Aplikasi Algoritma Hashing dan Merkle Tree dalam Sistem Blockchain

Timothy 13517087
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13517087@std.stei.itb.ac.id

Abstrak*—Perdagangan menjadi salah satu tulang punggung dari aktivitas ekonomi dunia. Perdagangan pada umumnya mengandung unsur pertukaran barang dengan barang lainnya, dapat berupa uang ataupun dalam bentuk lain. Semua transaksi ini didata oleh pihak yang berwenang dalam suatu bentuk yang kita kenal sebagai *ledger*. Sistem *ledger* yang kita kenal saat ini biasanya menganut sistem sentralisasi, dimana semua catatan dibuat dan dikontrol oleh satu pihak saja. Mata uang digital menawarkan sistem pencatatan yang berbeda dengan sistem *ledger* yang kita kenal sekarang. Sistem pencatatan yang ditawarkan ini menganut sistem desentralisasi, dimana catatan transaksi dikontrol oleh banyak pengguna internet dari seluruh penjuru dunia. Selain sistem desentralisasi, mata uang digital juga menawarkan fitur keamanan melalui enkripsi yang tidak dapat kita temui pada sistem *ledger* yang kita pakai sekarang. Tetapi kenyataannya, mata uang digital ini masih menjadi bahan perdebatan. Sistem terdesentralisasi menyebabkan tidak adanya pihak yang dapat mengatur kestabilan dari nilai mata uang ini. Fluktuasi harga masing masing mata uang disebabkan oleh perbandingan banyaknya *supply and demand* mereka. Sekarang ini, sudah ada beberapa negara yang melegalkan mata uang ini sebagai alat tukar perdagangan. Di Indonesia sendiri, Bank Indonesia menyatakan bahwa *Bitcoin* dan *Virtual Currency* lainnya bukan merupakan mata uang yang sah sebagai alat tukar, sehingga segala resiko kepemilikan dan penggunaannya menjadi tanggungan masing-masing pribadi. Salah satu dari penyebabnya adalah belum adanya regulasi yang jelas mengenai penggunaannya.

Kata kunci—blockchain, proof-of-work, hash, SHA256, cryptocurrency .

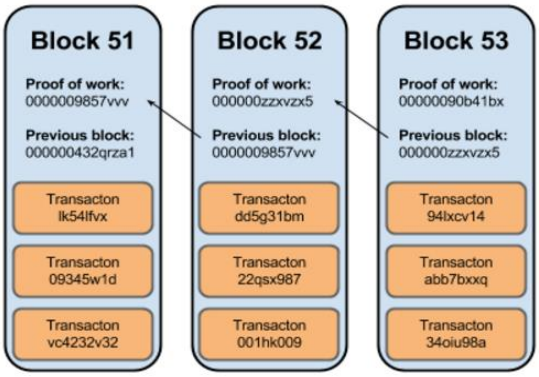
I. PENDAHULUAN

Istilah *blockchain* mungkin sudah tidak terdengar asing lagi. Beberapa dari anda mungkin sudah mengenal istilah ini melalui uang-uang digital seperti *Bitcoin, Ethereum*, dan lain-lain. Mata uang digital ini biasanya juga disebut dengan istilah *cryptocurrency*. Tetapi apa sebenarnya yang dimaksud dari *blockchain?* Apa dampak dari hadirnya *blockchain* ini dalam perbankan dan bisnis di era digital saat ini?

Blockchain adalah sebuah metode baru yang berfungsi untuk menyimpan informasi transaksi antar pelaku transaksi tersebut. Seperti metode penyimpanan transaksi pada umumnya, *blockchain* menyimpan data-data seperti alamat pengirim, alamat tujuan pengiriman, jumlah yang dikirim, dan data-data lainnya yang berkaitan tentang transaksi tersebut. Untuk setiap data-data yang tersimpan dalam masing-masing transaksi disimpan dan diamankan dengan menggunakan sistem *hash*. Untuk setiap transaksi tentunya akan menghasilkan nilai *hash* yang berbeda-beda. Sistem *hash* yang digunakan dalam sistem *blockchain* ini dikenal dengan SHA. SHA sendiri merupakan singkatan dari *Secure Hash Algorithms.* Untuk lebih jelasnya, SHA akan dijelaskan dalam subbab II.

