Journal of Artificial Intelligence and Innovative Applications
Vol. 7, No. 3, Agustus 2026
ISSN : 2716-1501 | E-ISSN : 2775-4057
https://doi.org/10.32493/joaiia.v7i3.62982 | Hal : 829-840

Penerapan SHA-256 untuk Analisis Pencegahan Double Spending Pada Cryptocurrency Bitcoin

Arif Rahman Hakim [1], Dola Irwanto [2]

[1,2] Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Pamulang, Tangerang Selatan, 15310, Indonesia
[1]hakimqwerti19@gmail.com, [2]dosen01115@unpam.ac.id

Info Artikel

Riwayat Artikel:

Received July 23, 2026
Revised July 27, 2026
Accepted August 25, 2026

Abstract – *This study aims to analyze the effectiveness of the 256-bit Secure Hash Algorithm (SHA-256) in preventing double-spending attacks during the Bitcoin mining process. This study employs a descriptive qualitative approach that combines a literature review of technical documents with isolated simulation testing based on the Python programming language. The simulation is specifically designed to evaluate parameters such as hash generation, the percentage calculation of the avalanche effect upon data manipulation, block chain formation, nonce discovery for proof-of-work, and Unspent Transaction Output (UTXO) validation scenarios. The testing results indicate that SHA-256 absolutely guarantees block integrity and instantly detects data manipulation, as demonstrated by the even distribution of bit changes. However, the function of this algorithm is proven to not directly act to reject double transactions attempting to use identical inputs. In conclusion, SHA-256 serves as an essential cryptographic foundation for maintaining inter-block structures and network computation, while the overall effectiveness of rejecting double-spending absolutely requires a multi-layered integration of hash functions, UTXO availability status verification, and consensus protocols.*

Keywords: *Bitcoin; Blockchain; Double Spending; Proof-of-work; SHA-256*

Corresponding Author:
Arif Rahman Hakim
Email: hakimqwerti19@gmail.com

Abstrak – Penelitian ini bertujuan menganalisis efektivitas algoritma Secure Hash Algorithm 256-bit (SHA-256) dalam mencegah serangan double spending pada proses mining Bitcoin. Penelitian ini menggunakan pendekatan kualitatif deskriptif yang mengombinasikan studi literatur dari dokumen teknis dengan pengujian simulasi terisolasi berbasis bahasa pemrograman Python. Simulasi tersebut dirancang secara spesifik untuk mengevaluasi parameter pembentukan hash, perhitungan persentase avalanche effect saat terjadi manipulasi data, pembentukan rantai blok, pencarian nonce pada proof-of-work, serta skenario validasi Unspent Transaction Output (UTXO). Hasil pengujian menunjukkan bahwa SHA-256 secara mutlak menjamin integritas blok dan mendeteksi manipulasi secara instan, yang ditunjukkan oleh penyebaran perubahan bit yang merata. Meskipun demikian, fungsi algoritma ini terbukti tidak secara langsung bertindak menolak transaksi ganda yang memakai input identik. Kesimpulannya, SHA-256 merupakan fondasi kriptografis esensial untuk mempertahankan struktur antarblok dan komputasi jaringan, sedangkan efektivitas penolakan double spending secara utuh mutlak membutuhkan integrasi berlapis antara fungsi hash, pemeriksaan status ketersediaan UTXO, dan protokol konsensus.

Kata Kunci*: Bitcoin, Blockchain, Double Spending, Proof-of-work, SHA-256.*

1. PENDAHULUAN

Bitcoin berkembang sebagai sistem pembayaran digital yang memungkinkan transaksi dilakukan secara langsung antar pengguna melalui jaringan *peer-to-peer*[1]. Berbeda dari layanan keuangan konvensional, pengelolaan transaksi tidak berada pada satu lembaga pusat, melainkan tersebar pada *node* yang saling memeriksa data transaksi. Setiap transaksi yang memenuhi aturan jaringan dapat dicatat dalam blockchain sebagai rekam digital yang dapat diverifikasi. Model ini menjadikan Bitcoin tidak hanya dipahami sebagai aset digital, tetapi juga sebagai sistem transaksi yang bergantung pada kriptografi, konsensus, dan partisipasi jaringan untuk menjaga kepercayaan pengguna. Keamanan pada sistem

cryptocurrency tidak cukup dijaga dengan menyimpan data transaksi dalam *blockchain*. Jaringan juga harus memastikan bahwa blok baru dibentuk melalui prosedur yang dapat diperiksa oleh *node* lain dan tidak mudah dimanipulasi[2]. Dalam konteks ini, mekanisme *proof-of-work* berfungsi sebagai dasar komputasi yang membatasi pembentukan blok secara sembarangan. Proses tersebut membuat keamanan Bitcoin berkaitan dengan validitas data, ketahanan mekanisme konsensus, dan kemampuan jaringan mempertahankan rantai blok yang sah. Salah satu ancaman yang perlu diperhatikan dalam transaksi digital adalah *double spending*. Serangan ini terjadi ketika pemilik aset digital mencoba menggunakan input transaksi yang sama untuk lebih dari satu tujuan pembayaran[3]. Masalah tersebut tidak dapat diatasi dengan pemeriksaan saldo sederhana karena Bitcoin tidak memiliki satu server pusat yang mengendalikan seluruh transaksi. Oleh sebab itu, jaringan memerlukan proses validasi yang memastikan setiap input hanya dapat digunakan sekali pada transaksi yang diterima. Risiko *double spending* menjadi lebih besar ketika penerima pembayaran langsung menganggap transaksi yang belum terkonfirmasi sebagai transaksi final. Dalam kondisi tersebut, transaksi pertama masih mungkin bersaing dengan transaksi lain yang memakai input yang sama. Penelitian mengenai pembayaran Bitcoin cepat menunjukkan bahwa transaksi dengan konfirmasi rendah memiliki tingkat kepastian yang lebih lemah dibandingkan transaksi yang sudah masuk ke beberapa blok berikutnya[4]. Karena itu, jumlah konfirmasi perlu dipertimbangkan sebelum suatu pembayaran dianggap aman. Selain transaksi cepat, pembayaran tanpa konfirmasi juga dapat membuka ruang bagi konflik transaksi. Penyerang dapat menyebarkan transaksi yang berbeda dengan menggunakan sumber dana yang sama ke pihak yang berbeda dalam waktu berdekatan. Jaringan pada akhirnya hanya akan menerima satu transaksi yang masuk ke rantai blok sah, sedangkan transaksi lain akan ditolak[5]. Situasi tersebut menunjukkan bahwa kecepatan transaksi perlu diimbangi dengan mekanisme verifikasi yang memadai agar risiko penggunaan aset secara berulang dapat dikurangi. Pencegahan penggunaan input secara berulang dalam Bitcoin dilakukan melalui model Unspent Transaction Output atau UTXO. Setiap transaksi menggunakan keluaran transaksi sebelumnya yang belum dibelanjakan sebagai input baru. Node akan memeriksa status input tersebut sebelum transaksi diteruskan atau dimasukkan ke dalam blok. Apabila input telah digunakan oleh transaksi yang diterima sebelumnya, transaksi berikutnya yang memakai input sama tidak dapat dinyatakan valid[6]. Mekanisme ini membuat UTXO menjadi lapisan pemeriksaan langsung terhadap transaksi yang saling bertentangan .

