Implementasi Teori Bilangan dalam Blockchain dan Crypto Currency

Studi Kasus:Transaksi Bitcoin(Public Distributed Ledger dan pengecekan jika transaksi invalid)

Yohanes Nicholas Setiawan - 13525048
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail: yohanes12407@gmail.com , 13525048@std.stei.itb.ac.id

Abstract—**Perkembangan teknologi blockchain dan cryptocurrency seperti Bitcoin telah mengubah paradigma sistem transaksi digital menjadi terdesentralisasi tanpa melibatkan otoritas pusat. Keamanan dan integritas sistem ini sepenuhnya bersandar pada fondasi matematika yang kokoh. Makalah ini menganalisis implementasi teori bilangan dan struktur logika diskrit yang mendasari keamanan jaringan Bitcoin, dengan fokus pada algoritma Secure Hash Algorithm 256-bit (SHA-256), Elliptic Curve Digital Signature Algorithm (ECDSA), struktur data Merkle Tree, serta mekanisme konsensus Proof of Work (PoW). Melalui pendekatan studi kasus, makalah ini membedah bagaimana sifat keacakan bilangan prima digunakan dalam inisialisasi konstanta SHA-256 dan bagaimana fungsi aritmetika modular diterapkan dalam proses pembatasan panjang pesan (padding). Lebih lanjut, makalah ini menguraikan bagaimana matematika digunakan secara asimetris pada pasang kunci ECDSA untuk menjamin keaslian kepemilikan koin, serta bagaimana ketidakcocokan persamaan hash bertingkat pada Merkle Tree digunakan oleh nodes untuk mendeteksi dan menolak transaksi invalid atau manipulasi data secara otomatis. Hasil analisis menunjukkan bahwa sifat deterministik, irreversibilitas kriptografi, dan konsensus terdistribusi berbasis matematika diskrit berhasil menciptakan sistem public distributed ledger yang aman, transparan, dan tahan terhadap ancaman pemalsuan (double-spending).**

Kata Kunci—Blockchain, Bitcoin, SHA-256, Aritmatika Modular, Bilangan Prima, ECDSA, Transaksi Invalid.

I. Pendahuluan

A. *Latar Belakang*

Perkembangan teknologi pada era ini telah membawa banyak perubahan mendasar di kehidupan manusia, salah satunya dalam bertransaksi barang/perdagangan barang dan cara bayarnya. Salah satu inovasi paling revolusioner pada zaman kini adalah *blockchain* dan *cryptocurrency*, yang diperkenalkan oleh seorang jenius teknologi yaitu, Satoshi Nakamoto pada tahun 2008 dan diresmikan pada tahun 2009 dengan makalahnya yang berjudul "Bitcoin: A Peer-to-Peer Electronic Cash System".

Dibalik keamanan dari sebuah blockchain dan cryptocurrency, terdapat model matematika yang disusun dan dipakai untuk membuat sebuah algoritma kriptografi unik yang hingga kini masih digunakan untuk memvalidasi setiap transaksi digital kita, salah satu yang saya akan bahas adalah algoritma SHA-256 yang digunakan pada salah satu cryptocurrency yang paling tua dan populer yaitu, Bitcoin. Algoritma SHA-256 ini dirancang untuk mengamankan data, membentuk struktur data Merkle Tree, hingga proses penambangan (*mining*) blok baru. Secara singkat, algoritma ini bekerja dengan mengubah data berukuran sembarang menjadi nilai hash sepanjang 256-bit yang bersifat unik, satu arah, dan sangat sensitif terhadap perubahan input sekecil apapun.

Namun demikian, pemahaman mendalam mengenai bagaimana teori bilangan diterapkan secara konkret dalam sistem blockchain masih belum banyak dibahas secara akademis di tingkat mahasiswa. Sebagian besar literatur yang tersedia cenderung membahas aspek teknis implementasi tanpa mengaitkannya secara eksplisit dengan konsep matematika yang mendasarinya.