Hash-hash ini kemudian akan disimpan dalam sebuah struktur yang dinamakan dengan blok. Blok-blok inilah yang membedakan *blockchain* dari sistem penyimpanan transaksi yang kita kenal selama ini. Untuk menyatakan setiap blok adalah valid, diperlukan sebuah kode yang disebut dengan *Proof of Work*. Kode ini adalah kombinasi SHA dari setiap *hash* yang terbentuk pada transaksi-transaksi yang telah dicatat sebelumnya.

Setiap blok-blok yang terbentuk menyimpan kode *hash* dari blok sebelumnya. Inilah yang membuat sistem *blockchain* lebih unggul dari segi keamanan. Jika ada manipulasi data pada transaksi sebelumnya, kode hash blok tersebut akan berubah dan akan dinyatakan *invalid* pada blok berikutnya karena tidak sesuai dengan kode hash yang telah disimpan sebelumnya.



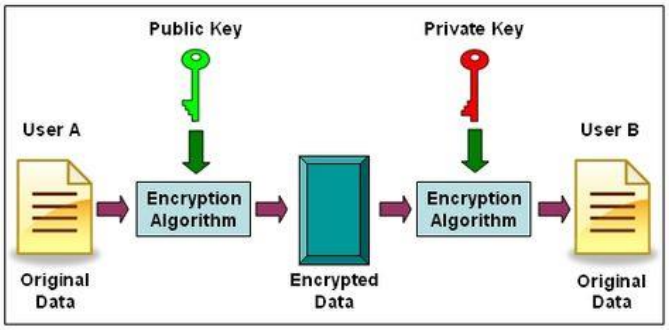
Gambar 1 *Blockchain*
Sumber: Quora

Untuk keperluan verifikasi data, setiap blok juga mengandung *merkle root* dari setiap hash yang terdapat pada blok tersebut. Untuk penjelasan lebih lanjut mengenai *merkle root* akan dipaparkan pada subbab III.

Sistem *blockchain* sebenarnya mirip dengan sistem *ledger* yang dipakai saat ini. Hanya saja, tidak seperti *ledger* pada umumnya yang dipegang dan dapat diisi oleh satu pihak sebagai pengawas transaksi, *ledger* pada *blockchain* disimpan dan dipegang oleh pengguna internet yang dikenal dengan istilah

miner (Sistem Desentralisasi). Setiap *miner* bertugas untuk menemukan kode hash yang sesuai supaya transaksi dapat dinilai valid.

Setiap pelaku transaksi yang menggunakan sistem ini akan mendapatkan sebuah kunci yang dikenal dengan istilah *Private Key* atau *Secret Key*. Selain itu terdapat pula kunci yang disebut sebagai *Public Key*. Untuk menjelaskan sistem keamanan ini, analogi yang ditemukan paling cocok adalah sebagai berikut.



Gambar 2 *Private Key* dan *Public Key*
Sumber: http://itlaw.wikia.com/wiki/Key_pair