Penelitian ini menganalisis penerapan SHA-256 dalam proses mining Bitcoin melalui studi literatur dan simulasi terisolasi berbasis Python. Walaupun topik double spending merupakan pengetahuan fundamental, kebaruan (novelty) dari penelitian ini terletak pada pendekatan simulasi isolatif yang secara spesifik memetakan batas efektivitas SHA-256 saat dikomparasikan secara langsung dengan mekanisme *UTXO*. Penelitian sebelumnya seringkali menggabungkan evaluasi keamanan secara umum, sedangkan studi ini menawarkan analisis perbandingan spesifik melalui pengukuran kuantitatif avalanche effect dan visualisasi penolakan transaksi ganda dalam lingkungan parameter yang dikontrol. Pendekatan ini bertujuan untuk memberikan demistifikasi terhadap miskonsepsi bahwa fungsi hash secara tunggal dapat mencegah double spending, serta membuktikan bahwa penolakan transaksi secara langsung mutlak bergantung pada validasi input (*UTXO*) dan konsensus jaringan.

2. TINJAUAN PENELITIAN TERKAIT

Kajian terhadap literatur arsitektur keamanan cryptocurrency menunjukkan bahwa di samping proses validasi transaksi, teknologi *blockchain* sangat mengandalkan fungsi hash kriptografis guna mempertahankan integritas data secara mutlak. Secara konseptual, fungsi hash beroperasi dengan mentransformasikan ragam data input berskala dinamis menjadi suatu nilai keluaran berukuran statis, yang selanjutnya difungsikan sebagai identitas digital dari data tersebut. Nilai hash ini memiliki sensitivitas tinggi dan akan berubah seketika apabila isi data mengalami modifikasi, sehingga sistem dapat mendeteksi setiap bentuk ketidaksesuaian antara data orisinal dengan data yang telah dimanipulasi [7]. Dalam *blockchain*, karakteristik ini berperan penting karena transaksi dan blok harus tetap konsisten setelah dicatat dalam jaringan. *SHA-256* merupakan salah satu algoritma hash yang digunakan dalam Bitcoin. Algoritma ini menghasilkan keluaran sepanjang 256 *bit* dari data masukan yang diproses. Perubahan kecil pada data, misalnya nominal transaksi, alamat penerima, atau nilai *nonce*, menghasilkan hash yang berbeda secara signifikan. Sifat tersebut mendukung pengujian integritas data karena perubahan pada blok dapat diketahui melalui perbedaan hasil hash[8]. Penerapan SHA-256 dalam konteks *blockchain* juga relevan untuk

menunjukkan hubungan antara kepekaan perubahan data dan keamanan informasi digital. Dalam struktur *blockchain*, setiap blok menyimpan hash dari blok yang terbentuk sebelumnya. Pola tersebut membuat blok tidak berdiri sendiri, melainkan menjadi bagian dari rangkaian data yang saling bergantung. Jika isi suatu blok diubah, hash blok tersebut ikut berubah dan tidak lagi sesuai dengan nilai referensi pada blok sesudahnya[9]. Ketidaksesuaian ini dapat digunakan untuk mengenali perubahan riwayat data. Penggunaan SHA-256 pada *blockchain* mendukung pembentukan hubungan antarbIok yang lebih sulit diubah tanpa memengaruhi seluruh rangkaian berikutnya. Pemilihan SHA-256 juga berkaitan dengan kebutuhan keamanan dan efisiensi pada sistem *blockchain*. Algoritma hash perlu memiliki keluaran yang sulit diprediksi, tahan terhadap upaya pencarian kolisi, serta dapat diproses secara memadai dalam lingkungan jaringan terdistribusi[10]. Kajian perbandingan fungsi hash pada *blockchain* menempatkan SHA-256 sebagai algoritma yang memiliki keseimbangan antara kebutuhan keamanan dan proses komputasi. Oleh karena itu, SHA-256 tetap digunakan secara luas dalam mekanisme yang membutuhkan pengamanan data dan validasi blok.

SHA-256 berperan pula dalam mekanisme proof-of-work pada proses mining. Penambang menyusun data blok dan menghitung hash berulang kali dengan mengubah nilai nonce. Blok dapat diterima apabila hash yang dihasilkan memenuhi target kesulitan jaringan. Proses pencarian tersebut tidak dapat diselesaikan dengan menentukan nonce secara langsung sehingga memerlukan sejumlah percobaan komputasi[11]. Dengan demikian, *proof-of-work* menciptakan biaya bagi pihak yang hendak membentuk atau mengubah blok. Kekuatan proof-of-work tidak hanya dipengaruhi oleh algoritma hash, tetapi juga oleh daya komputasi yang tersedia pada jaringan. Semakin besar kekuatan hashing yang mendukung rantai blok sah, semakin sulit pihak lain membangun rantai alternatif untuk menggantikan catatan transaksi yang telah diterima. Kondisi penambang, insentif ekonomi, dan distribusi daya komputasi berpengaruh terhadap kestabilan mekanisme tersebut[12]. Karena itu, keamanan jaringan Bitcoin perlu dipahami sebagai hasil interaksi antara teknologi hashing dan kondisi ekosistem mining . Konfirmasi blok menjadi unsur lain yang memperkuat keamanan transaksi Bitcoin. Setelah transaksi masuk ke sebuah blok, blok baru yang terbentuk di atasnya menambah beban komputasi yang harus diatasi apabila ada pihak yang ingin mengubah transaksi tersebut. Semakin banyak blok konfirmasi yang diperoleh, semakin besar pula usaha yang diperlukan untuk menyusun ulang rantai blok[13]. Proses ini menjelaskan alasan transaksi dengan beberapa konfirmasi dipandang memiliki tingkat kepastian yang lebih tinggi daripada transaksi yang baru disiarkan. Pencegahan double spending dapat diperkuat melalui gabungan mekanisme hashing dan proses konfirmasi berlapis. Hash membantu menjaga agar perubahan data dapat diketahui, sedangkan konfirmasi memberi waktu bagi jaringan untuk menyepakati transaksi dan blok yang sah[14]. Pendekatan ini memperlihatkan bahwa keamanan transaksi digital tidak hanya bergantung pada satu algoritma, tetapi dibangun dari proses verifikasi yang dilakukan secara berulang oleh jaringan. Dengan demikian, SHA-256 berperan sebagai dasar kriptografis yang perlu didukung oleh mekanisme validasi dan konsensus.