Berdasarkan latar belakang tersebut, saya tertarik untuk mengkaji lebih dalam mengenai implementasi teori bilangan dalam sistem blockchain dan cryptocurrency, dengan fokus pada mekanisme kriptografi dalam alur transaksi, mulai dari proses hashing, enkripsi menggunakan private key, verifikasi melalui public key, hingga pencatatan pada distributed ledger.

B. *Rumusan Masalah*

Berdasarkan latar belakang yang telah diuraikan, makalah ini merumuskan permasalahan sebagai berikut:

1. Bagaimana konsep teori bilangan, khususnya aritmatika modular dan sifat bilangan prima, diterapkan dalam algoritma SHA-256 pada sistem blockchain?
2. Bagaimana mekanisme kriptografi menggunakan private key dan public key menjamin keamanan dan keaslian transaksi dalam jaringan blockchain?

3. Bagaimana alur lengkap sebuah transaksi — mulai dari pembentukan data, proses hashing, penandatanganan digital, verifikasi, hingga pencatatan pada distributed ledger — bekerja secara matematis.

C. *Tujuan Penulisan*

Berdasarkan Latar Belakang yang telah ditulis maka tujuannya dapat diuraikan secara berikut:

1. Menjelaskan peran teori bilangan sebagai fondasi matematis dari algoritma SHA-256 yang digunakan dalam sistem blockchain.
2. Menganalisis mekanisme enkripsi dan verifikasi transaksi menggunakan pasangan *private key* dan *public key* berbasis ECDSA.
3. Memaparkan alur transaksi blockchain secara menyeluruh dan mengaitkan setiap tahapannya dengan konsep matematika yang mendasarinya.

II. Landasan Teori

A. *Teori Aritmatika Modular*

Kriptografi modern bertumpu pada beberapa konsep teori bilangan. Aritmetika modular mendefinisikan relasi kongruensi sebagai berikut:

$$a \equiv b \pmod{n}$$

yang berarti a dan b memiliki sisa yang sama ketika dibagi n. Sebagai contoh, $17 \equiv 2 \pmod 5$ karena $17 \bmod 5 = 2$. Operasi ini menjadi dasar dari hampir seluruh algoritma kriptografi yang digunakan dalam blockchain.

B. *Fungsi Hash*

Pada algoritma SHA256(Secure Hash 256), Fungsi hash merupakan fondasi utama dari algoritma SHA256 yang bekerja dengan memetakan data masukan berukuran sembarang ke keluaran berukuran tetap. Proses SHA-256 terdiri dari empat tahap utama:

1. Pre-processing: pesan dipadding hingga panjangnya kongruen dengan 448 (mod 512), lalu ditambahkan representasi 64-bit dari panjang pesan asli
2. Inisialisasi: delapan konstanta hash awal H_0–H_7 diambil dari 32-bit pertama bagian fraksional akar kuadrat delapan bilangan prima pertama (2, 3, 5, 7, 11, 13, 17, 19)
3. Kompresi: 64 putaran operasi bitwise (AND, OR, XOR, ROTR) terhadap blok 512-bit, menggunakan 64 konstanta K_0–K_{63} yang diturunkan dari akar kubik 64 bilangan prima pertama
4. Output: nilai hash 256-bit final

C. *Algoritma RSA*

Bitcoin menggunakan **Elliptic Curve Digital Signature Algorithm (ECDSA)** dengan kurva secp256k1 untuk pembangkitan pasangan kunci. Hubungan antara private key dan public key dinyatakan sebagai:

$$K = k \cdot G$$

di mana K adalah public key, k adalah private key (bilangan bulat 256-bit), dan G adalah titik generator pada kurva eliptik. Keamanan sistem ini didasarkan pada sulitnya **Elliptic Curve Discrete Logarithm Problem (ECDLP)** operasi perkalian mudah dihitung maju, namun secara komputasi tidak mungkin dibalik.

