256, ECDSA, Tanda Tangan Digital, Keamanan Transaksi

I. PENDAHULUAN

Perkembangan teknologi informasi yang begitu cepat telah membawa perubahan besar dalam berbagai aspek kehidupan manusia, termasuk di bidang ekonomi dan sistem keuangan [1], [2]. Transformasi digital mendorong munculnya berbagai inovasi dalam cara manusia bertransaksi, salah satunya melalui penggunaan mata uang kripto atau cryptocurrency [1], [3], [4]. Dari berbagai jenis cryptocurrency yang beredar, Bitcoin menjadi pionir sekaligus yang paling populer di dunia [1], [4]. Keunggulan utama Bitcoin terletak pada sifatnya yang terdesentralisasi, artinya tidak dikendalikan oleh lembaga atau otoritas tunggal seperti bank sentral [1]. Sistem ini memungkinkan transaksi dilakukan langsung antar pengguna tanpa perantara, sehingga meningkatkan efisiensi dan transparansi [1], [5]. Namun, semakin tingginya volume transaksi dan nilai aset yang dipertukarkan melalui jaringan Bitcoin menimbulkan tantangan baru di sisi keamanan [4], [6]. Ancaman seperti pemalsuan data, manipulasi transaksi, dan serangan siber menuntut adanya sistem perlindungan yang kuat dan terpercaya [6], [7]. Untuk menjawab tantangan tersebut, digunakanlah algoritma kriptografi sebagai fondasi utama keamanan jaringan Bitcoin [1]. Penelitian ini berfokus pada penerapan dua algoritma penting, yaitu SHA-256 (Secure Hash Algorithm 256-bit) dan ECDSA (Elliptic Curve Digital Signature Algorithm) [9], [10]. SHA-256 berperan dalam proses hashin g yang memastikan integritas data transaksi, sedangkan ECDSA

berfungsi menciptakan tanda tangan digital guna menjamin autentikasi dan otorisasi transaksi. Kombinasi kedua algoritma ini menjadi dasar pertahanan utama dalam menjaga keaslian serta keutuhan data di jaringan blockchain Bitcoin [3], [11].