Konsensus proof-of-work masih menjadi pendekatan yang relevan untuk membatasi peluang double spending karena setiap blok harus melewati proses komputasi sebelum diterima jaringan. Walaupun demikian, pendekatan tersebut memiliki keterbatasan, antara lain kebutuhan energi dan kemungkinan konsentrasi kekuatan komputasi[15]. Hal ini menunjukkan bahwa evaluasi keamanan tidak cukup dilakukan dengan melihat keandalan SHA-256 secara terpisah. Analisis perlu mempertimbangkan hubungan antara fungsi hash, validasi *UTXO*, *proof-of-work*, konsensus, dan konfirmasi blok dalam satu sistem keamanan Penelitian ini menganalisis penerapan SHA-256 dalam proses mining Bitcoin melalui studi literatur dan simulasi sederhana berbasis Python. Simulasi mencakup pengujian perubahan hash, avalanche effect, pembentukan rantai hash, pencarian nilai nonce, dan pemeriksaan transaksi ganda menggunakan model *UTXO*. Penelitian ini tidak mensimulasikan seluruh kondisi pada jaringan Bitcoin utama, tetapi berfokus memperlihatkan fungsi setiap mekanisme keamanan secara terukur. Hasil analisis diharapkan dapat menjelaskan bahwa *SHA-256* mendukung pencegahan double spending melalui pengamanan integritas data dan *proof-of-work*, sementara penolakan transaksi ganda secara langsung tetap bergantung pada validasi *input* transaksi serta konsensus jaringan.

3. METODE PENELITIAN

Penelitian ini menggunakan pendekatan deskriptif dengan dukungan data kualitatif dan kuantitatif sederhana. Kajian literatur digunakan untuk membangun pemahaman mengenai fungsi SHA-256, blockchain, *proof-of-work*, model *Unspent Transaction Output* (*UTXO*), serta risiko *double spending* pada Bitcoin. Selanjutnya, simulasi dibuat menggunakan Python untuk memperlihatkan penerapan konsep tersebut dalam lingkungan pengujian terbatas. Simulasi tidak ditujukan untuk merepresentasikan jaringan Bitcoin utama secara penuh. Data yang digunakan berupa data uji buatan, sehingga pengujian berfokus pada hubungan antara perubahan data, nilai hash, keterkaitan blok, pencarian *nonce*, dan validasi penggunaan *input* transaksi. Pendekatan ini memungkinkan mekanisme keamanan Bitcoin diamati secara terukur tanpa melakukan penambangan pada jaringan nyata. Penelitian ini menggunakan pendekatan deskriptif dengan dukungan data kualitatif dan kuantitatif sederhana. Simulasi dibangun menggunakan bahasa pemrograman Python (versi 3.14) dan dijalankan pada lingkungan *Jupyter Notebook*. Pengujian dilakukan pada perangkat keras dengan spesifikasi standar (Processor Intel Core i5 12700H setara, RAM 16GB) untuk membuktikan bahwa mekanisme dasar algoritma dapat direplikasi pada komputasi skala menengah.Tahapan penelitian diawali dengan penyusunan struktur data transaksi, blok, dan antrean *UTXO (Unspent Transaction Output)*. Untuk mereplikasi fungsi keamanan, modul hashlib pada Python digunakan untuk mengalkulasi nilai hash. Berikut adalah cuplikan kode (*snippet*) esensial yang digunakan dalam simulasi untuk menghitung *hash* blok dan memvalidasi percobaan *double spending* berdasarkan status *UTXO*:

```
import hashlib
import json

# Fungsi untuk menghasilkan identitas digital (SHA − 256) dari blok
def hitung_hash_blok(blok):
string_blok = json.dumps(blok, sort_keys = True).encode()
return hashlib.sha256(string_blok).hexdigest()

# Fungsi validasi pencegahan Double Spending menggunakan UTXO
def validasi_transaksi(transaksi, utxo_pool):
for input_tx in transaksi['inputs']:
# Memeriksa apakah input ada dan belum terpakai (UNSPENT)
if input_tx not in utxo_pool or utxo_pool[input_tx]['status'] == 'SPENT':
return False, "DITOLAK: Double Spending Terdeteksi! "

# Jika lolos validasi, status UTXO diubah menjadi terpakai
for input_tx in transaksi['inputs']:
utxo_pool[input_tx]['status'] = 'SPENT'
return True, "DITERIMA: Transaksi Valid"
```

Pendekatan di atas memungkinkan mekanisme keamanan *Bitcoin*, khususnya isolasi antara tugas pembuatan hash dan validasi aset, dapat diamati secara terukur tanpa perlu melakukan penambangan langsung pada jaringan utama Bitcoin. Tahapan penelitian diawali dengan penyusunan data transaksi, data blok, daftar *UTXO*, dan parameter tingkat kesulitan *proof-of-work*. Data transaksi kemudian diproses menggunakan algoritma SHA-256 untuk memperoleh nilai hash awal. Sebagian isi transaksi diubah untuk membandingkan perubahan hash yang dihasilkan. Perbedaan kedua hash juga dihitung dalam bentuk jumlah bit berubah dan persentase avalanche effect. Setelah itu, beberapa blok disusun menjadi rantai sederhana dengan menyimpan hash blok sebelumnya pada blok berikutnya. Validasi dilakukan sebelum dan sesudah perubahan data pada salah satu blok untuk menunjukkan dampak manipulasi terhadap kesinambungan rantai. Pengujian *proof-of-work* dilakukan melalui pencarian nonce secara berulang sampai diperoleh hash yang memenuhi pola tingkat kesulitan yang telah ditetapkan. Parameter yang diamati meliputi nonce berhasil, jumlah percobaan, dan waktu proses.