D. *Struktur Merkle Tree*

Merkle Tree adalah struktur data pohon biner di mana setiap node interior menyimpan hash dari kedua child-nya. Root dari pohon ini, disebut **Merkle Root**, merangkum seluruh transaksi dalam satu blok ke dalam satu nilai hash tunggal. Properti ini memungkinkan verifikasi parsial transaksi (Simplified Payment Verification) tanpa mengunduh seluruh blockchain.

III. Pembahasan

A. Gambaran Umum Alur Transaksi Blockchain (Awal-Akhir)

Misal kita ambil contoh si A ingin membayar tagihan makan siang sebanyak 2 btc ke si B, Langkah-langkah yang dilakukan oleh sistem secara kasar akan seperti berikut:

1. Membuat Input dan Transaksi
   - Mencari UTXO(Unspent Transaction Output)

     Wallet/dompet digital A akan memindai pecahan2 mata uang yang belum digunakan oleh si A pada yang waktu yang lalu, kemudian sistem mendeteksi bahwa funds/saldo si A adalah 10 btc.

   - Membuat Transaksi(Tujuan Transfer)

     Dari data yang telah diambil tadi, akan dibuat sebuah transaksi yang berisi alamat yang ingin dituju, seberapa banyak cryptocurrency yang diberikan dan kembalian yang diterima oleh si A(saldo baru=saldo -jumlah transfer - Fee transfer
2. Penandatanganan Secara Digital dan Pengamanan Kriptografi
   - Proses Double SHA-256 Pertama

     Seluruh data mentah transaksi (Input, Output, batasan waktu, dll.) digabungkan menjadi satu kesatuan data. Data ini kemudian di-hash menggunakan SHA-256 dua kali berturut-turut. Hasilnya adalah sebuah string 64 karakter yang merepresentasikan "isi ringkasan transaksi".

   - Penandatanganan dengan ECDS

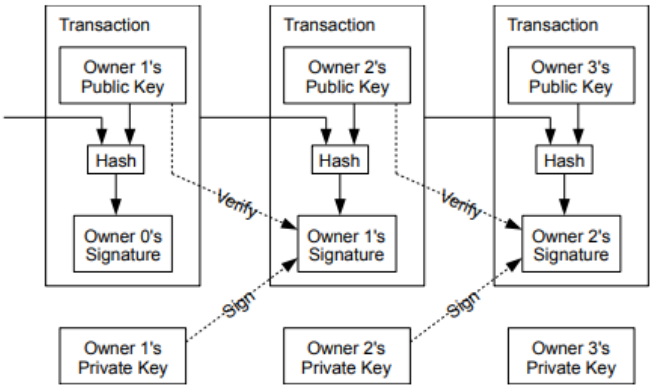
     Wallet mengambil Private Key milik si A (yang disimpan rahasia di dalam aplikasinya). Menggunakan algoritma matematika ECDSA (Elliptic Curve Digital Signature Algorithm), Private Key si A mengunci hasil hash transaksi tadi. Proses ini menghasilkan Digital Signature (Tanda Tangan Digital).

   - Memasukkan Kunci Publik

Wallet kemudian menempelkan Digital Signature dan Public Key si A ke dalam bagian ScriptSig (kolom pembuktian) pada transaksi tersebut.

- Membuat TxID (ID Transaksi)

  Terakhir, seluruh dokumen transaksi yang sudah lengkap ditandatangani itu di-hash lagi menggunakan Double SHA-256. Hasil hash inilah yang disebut TxID (misalnya: e9a3...8f1a). TxID berfungsi sebagai nomor resi atau KTP transaksi tersebut.



3. Penyiaran dan Validasi Awal oleh Jaringan
   - Broadcasting

     Wallet si A akan mengirim data transaksi ini ke beberapa komputer (nodes) terdekat yang terhubung dengannya di jaringan Bitcoin P2P (Peer-to-Peer).