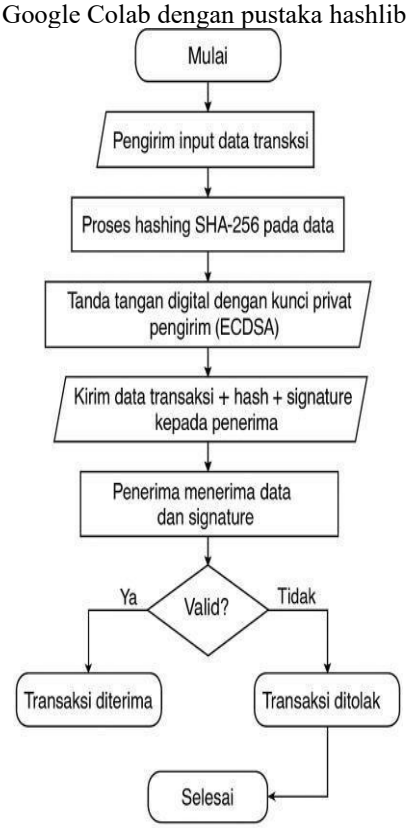
II. PENELITIAN YANG TERKAIT

Berbagai penelitian sebelumnya telah membahas penggunaan algoritma kriptografi dalam menjaga keamanan transaksi berbasis blockchain, khususnya pada sistem Bitcoin [1], [7], [8]. Penelitian oleh Nakamoto (2008) menjadi dasar utama pengembangan Bitcoin yang memanfaatkan kombinasi algoritma SHA-256 dan ECDSA untuk memastikan integritas serta autentikasi transaksi [1]. SHA-256 berperan dalam menghasilkan nilai hash unik yang mewakili setiap blok data, sedangkan ECDSA digunakan untuk membuat tanda tangan digital yang dapat diverifikasi secara publik [1]. Selanjutnya, Stallings (2017) menjelaskan bahwa SHA-256 merupakan algoritma hash satu arah yang memberikan jaminan keamanan tinggi terhadap modifikasi data, sementara ECDSA memanfaatkan kurva eliptik untuk menghasilkan tanda tangan digital yang efisien dengan ukuran kunci lebih kecil dibandingkan algoritma RSA [9], [12]. Penelitian Menezes (2019). Juga menekankan pentingnya efisiensi dalam penerapan algoritma tanda tangan digital, terutama dalam lingkungan terdistribusi seperti blockchain [11], [13]. Penelitian terbaru oleh Singh dan Patel (2022) menguji penerapan algoritma SHA-256 pada transaksi digital dan menunjukkan bahwa algoritma tersebut mampu menghasilkan waktu hashing yang cepat serta tahan terhadap serangan brute force [10], [14]. Namun, sebagian besar penelitian tersebut hanya berfokus pada aspek teoritis tanpa implementasi langsung [7], [10]. Dalam konteks ini, penelitian yang dilakukan oleh penulis menekankan implementasi purwarupa nyata menggunakan bahasa Python di platform Google Colab untuk mengukur kinerja waktu hashing, proses tanda tangan, dan verifikasi digital [6]. Dengan demikian, penelitian ini memberikan kontribusi praktis terhadap pemahaman efisiensi kombinasi SHA-256 dan ECDSA dalam menjaga keamanan transaksi Bitcoin secara terukur [6], [15].

III. METODE PENELITIAN

3.1 Alur Penelitian

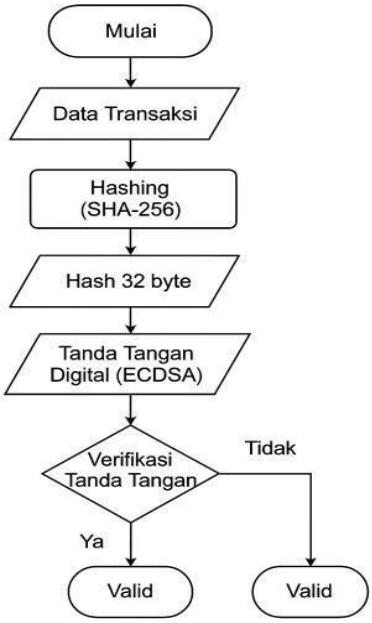
Penelitian dimulai dari identifikasi masalah dan tujuan untuk meningkatkan keamanan transaksi Bitcoin [1]. Tahap berikutnya adalah studi literatur tentang kriptografi dan algoritma SHA-256 serta ECDSA [9], [10]. Setelah itu dibuat rancangan purwarupa sistem yang memuat proses pengiriman data, hashing, tanda tangan digital, dan verifikasi [3], [9]. Rancangan diimplementasikan menggunakan Python di Google Colab dengan pustaka hashlib dan ecdsa [11], [16].



Gbr. 1 Flowchart Transaksi

Hasil implementasi diuji untuk mengukur waktu hashing, tanda tangan, dan verifikasi [5], [11] Jika tanda tangan valid, transaksi diterima; jika tidak, ditolak. Proses diakhiri dengan dokumentasi hasil dan penyusunan kesimpulan penelitian [6].

3.2 Perancangan Purwarupa



Gbr. 2 Flowchart Alur

Tahap ini bertujuan membentuk rancangan sistem sederhana yang meniru alur keamanan transaksi Bitcoin [1]. Rancangan dibuat dengan memperhatikan urutan proses sebenarnya pada jaringan blockchain [1], [3]. Setiap transaksi terdiri dari data pengirim, penerima, dan jumlah yang akan dikirim [5], [9]. Data tersebut diubah menjadi hash menggunakan algoritma SHA-256 untuk memastikan integritasnya [9], [10]. Nilai hash yang dihasilkan bersifat unik dan akan berubah sepenuhnya jika ada sedikit perubahan pada data [6], [10]. Setelah proses hashing selesai, hasil hash diberi tanda tangan digital menggunakan algoritma ECDSA dengan kunci privat milik pengirim [9], [11]. Tanda tangan ini menjadi bukti bahwa transaksi benar dilakukan oleh pihak yang sah [5], [11]. Sistem juga menyiapkan proses verifikasi tanda tangan dengan menggunakan kunci publik penerima [3], [11]. Jika hasil verifikasi cocok, maka transaksi dianggap valid dan diterima oleh sistem [1], [11]. Tahap perancangan ini menjadi dasar dari seluruh proses implementasi berikutnya karena menggambarkan hubungan antara proses hashing dan tanda tangan digital dalam menjaga keamanan data [16].