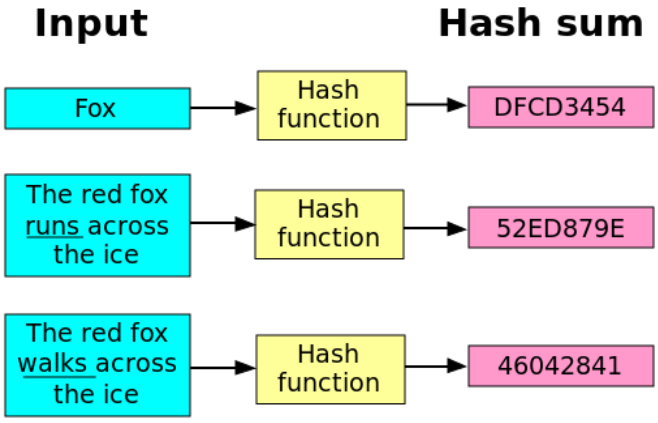
Jika seseorang membuka akun uang digital, maka ia akan mendapatkan sebuah box transparan dan kunci yang hanya bisa membuka box orang tersebut. Semakin banyak orang yang membuka akun uang digital tersebut, semakin banyak pula box dan kunci yang beredar. Kemudian, setiap box tersebut diberi kode nomor pada lapisan luarnya. Jika orang lain ingin mengirimkan uang pada kita, kita dapat memberitahu orang tersebut nomor yang terdapat pada box yang kita miliki. Pengirim kemudian dapat mengenali box tujuannya melalui kode nomor yang diberikan penerima tadi. Pengirim dapat menaruh uang pada box tersebut dan dapat melihat berapa banyak uang yang terdapat dalam box tersebut, tetapi tidak bisa mengambilnya karena box itu terkunci. Disinilah peran kunci tersebut. Hanya pemilik box itulah yang dapat membukanya.

Nomor yang tertera pada lapisan luar box itulah yang dinamakan dengan *public key* yang berupa alamat. Setiap orang yang diberikan kunci tersebut dapat mengakses akun seseorang hanya untuk mengirimkan uang dan mengetahui jumlah uang dalam akun tersebut, tetapi tidak dapat mengambilnya. Untuk dapat mengambil uang dari akun tersebut, diperlukan kunci untuk mendapat akses yang berupa kode hash sebagai alat dekripsi.

II. *SECURE HASH ALGORITHM*

Secure Hash Algorithm adalah sebuah metode untuk merubah sebuah string menjadi kode dengan ukuran panjang yang tetap. Metode ini ditemukan oleh *National Institute of Standards and Technology* (NIST), Amerika Serikat. SHA terdiri dari 4 macam jenis. Versi paling awal adalah SHA-0. Kode ini menghasilkan kode *hash* sepanjang 160 bit. SHA-0 ditemukan pada tahun 1993 dan pada akhirnya diperbaharui pada tahun 1995 karena terdapat beberapa kelemahan melalui SHA-1. Jika dibandingkan dengan metode SHA lainnya, metode ini adalah yang paling banyak digunakan dalam aplikasi yang banyak digunakan masyarakat secara umum. Selain aplikasi, SHA-1 ini juga digunakan dalam protokol-protokol keamanan seperti SSL, yang merupakan singkatan dari *Secure Socket Layer*. Lalu kemudian pada tahun 2005, ditemukan celah pada SHA-1 ini sehingga penggunaannya mulai diragukan. Lalu kemudian lahirlah SHA-2. Keluarga SHA-2 terdiri dari SHA-224, SHA-256, SHA-384, dan SHA-512. Keluaran *hash* ini tergantung dari jenisnya. Untuk tipe SHA-224, keluarannya adalah hash dengan panjang 224 bit, untuk tipe SHA-256 keluarannya adalah *hash* dengan panjang 256 bit, dan demikian seterusnya.

Gambar 3 *Hash*
Sumber: Quora



A. *Hashing*

SHA bersifat satu arah, yang artinya kita dapat mengubah bentuk sebuah string menjadi *hash* tetapi tidak sebaliknya. Oleh karena itu, untuk menebak sebuah *string* yang cocok dari sebuah kode *hash* tidaklah mudah. Dalam *cryptocurrency* semua data disimpan dalam bentuk string. Setiap string tersebut seperti yang telah dijelaskan di atas akan menghasilkan kode *hash* yang unik jika di-enkripsi menggunakan SHA-256.