   - Validasi Independen oleh Node

     Setiap node yang menerima transaksi si A tidak langsung percaya. Mereka akan melakukan pengecekan ketat secara mandiri berdasarkan aturan protokol Bitcoin (Bitcoin Core)

     - Komputer lain akan mengecek, Apakah matematika ECDSA-nya cocok? Node menggunakan Public Key si A untuk memverifikasi Digital Signature. Jika rumusnya pas, terbukti Alice yang punya koin itu.
     - Jika Tidak terbukti maka proses nya akan dilanjutkan, komputer lain akan mengecek juga, apakah funds si A yang tadi sudah dipakai apa belum? Hal ini sangat baik karena untuk mencegah double spending(saldo yang sama dapat melakukan 2 transaksi yang berbeda)

   - Masuk ke Mempool

     Jika lolos dari pengecekan 2x oleh node lain, maka node akan memasukkan transaksi si A ke dalam Mempool (Memory Pool) mereka sendiri, lalu meneruskan (relay) transaksi tersebut ke komputer tetangga mereka. Dalam hitungan detik, transaksi si A sudah tersebar dan mengantri di ribuan Mempool komputer di seluruh dunia

4. Penyusunan Blok oleh Penambang (The Merkle Tree/struktur data Block)
   - Pemilihan Transaksi

     Karena kapasitas satu blok terbatas (sekitar 1-4 MB dengan SegWit), Miner akan menyortir Mempool dan memilih transaksi-transaksi yang menawarkan Fee paling tinggi agar keuntungan mereka maksimal.

   - Membuat Merkle Tree (Struktur Pohon SHA-256)

     Ribuan transaksi yang terpilih tidak dimasukkan secara acak. Mereka dipasangkan dua-dua dan di-hash berulang kali menggunakan SHA-256(awal mula pembuatan Block dalam Blockchain).

   - Fungsi Merkle Root

     mengunci integritas seluruh transaksi di dalam blok yang sedang dibuat.

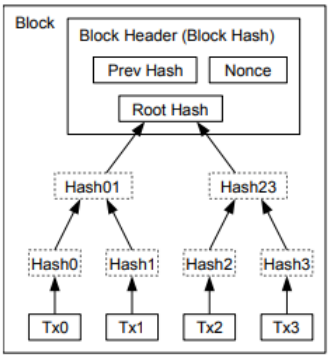
5. Proof of Work atau yang sering disebut Mining
   - Mencari Hash yang Sesuai

     Tugas Miner adalah memasukkan seluruh data Block Header tersebut ke dalam mesin Double SHA-256 dan menebak angka Nonce.

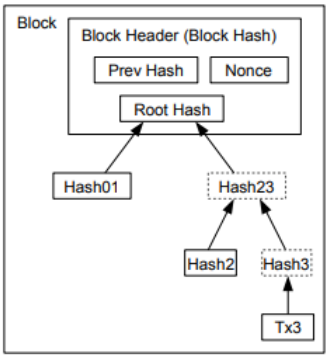
     Hasil Hash=(SHA256(SHA256(Block Header Data+Nonce)))

     Jika komputer berhasil menemukan Nonce maka akan dijanjikan sebuah reward yaitu sebuah bitcoin jika proses validasi yang akan dilanjutkan selanjutnya berhasil

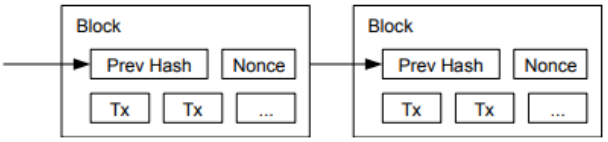
6. Penyebaran Blok & Konsensus Global (Public Distributed Ledger)
   - Intinya komputer/node2 lain akan memvalidasi block tadi dan menyambungkan headnya ke block yang lama dan akan tercatat secara abadi dan miner akan mendapatkan hadiah yang dijanjikan