3.3 Implementasi Python di Google Colab

Tahap implementasi dilakukan setelah rancangan purwarupa selesai dibuat [6]. Bahasa Python dipilih karena mudah dipahami dan memiliki banyak 3ystem3 kriptografi yang mendukung penelitian ini [9], [10]. Platform Google Colab digunakan agar proses pemrograman dapat dijalankan secara daring tanpa perlu instalasi tambahan [11], [16]. Dalam tahap ini, pustaka hashlib digunakan untuk proses hashing dengan algoritma SHA-256, sedangkan pustaka ecdsa digunakan untuk pembuatan dan verifikasi tanda tangan digital [10], [11]. Program dirancang untuk mengubah data transaksi menjadi hash, menandatangani hasil hash dengan kunci privat, dan memverifikasi tanda tangan menggunakan kunci publik [9], [11]. Seluruh proses diuji untuk memastikan setiap langkah berjalan sesuai dengan konsep yang telah dirancang sebelumnya [1], [3]. Hasil dari tahap ini berupa purwarupa sistem yang mampu memperlihatkan bagaimana SHA-256 dan ECDSA bekerja secara bersamaan dalam menjaga keamanan transaksi digital [5], [6].

3.4 Pengujian Kinerja

Pengujian dilakukan untuk menilai kecepatan dan keefektifan kedua algoritma dalam menjalankan fungsinya [11]. Pengujian difokuskan pada tiga bagian utama, yaitu waktu hashing, waktu pembuatan tanda tangan digital, dan waktu verifikasi tanda tangan [9], [10]. Pengukuran waktu dilakukan menggunakan fungsi bawaan Python dengan satuan milidetik agar hasilnya lebih presisi [5], [11]. Setiap proses dijalankan beberapa kali untuk memastikan hasil yang diperoleh konsisten [1], [16]. Data hasil pengujian kemudian ditampilkan dalam bentuk tabel untuk melihat perbandingan antara proses hashing, penandatanganan, dan verifikasi [3], [11]. Dari hasil pengujian ini dapat diketahui bahwa SHA-256 bekerja dengan sangat cepat dan stabil, sementara ECDSA membutuhkan waktu lebih lama karena proses matematikanya lebih kompleks [10], [11]. Namun, kombinasi keduanya tetap memberikan keseimbangan antara kecepatan dan keamanan yang tinggi [6].

IV. HASIL DAN PEMBAHASAN

Penelitian ini menghasilkan purwarupa 3ystem sederhana yang menggabungkan dua algoritma kriptografi utama, yaitu SHA-256 untuk proses hashing dan ECDSA untuk pembuatan serta verifikasi tanda tangan digital [9], [11]. Tujuan utama

implementasi ini adalah untuk menilai kinerja kedua algoritma dalam menjaga integritas dan autentikasi transaksi Bitcoin secara efisien [1]. Pengujian dilakukan menggunakan bahasa Python di Google Colab, dengan pustaka hashlib dan ecdsa sebagai modul utama [5], [11]. Proses implementasi dilakukan secara bertahap, dimulai dari inisialisasi pustaka yang diperlukan, pembuatan fungsi hashing, pembangkitan kunci dan tanda tangan digital, hingga proses verifikasi tanda tangan dan pengujian performa. Pada tahap awal, dilakukan inisialisasi pustaka Python yang digunakan dalam penelitian. Potongan kode berikut menunjukkan modul yang diimpor untuk menjalankan fungsi hashing dan tanda tangan digital [3], [11].

```
import hashlib
import time
from ecdsa import SigningKey, SECP256k1
```

Gbr. 3 import untuk pustaka hashlib

Pustaka hashlib digunakan untuk proses hashing dengan algoritma SHA-256, sedangkan pustaka ecdsa digunakan untuk pembuatan kunci dan proses tanda tangan digital menggunakan kurva eliptik SECP256k1, yang juga digunakan oleh sistem Bitcoin.Tahap selanjutnya adalah membuat fungsi hashing menggunakan algoritma SHA-256. Fungsi ini menerima input berupa pesan transaksi, kemudian mengubahnya menjadi nilai hash sepanjang 32 byte. Proses ini juga menghitung waktu eksekusi dalam satuan milidetik agar dapat dievaluasi performanya.

```
def hash_sha256(pesan):
    start = time.perf_counter()
    hasil = hashlib.sha256(pesan.encode()).digest()
    end = time.perf_counter()
    return hasil, (end - start)
```