Seperti yang telah dijelaskan di awal, setiap blok yang terbentuk dapat dinyatakan valid apabila sudah disertai dengan *proof of work*. *Proof of work* adalah sebuah *digital signature* berupa kode *hash*. Dalam Bitcoin, sebuah kode *Proof of Work* harus memiliki nilai lebih kecil dari 2^{240}. Oleh karena itu, setiap *string* yang terkandung dalam blok harus dimodifikasi sedemikian rupa agar dapat menghasilkan kode *hash* yang valid. Dalam Bitcoin, terdapat sebuah istilah yang dinamai *Nonce*. *Nonce* adalah sebuah ruang modifikasi yang diberikan pada setiap *string* yang terdapat pada blok. *Nonce* pada Bitcoin berisi kombinasi angka sebesar 32-*bit* (4-*byte)* yang terdiri dari 0 dan 1.

Miner dalam sistem yang diterapkan Bitcoin ini bertugas untuk menebak kombinasi string dan *Nonce* tersebut sehingga apabila dirubah dengan algoritma SHA-256 akan menghasilkan kode *hash* yang memiliki nilai lebih kecil dari 2^{240}.

0 / 1	0 / 1	0 / 1	0 / 1	0 / 1	0 / 1	0 / 1	0 / 1

Tabel diatas menggambarkan semua kemungkinan string biner jika panjang string tersebut adalah 8-bit. Pada setiap kolomnya, terdapat dua kemungkinan, yaitu antara 0 atau 1. Tabel diatas memiliki delapan kolom, yang berarti terdapat

delapan kolom yang masing-masing memiliki dua kemungkinan. Sehingga jika dibuat dalam tabel kemungkinannya menjadi

2	2	2	2	2	2	2	2

Jadi, perhitungan matematika untuk banyaknya kombinasi *string* yang mungkin menjadi 2^8 yang nilainya sama dengan 256 kemungkinan. Melalui perhitungan diatas, dapat ditarik kesimpulan bahwa dalam setiap string yang memiliki panjang n, memiliki kombinasi biner sebanyak 2^n.

Karena *Nonce* dalam Bitcoin memiliki panjang 32-*bit*, itu berarti tabel kemungkinan untuk *Nonce* memiliki 32 kolom dengan masing-masing kolom memiliki dua kemungkinan. Sehingga, untuk menemukan kombinasi string yang sesuai untuk mendapatkan *Proof Of Work* yang valid, memerlukan 2^{32} kali usaha menebak dalam kasus terburuk.

Komputer sekarang pada umumnya memiliki kemampuan untuk melakukan penebakan sebanyak 4 miliar kali per detik. Karena nilai 2^{32} mendekati 4 miliar, itu berarti komputer dapat menebak kombinasi yang sesuai dalam waktu satu detik.