Transactions Hashed in a Merkle Tree



After Pruning Tx0-2 from the Block



B. *Analisis Matematis Pembangkitan Kunci dan Verifikasi (ECDSA)*

Sistem kriptografi kunci publik pada blockchain Bitcoin dibangun di atas **Elliptic Curve Cryptography (ECC)** dengan kurva secp256k1 yang didefinisikan sebagai:

$y^2 = x^3 + 7 \pmod p$

di mana $p = 2^{256} - 2^{32} - 977$ adalah bilangan prima besar.

Pembangkitan Kunci

Private key dipilih sebagai bilangan bulat acak k dalam rentang [1, n-1] di mana n adalah orde dari titik generator G. Public key kemudian dihitung sebagai:

$K = k \cdot G$

Operasi $k \cdot G$ merupakan penjumlahan titik kurva eliptik sebanyak k kali. Keamanan sistem ini sepenuhnya bergantung pada **Elliptic Curve Discrete Logarithm Problem (ECDLP)** diberikan K dan G, secara komputasi tidak mungkin menemukan k kembali.

Pembangkitan Signature

Untuk menandatangani transaksi dengan TxID = z, Alice melakukan:

Pilih nonce acak k (bilangan acak sekali pakai)

Hitung titik: $(x_1, y_1) = k \cdot G$

Hitung: $r = x_1 \bmod n$

Hitung: $s = k^{-1} \cdot (z + r \cdot \text{privateKey}) \bmod n$

Pasangan (r, s) merupakan digital signature yang disertakan bersama transaksi.

Verifikasi Signature

Node penerima memverifikasi signature (r, s) menggunakan public key si A:

Hitung: $w = s^{-1} \bmod n$

Hitung: $u_1 = z \cdot w \bmod n$

Hitung: $u_2 = r \cdot w \bmod n$

Hitung titik: $(x_1, y_1) = u_1 \cdot G + u_2 \cdot \text{PublicKey}$

Signature valid jika dan hanya jika: $r \equiv x_1 \pmod n$

Keindahan matematis skema ini adalah siapapun dapat memverifikasi menggunakan public key, namun hanya si A yang bisa menghasilkan signature valid karena hanya ia yang mengetahui private key-nya.

C. *Implementasi Teori Bilangan dan Operasi Bitwise pada Algoritma SHA-256*

SHA-256 memproses data dalam blok 512-bit melalui empat tahap utama, di mana teori bilangan berperan langsung pada setiap tahapnya.

Keterkaitan Bilangan Prima dengan Konstanta SHA-256

SHA-256 menggunakan dua set konstanta yang keduanya diturunkan dari bilangan prima:

Konstanta	Sumber	Hasil
H_0–H_7	32-bit pertama dari $\sqrt{}$bilangan prima ke-1 s.d. ke-8	8 nilai
H_0–K_{63}	32-bit pertama dari $\sqrt[3]{}$bilangan prima ke-1 s.d. ke-64	64 nilai

Contoh untuk H_0 dari bilangan prima pertama (p = 2):
$\sqrt{2} = 1.41421356... \rightarrow$ bagian fraksional = 0.41421356...
$0.41421356 \times 2^{32} = 1779033703 = 0x6a09e667 \leftarrow$ nilai H_0
Penggunaan bilangan prima sebagai sumber konstanta menjamin nilai-nilai ini bersifat **nothing-up-my-sleeve** tidak dipilih sembarangan dan bebas dari backdoor tersembunyi.