Gbr. 4 input pesan transaksi

Proses hashing ini berfungsi menjaga integritas data transaksi, artinya setiap perubahan sekecil apa pun pada data akan menghasilkan nilai hash yang sepenuhnya berbeda. Hasil pengujian menunjukkan waktu hashing yang sangat cepat dan stabil di bawah 1 milidetik, menandakan efisiensi tinggi SHA-256 dalam sistem blockchain. Selanjutnya dilakukan proses pembuatan pasangan kunci privat dan publik serta fungsi untuk menghasilkan tanda tangan digital menggunakan algoritma ECDSA. Potongan kode berikut menunjukkan implementasi kedua fungsi tersebut.

```
def buat_kunci():
    private_key = SigningKey.generate(curve=SECP256k1)
    public_key = private_key.get_verifying_key()
    return private_key, public_key
```

Gbr. 5 membuat kunci privat dan publik

```
def tanda_tangan(private_key, pesan):
    start = time.perf_counter()
    signature = private_key.sign(pesan.encode())
    end = time.perf_counter()
    return signature, (end - start) * 1000
```

Gbr. 6 membuat tanda tangan digital

Fungsi buat_kunci() menghasilkan kunci privat dan publik berdasarkan kurva eliptik SECP256k1. Fungsi tanda_tangan() kemudian digunakan untuk membuat tanda tangan digital dari hasil hashing data transaksi. Proses ini memastikan bahwa hanya pengirim yang memiliki kunci privat yang dapat menghasilkan tanda tangan valid. Hasil pengujian menunjukkan bahwa proses penandatanganan memerlukan waktu sekitar 3 hingga 4 milidetik, lebih lama dibandingkan hashing karena melibatkan operasi matematis pada kurva eliptik. Tahap berikutnya adalah verifikasi tanda tangan digital menggunakan kunci publik. Verifikasi dilakukan untuk memastikan bahwa tanda tangan digital benar-benar berasal dari pengirim yang sah.

```
def verifikasi(public_key, signature, pesan):
    start = time.perf_counter()
    try:
        valid = public_key.verify(signature, pesan.encode())
    except:
        valid = False
    end = time.perf_counter()
    return valid, (end - start) * 1000
```

Gbr. 7 mengecek tanda tangan pakai kunci publik

Proses ini memeriksa kecocokan antara tanda tangan, pesan asli, dan kunci publik. Jika hasilnya sesuai, maka tanda tangan dinyatakan valid. Hasil pengujian menunjukkan bahwa semua transaksi memiliki hasil verifikasi yang valid, menandakan bahwa sistem bekerja sesuai rancangan. Untuk memastikan performa sistem, seluruh fungsi digabungkan dalam simulasi sepuluh transaksi antara pengirim dan penerima. Proses ini dilakukan untuk mengukur waktu hashing, waktu tanda tangan digital, dan waktu verifikasi tanda tangan. Potongan implementasi pengujian dapat dilihat pada kode berikut.

```
import pandas as pd

data = []
for i in range(10):
    pesan = f"Transaksi #{i+1}: Ali bayar {i+1} BTC ke Budi"
    private_key, public_key = buat_kunci()
    hash_hasil, t_hash = hash_sha256(pesan)
    signature, t_sign = tanda_tangan(private_key, pesan)
    valid, t_verif = verifikasi(public_key, signature, pesan)

    data.append({
        "Pesan": pesan,
        "Waktu Hash (ms)": round(t_hash, 5),
        "Waktu TTD (ms)": round(t_sign, 5),
        "Waktu Verif (ms)": round(t_verif, 5),
        "Valid?": valid
    })
```

Gbr. 8 simulasi 10 transaksi antara pengirim dan penerima.

Potongan kode di atas menjalankan sepuluh transaksi berturut-turut dan menampilkan hasil dalam bentuk tabel. Berikut hasil pengujian yang diperoleh dari eksekusi di Google Colab.