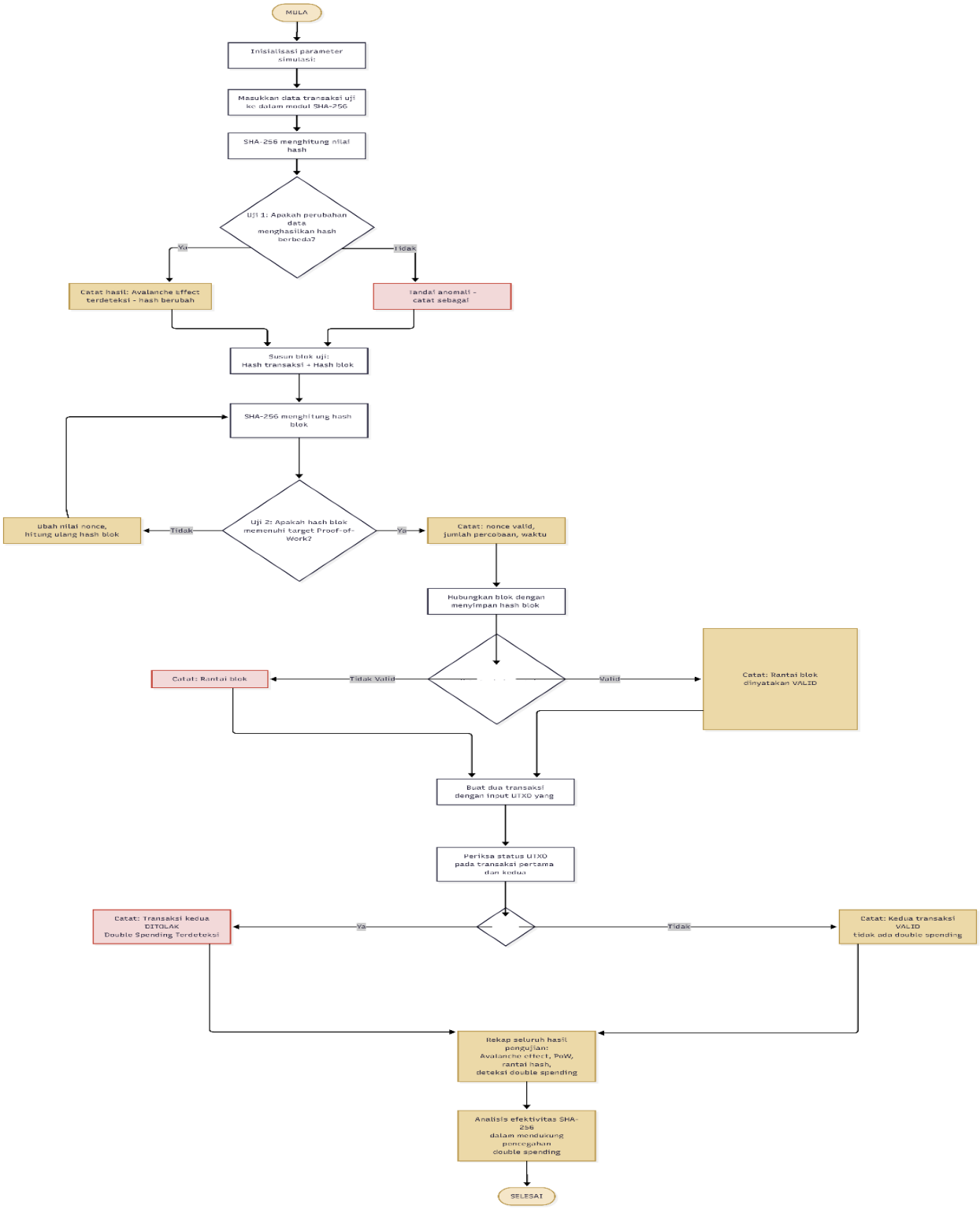
Pengujian terakhir menggunakan model UTXO sederhana. Transaksi pertama yang memakai input tersedia akan diterima dan status input diperbarui menjadi telah digunakan. Transaksi berikutnya yang memakai input identik akan ditolak. Dengan demikian, simulasi menunjukkan bahwa SHA-256 berperan menjaga integritas data dan struktur blok, sedangkan pencegahan penggunaan aset secara berulang secara langsung dilakukan melalui pemeriksaan status input transaksi. Rancangan pengujian tersebut selaras dengan skenario perubahan hash, avalanche effect, validasi rantai blok, proof-of-work, dan transaksi ganda yang diterapkan pada skripsi.

Tabel I Rancangan Skenario Pengujian

No.	Skenario	Prosedur Ringkas	Indikator Hasil
1	Perubahan hash	Membandingkan hash data transaksi awal dan data yang telah diubah	Hash berbeda
2	Avalanche effect	Menghitung bit berbeda antara dua hash	Jumlah bit berubah dan persentase perubahan
3	Validasi rantai blok	Mengubah data pada salah satu blok kemudian memeriksa keterkaitan hash	Status valid atau tidak valid
4	Proof-of-work	Mencari nonce hingga hash memenuhi target kesulitan	Nonce, hash valid, jumlah percobaan, waktu proses
5	Transaksi ganda	Memeriksa dua transaksi yang menggunakan input UTXO sama	Status transaksi diterima atau ditolak

Tahapan pelaksanaan simulasi dapat digambarkan pada Gambar I Alur tersebut menunjukkan bahwa seluruh hasil pengujian dianalisis untuk menjelaskan posisi *SHA-256* dalam keamanan Bitcoin secara proporsional, yaitu sebagai penguat integritas dan *proof-of-work*, bukan sebagai satu-satunya mekanisme penolak double spending.