Operasi Bitwise dalam 64 Putaran Kompresi

Inti SHA-256 adalah 64 putaran kompresi yang menggunakan enam fungsi bitwise berikut:
Ch(e,f,g) = (e AND f) XOR (NOT e AND g)
Maj(a,b,c) = (a AND b) XOR (a AND c) XOR (b AND c)
$\Sigma_0(a) = \text{ROTR}^2(a) \text{ XOR } \text{ROTR}^{13}(a) \text{ XOR } \text{ROTR}^{22}(a)$
$\Sigma_1(e) = \text{ROTR}^6(e) \text{ XOR } \text{ROTR}^{11}(e) \text{ XOR } \text{ROTR}^{25}(e)$
$\sigma_0(w) = \text{ROTR}^7(w) \text{ XOR } \text{ROTR}^{18}(w) \text{ XOR } \text{SHR}^3(w)$
$\sigma_1(w) = \text{ROTR}^{17}(w) \text{ XOR } \text{ROTR}^{19}(w) \text{ XOR } \text{SHR}^{10}(w)$
Setiap putaran memperbarui delapan variabel kerja (a–h) sebagai berikut:
$T_1 = h + \Sigma_1(e) + Ch(e,f,g) + K_i + W_i$
$T_2 = \Sigma_0(a) + Maj(a,b,c)$
$h = g, g = f, f = e, e = d + T_1$
$d = c, c = b, b = a, a = T_1 + T_2$

Avalanche Effect — Bukti Empiris

Perubahan 1 bit pada input menghasilkan perubahan total pada output, contohnya:
Input : "Alice→Bob:1BTC"
SHA-256: 3a7bd3e2360a3d29eea436fcfb7e44c7...
Input : "alice→Bob:1BTC" ← hanya beda kapital 'A'
SHA-256: 9f86d081884c7d659a2feaa0c55ad015…
Sifat ini secara langsung menjamin bahwa tidak ada dua transaksi berbeda yang dapat menghasilkan TxID yang sama.

IV. KESIMPULAN

Berdasarkan pembahasan yang telah diuraikan, dapatditarik kesimpulan sebagai berikut:

1. Teori bilangan, khususnya aritmatika modular dan sifat bilangan prima, memiliki peran fundamental dalam algoritma SHA-256. Konstanta-konstanta yang digunakan dalam SHA-256 diturunkan secara langsung dari bilangan prima, menjamin sifat pseudorandom yang mencegah adanya backdoor tersembunyi dan memastikan integritas setiap transaksi yang diproses.

2. Keamanan mekanisme enkripsi dan verifikasi transaksi blockchain bertumpu pada sulitnya memecahkan Elliptic Curve Discrete Logarithm Problem (ECDLP). Pasangan private key dan public key yang dibangkitkan melalui ECDSA memungkinkan siapapun untuk memverifikasi keaslian transaksi, namun hanya pemilik private key yang dapat menghasilkan tanda tangan digital yang valid.
3. Alur transaksi blockchain merupakan rangkaian proses matematis yang saling bergantung dari pembentukan data transaksi, double hashing SHA-256, penandatanganan dengan private key, verifikasi dengan public key, hingga pencatatan permanen pada distributed ledger melalui mekanisme Proof of Work. Setiap tahap dirancang sedemikian rupa sehingga kegagalan pada satu komponen kriptografis akan langsung terdeteksi oleh seluruh jaringan.

References

[1] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," 2008. [Online]. Available: https://bitcoin.org/ bitcoin.pdf

[2] National Institute of Standards and Technology (NIST), "Secure Hash Standard (SHS)," FIPS PUB 180-4, Aug. 2015.

[3] D. Johnson, A. Menezes, and S. Vanstone, "The Elliptic Curve Digital Signature Algorithm (ECDSA)," International Journal of Information Security, vol. 1, pp. 36–63, 2001.

[4] R. L. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," Communications of the ACM, vol. 21, no. 2, pp. 120–126, Feb. 1978.

[5] A. Back, "Hashcash - A Denial of Service Counter-Measure," 2002. [Online]. Available: http://www.hashcash.org/papers/ hashcash.pdf

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 1 Juni 2025

Yohanes Nicholas Setiawan, 13525048