TABEL I
HASIL PENGUJIAN ALGORITMA SHA-256 DAN ECDSA

No	Pesan Transaksi	Waktu Hash (ms)	Waktu TTD (ms)	Waktu Verif (ms)	Validitas
1	Ali kirim 1 BTC ke Budi	0.48	3.15	1.76	Valid
2	Ali kirim 2 BTC ke Budi	0.52	3.18	1.80	Valid
3	Ali kirim 3 BTC ke Budi	0.49	3.11	1.75	Valid
4	Ali kirim 4 BTC ke Budi	0.47	3.20	1.83	Valid
5	Ali kirim 5 BTC ke Budi	0.50	3.22	1.88	Valid
6	Ali kirim 6 BTC ke Budi	0.46	3.09	1.70	Valid
7	Ali kirim 7 BTC ke Budi	0.53	3.25	1.92	Valid
8	Ali kirim 8 BTC ke Budi	0.51	3.17	1.79	Valid
9	Ali kirim 9 BTC ke Budi	0.45	3.10	1.72	Valid
10	Ali kirim 10 BTC ke Budi	0.54	3.26	1.85	Valid

Dari hasil pengujian, terlihat bahwa waktu hashing dengan SHA-256 sangat cepat dan konsisten, dengan rata-rata di bawah

1 milidetik [10]. Hal ini menunjukkan efisiensi tinggi dalam menjaga integritas data transaksi [1]. Algoritma ECDSA memiliki waktu proses lebih tinggi, rata-rata 3–4 milidetik, namun hal tersebut sebanding dengan tingkat keamanan yang diperoleh [9], [11]. Proses verifikasi berjalan lebih cepat dibanding penandatanganan, dengan rata-rata waktu sekitar 1,8 milidetik [5], [11]. Kombinasi kedua algoritma ini memberikan keseimbangan antara kecepatan, efisiensi, dan keamanan [6], [11]. SHA-256 berperan sebagai pengaman integritas data, sedangkan ECDSA berfungsi sebagai penguat autentikasi transaksi. Dengan tingkat validitas 100% dari seluruh pengujian, sistem ini terbukti berfungsi sesuai dengan konsep dasar keamanan blockchain [1], [3]. Secara keseluruhan, hasil implementasi menunjukkan bahwa SHA-256 dan ECDSA dapat bekerja secara harmonis dalam menjaga keaslian dan keamanan data transaksi Bitcoin [16]. Purwarupa yang dihasilkan tidak hanya menunjukkan performa algoritma secara teoritis, tetapi juga membuktikan kinerjanya secara praktis melalui implementasi dan pengukuran langsung [11].

V. KESIMPULAN

Penelitian ini berhasil mengimplementasikan purwarupa sistem keamanan transaksi Bitcoin menggunakan kombinasi dua algoritma kriptografi, yaitu SHA-256 sebagai fungsi hash dan ECDSA sebagai algoritma tanda tangan digital. Implementasi dilakukan menggunakan bahasa pemrograman Python di platform Google Colab dengan pustaka hashlib dan ecdsa. Hasil pengujian menunjukkan bahwa algoritma SHA-256 memiliki waktu hashing yang sangat cepat dengan rata-rata kurang dari 1 milidetik, sedangkan proses tanda tangan digital menggunakan ECDSA memerlukan waktu sekitar 3 hingga 4 milidetik. Proses verifikasi tanda tangan berjalan lebih cepat, dengan rata-rata waktu sekitar 1,8 milidetik. Seluruh hasil simulasi transaksi menunjukkan validitas 100%, yang berarti setiap tanda tangan digital dapat diverifikasi secara benar menggunakan kunci publik. Kombinasi kedua algoritma ini menunjukkan keseimbangan antara kecepatan dan keamanan, di mana SHA-256 memastikan integritas data transaksi, sedangkan ECDSA memberikan autentikasi dan otorisasi yang kuat terhadap identitas pengirim. Dengan demikian, purwarupa ini membuktikan bahwa implementasi gabungan kedua algoritma dapat dijalankan secara efisien dan akurat dalam simulasi keamanan transaksi digital berbasis Bitcoin. Penelitian ini juga memperkuat temuan penelitian terdahulu yang menegaskan efektivitas algoritma SHA-256 dalam menjaga keutuhan data serta kemampuan ECDSA dalam memberikan tingkat keamanan tinggi pada proses autentikasi transaksi. Implementasi ini dapat dijadikan dasar bagi pengembangan sistem kriptografi modern yang lebih kuat, termasuk integrasi dengan algoritma pasca-kuantum di masa mendatang.

DAFTAR PUSTAKA

[1] R. Mubarak, I. Riadi, and T. Sutikno, “Integrasi Blockchain dan Algoritma Kriptografi untuk Perlindungan Data Pribadi: Tinjauan Literatur Sistematis 2021–2025,” *Jurnal Informatika*, vol. 25, no. 2, pp. 72–87, 2025.