Journal of Artificial Intelligence and Innovative Applications
Vol. 7, No. 3, Agustus 2026
ISSN : 2716-1501 | E-ISSN : 2775-4057
https://doi.org/10.32493/joaiia.v7i3.62982 | Hal : 829-840



Gambar 1 Alur Sistem

Data yang dipakai bukan data transaksi Bitcoin sesungguhnya, melainkan contoh data yang dibuat khusus agar prinsip kerja *SHA-256* dan *UTXO* mudah diikuti. Susunan data mencakup empat kelompok, yaitu data transaksi, data blok, daftar *UTXO*, dan tingkat kesulitan *proof-of-work*.Data transaksi memuat kode transaksi, input yang dipakai, pihak pengirim, pihak penerima, jumlah aset, serta status

pemeriksaannya. Salah satu contoh yang disiapkan sengaja memakai input yang sama pada dua transaksi berbeda, sehingga dapat dipakai untuk menguji apakah sistem mampu menolak transaksi kedua yang mencoba memakai aset yang sudah terpakai.

Tabel II Perancangan Data Transaksi

No	ID Transaksi	Input Transaksi	Pengirim	Penerima	Jumlah	Status Awal
1	TX001	UTXO001	A	B	1 BTC	Belum diuji
2	TX002	UTXO002	A	C	0,5 BTC	Belum diuji
3	TX003	UTXO001	A	D	1 BTC	Belum diuji
4	TX004	UTXO003	C	B	2 BTC	Belum diuji

Data blok memuat indeks urutan blok, isi transaksi, hash dari blok sebelumnya, nilai nonce, dan hash blok itu sendiri. Setiap blok baru menyimpan jejak hash dari blok yang mendahuluinya, sehingga bila salah satu blok diubah, jejak tersebut menjadi tidak cocok lagi dengan blok berikutnya.

Tabel III Perancangan Data Blok

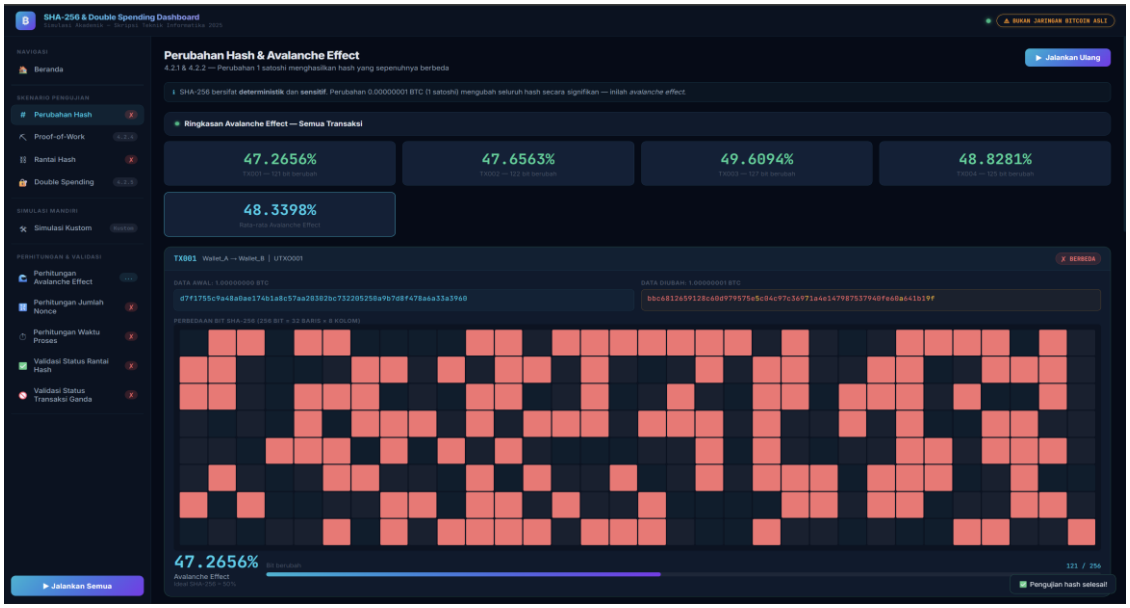
No	Indeks Blok	Data Transaksi	Hash Sebelumnya	Nonce	Hash Saat Ini	Status Awal
1	Blok 1	TX001	0000000000	Dicari sistem	Hasil SHA-256	Belum diuji
2	Blok 2	TX002	Hash Blok 1	Dicari sistem	Hasil SHA-256	Belum diuji
3	Blok 3	TX004	Hash Blok 2	Dicari sistem	Hasil SHA-256	Belum diuji

Daftar *UTXO* dipakai untuk menandai apakah suatu output transaksi masih dapat dipakai atau sudah terpakai. Begitu satu *UTXO* dipakai oleh transaksi yang sah, statusnya berubah menjadi terpakai, dan transaksi lain yang mencoba memakai *UTXO* yang sama akan ditolak.

4. HASIL DAN PEMBAHASAN

Pengujian pertama mengamati respons *SHA-256* terhadap perubahan kecil pada data transaksi. Data awal diproses untuk memperoleh hash pertama, kemudian salah satu unsur data misalnya nominal atau identitas penerima diubah dan diproses ulang. Hasil menunjukkan bahwa *hash* kedua tampil sama sekali berbeda dari hash pertama, tanpa ada kemiripan pola sebagian karakter.Temuan ini membuktikan sifat deterministik dan sensitif *SHA-256*: input yang identik selalu menghasilkan hash identik, tetapi perubahan sekecil apa pun pada input menghasilkan *hash* yang benar-benar baru. Karakteristik ini menjadi dasar mengapa manipulasi data transaksi dapat dideteksi hanya dengan menghitung ulang hash dan membandingkannya dengan nilai yang tersimpan.

Journal of Artificial Intelligence and Innovative Applications
Vol. 7, No. 3, Agustus 2026
ISSN : 2716-1501 | E-ISSN : 2775-4057
https://doi.org/10.32493/joaiia.v7i3.62982 | Hal : 829-840



Gambar 2 hasil pengujian perubahan hash

Pengujian avalanche effect mengonversi hash awal dan hash setelah perubahan ke bentuk biner, lalu membandingkan posisi bit satu per satu. Perhitungan dilakukan dengan rumus:

$$AE = HB \times 100$$

dengan B adalah jumlah bit yang berbeda dan H adalah total 256 bit keluaran *SHA-256*.

Hasil simulasi mencatat terjadinya perubahan pada 127 bit dari total 256 bit keluaran, yang merepresentasikan nilai *avalanche effect* sebesar 49,6094 persen. Pencapaian metrik komputasional ini memiliki signifikansi analitis yang sangat krusial apabila dikomparasikan dengan kerangka teori kriptografi ideal. Berdasarkan literatur keamanan data, sebuah fungsi hash diklasifikasikan mencapai tingkat keamanan optimum dan memenuhi kriteria *strict avalanche criterion (SAC)* jika rasio perubahannya menyentuh probabilitas persis di angka 50 persen. Nilai 49,6% dalam pengujian ini mengonfirmasi temuan teoretis dari berbagai studi terdahulu bahwa arsitektur internal *SHA-256* terbukti sangat tangguh terhadap upaya peretasan melalui analisis kriptanalisis diferensial maupun linier. Implikasi dari tingkat difusi yang nyaris ekuivalen dengan kondisi ideal ini adalah bahwa anomali sekecil apa pun pada struktur data transaksi (misalnya manipulasi nominal input oleh penyerang) akan mendisrupsi seluruh pola keluaran hash secara acak dan asimetris. Alhasil, probabilitas eksploitasi data melalui metode *collision attack* menjadi usang dan tidak relevan untuk dieksekusi. Kendati tingkat resistensi algoritmik ini sangat superior dalam mengamankan integritas entitas blok secara tunggal, perlu digarisbawahi bahwa pencapaian avalanche effect murni bertindak sebagai instrumen pendeteksi mutasi data secara pasif, bukan sebagai eksekutor aktif yang menggagalkan transaksi ganda. Entitas transaksi *double spending* yang disiarkan secara terpisah dengan nominal berbeda tetap akan melahirkan validitas *hash* yang seolah-olah sah; oleh karenanya, determinasi akhir untuk mengeliminasi anomali tersebut sepenuhnya dialihkan pada kapabilitas protokol validasi *Unspent Transaction Output (UTXO)* dalam memverifikasi riwayat aset.

Beberapa blok disusun berurutan dengan setiap blok menyimpan hash blok sebelumnya (previous hash). Aturan validitas rantai dirumuskan sebagai:

$$Pi = Hi - 1$$

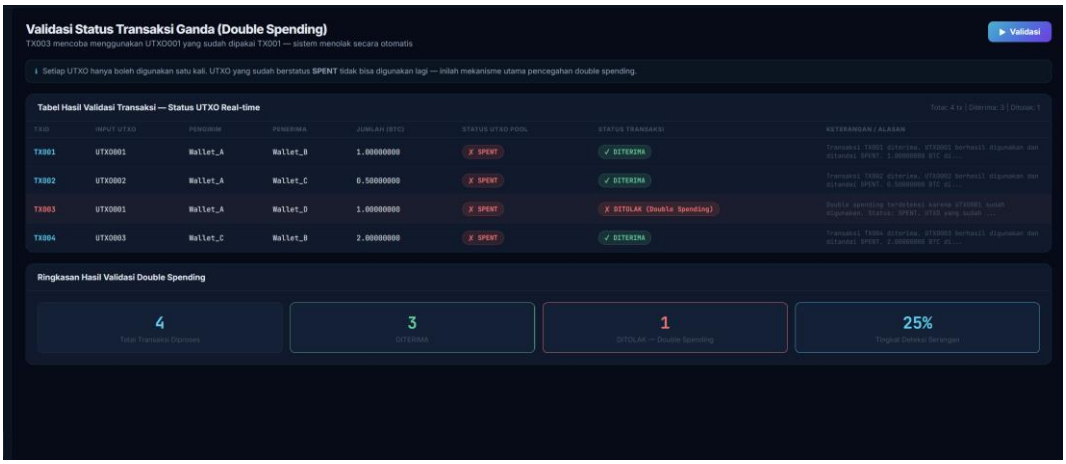
Sebelum ada perubahan data, seluruh blok dinyatakan valid karena nilai previous hash sesuai dengan hash aktual blok pendahulunya. Setelah salah satu blok dimodifikasi, hash aktual blok tersebut berubah sementara previous hash pada blok berikutnya masih mengacu ke nilai lama, sehingga sistem menandai rantai sebagai tidak valid. Temuan ini menegaskan bahwa struktur rantai hash membuat riwayat transaksi

sulit diubah secara diam-diam, karena satu perubahan langsung memutus konsistensi seluruh blok setelahnya.

Pencarian nonce dilakukan secara berulang sampai hash memenuhi target kesulitan, misalnya harus diawali dengan beberapa karakter nol. Jumlah percobaan dihitung dari nilai nonce valid ditambah satu:

$$J = N + 1$$

Peluang teoritis menemukan hash dengan target empat karakter nol adalah 1 berbanding 65.536, karena setiap karakter heksadesimal memiliki 16 kemungkinan nilai. Hasil pengujian pada tiga tingkat kesulitan (diawali 00, 000, dan 0000) menunjukkan pola yang konsisten: semakin banyak nol yang disyaratkan, semakin besar jumlah percobaan dan semakin lama waktu proses yang dibutuhkan. Waktu proses dihitung dari selisih waktu selesai dan waktu $mulai$ ($T = t2 - t1T = t\_2 - t\_1T = t2 - t1$). Pola ini menjelaskan mengapa proof-of-work menambah biaya komputasi bagi siapa pun yang mencoba mengubah blok lama pihak tersebut harus mengulang pencarian nonce dari awal, dan jika blok yang diubah bukan blok terakhir, seluruh blok setelahnya juga perlu diperbarui. Pengujian terakhir memeriksa dua transaksi yang memakai input sama. Transaksi pertama (*TX001*) diterima karena *UTXO001* masih berstatus belum digunakan. Transaksi kedua (*TX003*) yang memakai input identik langsung ditolak karena status *UTXO001* sudah berubah menjadi terpakai. Hasil ini memperlihatkan bahwa penolakan transaksi ganda tidak berasal dari nilai hash, melainkan dari pemeriksaan ketersediaan input pada daftar *UTXO*. *SHA-256* berperan menjaga integritas data, sedangkan validasi *UTXO* berperan langsung menutup peluang penggunaan aset yang sama lebih dari satu kali.