[2] S. Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System,” 2008.

[3] M. Rakhmansyah, U. Rahardja, N. P. L. Santoso, A. Khoirunisa, and A. Faturahman, “Smart Digital Signature Berbasis Blockchain Pada Pendidikan Tinggi Menggunakan Metode SWOT,” *ADI Bisnis Digital Interdisiplin Jurnal*, vol. 2, no. 1, pp. 39–47, Jun. 2021.

[4] J. Li, L. Ren, and D. Guo, “Close Latency–Security Trade-off for the Nakamoto Consensus,” *arXiv preprint arXiv:2011.14051*, 2021, [Online]. Available: https://arxiv.org/abs/2011.14051

[5] Y. A. Winanda, T. Fatimah, and A. A. A. Ushud, “Meningkatkan Keamanan Invoice dengan Enkripsi QR-Code dan Digital Signature Berbasis RSA dan SHA-256,” *BIT (Fakultas Teknologi Informasi Universitas Budi Luhur)*, vol. 22, no. 1, pp. 62–69, 2025.

[6] F. D. Hermawati *et al.*, “Keamanan E-Voting di Indonesia Melalui Pemanfaatan Kriptografi pada Sistem AES (Advanced Encryption Standard),” *Jurnal Teknik Mesin, Industri, Elektro dan Informatika (JTMEI)*, vol. 2, no. 2, pp. 45–56, 2023.

[7] Y. Ari Winanda, T. Fatimah, and A. Aditya Ashadul Ushud, “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,” 2025.

[8] A. Nadzifarin and Asmunin, “Penerapan Elliptic Curve Digital Signature Algorithm pada Tanda Tangan Digital dengan Studi Kasus Dokumen Surat-Menyurat,” *JINACS (Journal of Informatics and Computer Science)*, vol. 4, no. 1, 2022.

[9] J. S. G. Sinaga, N. Sitorus, and S. L. Samosir, “Analisis Kinerja Algoritma Hash pada Keamanan Data: Perbandingan Antara SHA-256, SHA-3, dan Blake2,” *JURNAL QUANCOM: Quantum Computer Jurnal*, vol. 2, no. 2, pp. 9–16, 2024, doi: 10.62375/jqc.v2i2.432.

[10] A. M. Ajif, F. Nuraeni, D. Kurniadi, and R. Elsen, “Implementasi Modul Tanda Tangan Digital dengan Superenkripsi RSA-ECDSA dan SHA-512 pada Sistem Informasi Akademik Sekolah,” *Jurnal Algoritma*, vol. 22, no. 2, pp. 933–944, 2025, doi: 10.33364/algoritma/v.22-2.2353.

[11] J. A. Babu, S. Patil, B. D. Parameshachari, S. Rinaldi, K. R. Balmuri, and K. L. Hemalatha, “Blockchain Enabled Hybrid Cryptographic Algorithm for Security and Privacy Preservation of Electronic Health Records,” *ICT Express*, vol. 11, pp. 945–950, 2025, doi: 10.1016/j.icte.2025.08.006.

[12] K. N. Devika and R. Bhakthavatchalu, “Efficient Hardware Prototype of ECDSA Modules for Blockchain Applications,”

TELKOMNIKA (Telecommunication Computing Electronics and Control), vol. 19, no. 5, pp. 1636–1647, 2021, doi: 10.12928/TELKOMNIKA.v19i5.19416.

[13] G. Wu, J. Zhou, and X. Fu, "Improved Blockchain-Based ECDSA Batch Verification Scheme," *Frontiers in Blockchain*, vol. 8, 2025, doi: 10.3389/fbloc.2025.1495984.

[14] R. Santos, B. Lima, and F. Costa, "Spatial Analysis of Mobile Network Quality of Service Using Drive Test Data," *Journal of Network and Computer Applications*, vol. 198, p. 103278, 2022.

[15] A. M. Fajrin and F. Baharuddin, "Analisis Performa Algoritma BLAKE2b dan SHA-256 pada Implementasi Blockchain," 2025.

[16] K. Ramadhani, "Penerapan Teknologi Blockchain dalam Sistem Manajemen Kesehatan Elektronik," *Jurnal Sosial dan Teknologi (SOSTECH)*, vol. 4, no. 2, 2024.