Gambar 3 validasi rantai hash

Kelima hasil pengujian di atas menunjukkan bahwa pencegahan double spending pada Bitcoin bukan hasil kerja satu komponen tunggal, melainkan gabungan beberapa mekanisme yang saling melengkapi. *SHA-256* memastikan setiap data transaksi dan blok memiliki sidik jari digital yang berubah begitu ada modifikasi; avalanche effect memperkuat bukti bahwa perubahan tersebut menyebar luas dan tidak mudah ditebak; rantai hash menjaga agar manipulasi satu blok langsung terdeteksi lewat ketidaksesuaian dengan blok berikutnya; proof-of-work menambah biaya komputasi bagi upaya penulisan ulang blok; sementara validasi *UTXO* menjadi penentu langsung yang menolak penggunaan ulang input transaksi. Dengan demikian, *SHA-256* dapat dipahami sebagai fondasi kriptografis yang menopang integritas dan keterhubungan data, tetapi efektivitas pencegahan double spending secara utuh baru tercapai ketika fungsi hash tersebut bekerja bersama mekanisme proof-of-work, rantai blok, dan pemeriksaan *UTXO* dalam satu sistem keamanan yang berlapis. Sebagai bentuk disklaimer akademis dan batasan interpretasi, penjabaran hasil dari penelitian ini dihadapkan pada sejumlah keterbatasan metodologis yang perlu dipertimbangkan. Eksekusi pengujian algoritma dan mekanisme validasi transaksi dilakukan secara eksklusif di dalam kompartemen simulasi yang terisolasi menggunakan bahasa pemrograman *Python*. Hal ini mengimplikasikan bahwa seluruh metrik operasional yang diobservasi bersumber dari penggunaan set data

sintetis buatan yang tidak terintegrasi secara riil dengan arsitektur *mainnet Bitcoin*. Berbagai parameter esensial pada jaringan terdistribusi yang sesungguhnya—seperti fluktuasi hash rate global, latensi propagasi blok antar-node, kepadatan antrean transaksi pada memory pool *(mempool)*, serta dinamika persaingan daya komputasi antar-penambang *(miners)* tidak diakomodasi di dalam parameter simulasi ini. Oleh sebab itu, generalisasi dari tingkat efektivitas arsitektur keamanan yang didemonstrasikan dalam studi ini harus dimaknai sebagai pembuktian validitas konseptual secara teoretis, yang belum tentu merefleksikan kompleksitas dan anomali performa secara real-time di dalam ekosistem blockchain publik.

5. KESIMPULAN

Kajian atas penerapan *SHA-256* pada proses mining Bitcoin ini memberi gambaran yang cukup jelas mengenai posisi algoritma tersebut dalam mencegah serangan *double spending*. Potensi serangan ini muncul ketika satu unit aset digital dicoba dipakai pada lebih dari satu transaksi sebelum jaringan menyelesaikan proses validasi dan konfirmasi blok secara memadai. Kondisi tersebut memperlihatkan bahwa pencatatan transaksi semata tidak cukup untuk memastikan sebuah aset hanya terpakai satu kali; diperlukan pemeriksaan status input, keterlibatan node dalam validasi, mekanisme konsensus, serta jumlah konfirmasi yang mencukupi.Dalam simulasi yang dibangun, *SHA-256* dipakai untuk membentuk identitas digital pada data transaksi maupun blok. Data seperti isi transaksi, hash blok sebelumnya, waktu, dan nonce diolah melalui fungsi ini hingga menghasilkan keluaran tetap sepanjang 256 bit. Pengamatan menunjukkan bahwa masukan yang sama selalu menghasilkan keluaran yang sama, sementara perubahan sekecil apa pun pada data langsung menghasilkan hash yang berbeda jauh dari sebelumnya. Sifat inilah yang membuat manipulasi pada isi transaksi atau blok dapat terdeteksi hanya dengan menghitung ulang hash lalu mencocokkannya dengan nilai yang telah tersimpan, tanpa perlu memeriksa seluruh data secara manual. Pengukuran avalanche effect memperkuat gambaran tersebut. Dari total 256 bit keluaran, tercatat 127 bit berubah setelah data dimodifikasi sedikit, atau setara 49,6094 persen. Angka yang mendekati separuh ini menandakan difusi yang baik — perubahan kecil pada input menyebar luas ke berbagai posisi bit, bukan berkumpul pada satu bagian tertentu. Dengan pola penyebaran seperti ini, pihak yang ingin memanipulasi data tidak memiliki celah untuk menebak hubungan antara data asli dan *hash* yang dihasilkan.Pada sisi *proof-of-work*, pencarian nonce memerlukan sejumlah percobaan berulang sampai hash memenuhi target kesulitan yang ditetapkan, misalnya harus diawali beberapa karakter nol. Karena setiap percobaan hash bersifat tidak dapat diprediksi, proses ini menuntut biaya komputasi yang nyata — semakin tinggi target kesulitan, semakin banyak percobaan dan semakin lama waktu yang dibutuhkan. Beban komputasi inilah yang membuat perubahan pada blok lama menjadi mahal untuk dilakukan, sebab pelaku harus mengulang pencarian nonce dari awal, bahkan untuk seluruh blok yang mengikutinya jika blok yang diubah bukan blok terakhir.

Pengujian rantai hash menegaskan hal serupa dari sisi lain. Setiap blok menyimpan hash dari blok sebelumnya sebagai penghubung; ketika satu blok dimodifikasi, hash aktualnya berubah namun referensi pada blok berikutnya masih mengacu ke nilai lama, sehingga sistem langsung menandai rantai sebagai tidak konsisten. Temuan ini memperlihatkan bahwa struktur rantai hash berperan menjaga keutuhan riwayat transaksi secara keseluruhan, bukan hanya pada satu blok yang berdiri sendiri.Adapun penolakan transaksi ganda dalam simulasi ini justru ditentukan oleh pemeriksaan status *UTXO*, bukan oleh nilai hash. Transaksi pertama diterima karena input yang dipakai masih berstatus belum terpakai; begitu status tersebut berubah menjadi terpakai, transaksi kedua yang memakai input identik langsung ditolak. Dari sini terlihat pembagian peran yang jelas: SHA-256 menjaga keaslian dan keterhubungan data, sementara keputusan menerima atau menolak suatu transaksi tetap bergantung pada pemeriksaan ketersediaan input. Menyatukan seluruh temuan tersebut, dapat dikatakan bahwa efektivitas *SHA-256* dalam mencegah double spending bersifat mendukung, bukan berdiri sendiri. Algoritma ini menjadi fondasi kriptografis yang memperkuat integritas data dan keterikatan antarblok, sedangkan pencegahan double spending secara utuh baru terwujud melalui kerja bersama antara fungsi *hash*, *proof-of-work*, rantai blok, pemeriksaan *UTXO*, validasi transaksi, dan konsensus jaringan Beberapa hal berikut dapat dipertimbangkan untuk melanjutkan atau memperluas kajian ini.

1. Model validasi transaksi pada penelitian ini masih terbatas pada pemeriksaan status UTXO; penelitian berikutnya dapat menambahkan verifikasi tanda tangan digital agar proses pengesahan transaksi lebih mendekati praktik nyata pada jaringan Bitcoin.
2. Hasil simulasi ini tidak dapat disamakan dengan kondisi keamanan jaringan Bitcoin yang sesungguhnya, mengingat data uji, lingkungan pengujian, dan tingkat kesulitan yang dipakai jauh lebih sederhana; kajian lanjutan sebaiknya memanfaatkan data blockchain publik, jaringan testnet, atau perangkat analisis blockchain agar pengamatan lebih mendekati kondisi operasional sebenarnya
3. Perbandingan SHA-256 dengan mekanisme konsensus lain, seperti proof-of-stake, atau dengan fungsi hash lain dalam konteks keamanan blockchain, dapat menjadi arah kajian selanjutnya untuk memperluas pemahaman mengenai pengaruh desain konsensus dan daya komputasi terhadap ketahanan sistem terhadap manipulasi transaksi.
4. Pengujian tingkat kesulitan proof-of-work dapat diperluas dengan variasi target yang lebih beragam, disertai pencatatan pengaruh spesifikasi perangkat keras yang berbeda terhadap waktu pencarian nonce, agar hasil dapat dibandingkan pada berbagai kondisi komputasi.
5. Simulasi ke depan dapat mencoba skenario serangan yang lebih beragam, tidak hanya dua transaksi dengan input identik, melainkan juga percabangan blok (fork) sederhana untuk menggambarkan bagaimana jaringan menentukan rantai yang dianggap sah ketika muncul lebih dari satu versi blok.
6. Penelitian selanjutnya dapat pula menguji ketahanan simulasi ini terhadap skenario serangan lain di luar double spending, misalnya serangan penundaan transaksi atau manipulasi urutan blok, untuk melihat apakah kombinasi SHA-256, proof-of-work, dan model UTXO tetap memadai atau memerlukan mekanisme tambahan.

DAFTAR PUSTAKA

[1] S. Han, "Revolutionizing finance with bitcoin and blockchain : a literature review and research agenda," vol. 26, no. 4, pp. 413–430, 2023, doi: 10.1108/CAFR-04-2023-0044.

[2] C. Yu, W. Yang, F. Xie, and J. He, "Technology and Security Analysis of Cryptocurrency Based on Blockchain," vol. 2022, 2022, doi: 10.1155/2022/5835457.

[3] N. A. Akbar, A. Muneer, N. Elhakim, and S. M. Fati, "Distributed Hybrid Double-Spending Attack Prevention Mechanism for Proof-of-Work and Proof-of-Stake Blockchain Consensuses," 2021.

[4] D. Melo, S. E. Pomares-hernández, L. M. Rodríguez-henríquez, and J. C. Pérez-sansalvador, "DiFastBit : Transaction Differentiation Scheme to Avoid Double-Spending for Fast Bitcoin Payments," pp. 1–33, 2024.

[5] A. F. Elessawy, I. Gad, H. Abdul-kader, and A. Elsaid, "Double Spending Attacks in Decentralized Digital Currencies : Challenges and Countermeasures," vol. 10, 2023.

[6] H. Hashim, A. R. Alzighaibi, A. F. Elessawy, I. Gad, H. Abdul-kader, and A. Elsaid, "Securing Financial Transactions with a Robust Algorithm : Preventing Double-Spending Attacks," pp. 1–17, 2023.

[7] A. G. Gad, D. T. Mosa, L. Abualigah, and A. A. Abohany, "Emerging Trends in Blockchain Technology and Applications : A Review and Outlook," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, no. 9, pp. 6719–6742, 2022, doi: 10.1016/j.jksuci.2022.03.007.

[8] R. K. Salih and A. H. Kashmar, "Enhancing Blockchain Security by Developing the SHA256 Algorithm SHA256 خوارزمية تطوير من خالل تعزيز أم ا ن سلسلة الكتل," vol. 65, no. 10, pp. 5678–5693, 2024, doi: 10.24996/ijs.2024.65.10.30.

[9] C. E. B. Santos, L. M. D. Silva, M. F. Torquato, S. N. Silva, and M. A. C. Fernandes, "SHA-256 Hardware Proposal for IoT Devices in the Blockchain Context," pp. 1–25, 2024.
[10] P. K. Yien, K. Malik, and S. N. Ramli, "Comparative Study on Hash Function Algorithms for Blockchain Technology," vol. 5, no. 1, pp. 16–33, 2024.
[11] S. Shahrizoda, "INTERNATIONAL JOURNAL OF ARTIFICIAL INTELLIGENCE," vol. 05, no. 03, pp. 1660–1664, 2025.
[12] J. Stinner and M. Tyrell, "Proof-of-work Consensus Under Exogenous Distress : Evidence from Bitcoin Mining Shocks ∗," pp. 1–53, 2025.
[13] D. Bazzanella and A. Gangemi, "Bitcoin : a new proof - of - work system with reduced variance," *Financ. Innov.*, 2023, doi: 10.1186/s40854-023-00505-2.
[14] O. Kuznetsov, A. Yezhov, V. Yusiuk, and K. Kuznetsova, "Scalable Zero-Knowledge Proofs for Verifying Cryptographic Hashing in Blockchain Applications," vol. 6326, pp. 0–3, 2024.
[15] J. Ye, "Analyzing Blockchain Consensus Mechanisms for Double-Spending Attack Prevention," no. Daml 2024, pp. 432–437, 2025, doi: 10.5220/0013525500004619.