Metode dalam *hashing* SHA-256 akan dijelaskan dalam *pseudocode* dibawah ini

```
Note 1: All variables are 32 bit unsigned integers and addition is calculated modulo 2^32
Note 2: For each round, there is one round constant k[i] and one entry in the message schedule array w[i], 0 ≤ i ≤ 63
Note 3: The compression function uses 8 working variables, a through h
Note 4: Big-endian convention is used when expressing the constants in this pseudocode, and when parsing message block data from bytes to words, for example, the first word of the input message "abc" after padding is 0x61626380
Initialize hash values: (first 32 bits of the fractional parts of the square roots of the first 8 primes 2..19):

h0 := 0x6a09e667
h1 := 0xbb67ae85
h2 := 0x3c6ef372
h3 := 0xa54ff53a
h4 := 0x510e527f
h5 := 0x9b05688c
h6 := 0x1f83d9ab
h7 := 0x5be0cd19

Initialize array of round constants: (first 32 bits of the fractional parts of the cube roots of the first 64 primes 2..311):
k[0..63] :=    0x428a2f98,    0x71374491,    0xb5c0fbcf,    0xe9b5dba5,    0x3956c25b,    0x59f111f1,    0x923f82a4,    0xab1c5ed5,
0xd807aa98,    0x12835b01,    0x243185be,    0x550c7dc3,    0x72be5d74,    0x80deb1fe,
0x9bdc06a7,    0xc19bf174,    0xe49b69c1,    0xefbe4786,    0x0fc19dc6,    0x240ca1cc,
0x2de92c6f,    0x4a7484aa,    0x5cb0a9dc,    0x76f988da,    0x983e5152,    0xa831c66d,
0xb00327c8,    0xbf597fc7,    0xc6e00bf3,    0xd5a79147,    0x06ca6351,    0x14292967,
0x27b70a85,    0x2e1b2138,    0x4d2c6dfc,    0x53380d13,    0x650a7354,    0x766a0abb,
0x81c2c92e,    0x92722c85,    0xa2bfe8a1,    0xa81a664b,    0xc24b8b70,    0xc76c51a3,
0xd192e819,    0xd6990624,    0xf40e3585,    0x106aa070,    0x19a4c116,    0x1e376c08,
0x2748774c,    0x34b0bcb5,    0x391c0cb3,    0x4ed8aa4a,    0x5b9cca4f,    0x682e6ff3,
0x748f82ee,    0x78a5636f,    0x84c87814,    0x8cc70208,    0x90befffa,    0xa4506ceb,
0xbef9a3f7,    0xc67178f2

Pre-processing (Padding):
begin with the original message of length L bits append a single '1' bit
append K '0' bits, where K is the minimum number >= 0 such that L + 1 + K + 64 is a multiple of 512
append L as a 64-bit big-endian integer, making the total post-processed length a multiple of 512 bits

Process the message in successive 512-bit chunks:
break message into 512-bit chunks
for each chunk
    create a 64-entry message schedule array w[0..63] of 32-bit words
    (The initial values in w[0..63] don't matter, so many implementations zero them here)
    copy chunk into first 16 words w[0..15] of the message schedule array

Extend the first 16 words into the remaining 48 words w[16..63] of the message schedule array:
for i from 16 to 63
    s0 := (w[i-15] rightrotate 7) xor (w[i-15] rightrotate 18) xor (w[i-15] rightshift 3)
    s1 := (w[i-2] rightrotate 17) xor (w[i-2] rightrotate 19) xor (w[i-2] rightshift 10)
    w[i] := w[i-16] + s0 + w[i-7] + s1

Initialize working variables to current hash value:
a := h0
b := h1
c := h2
d := h3
```

```
e := h4
f := h5
g := h6
h := h7

Compression function main loop:
for i from 0 to 63
   S1 := (e rightrotate 6) xor (e rightrotate 11) xor (e rightrotate 25)
   ch := (e and f) xor ((not e) and g)
   temp1 := h + S1 + ch + k[i] + w[i]
   S0 := (a rightrotate 2) xor (a rightrotate 13) xor (a rightrotate 22)
   maj := (a and b) xor (a and c) xor (b and c)
   temp2 := S0 + maj
   h := g
   g := f
   f := e
   e := d + temp1
   d := c
   c := b
   b := a
   a := temp1 + temp2

   Add the compressed chunk to the current hash value:
   h0 := h0 + a
   h1 := h1 + b
   h2 := h2 + c
   h3 := h3 + d
   h4 := h4 + e
   h5 := h5 + f
   h6 := h6 + g
   h7 := h7 + h

   Produce the final hash value (big-endian):
   digest := hash := h0 append h1 append h2 append h3 append h4 append h5 append h6 append h7
```

Pada awal algoritma ini, terdapat 8 kode *hash* sebagai inisialisasi yang dinamakan h0, h1, sampai h7. Lalu program membuat sebuah tabel yang berisikan 32-*bit* pertama dari bagian pecahan 64 bilangan prima pertama. Kemudian, program mengatur panjang *string* masukan agar panjangnya merupakan kelipatan 512. Lalu program memecah *string* menjadi beberapa bagian yang per bagiannya 512 *bit*. Setelah itu program membuat tabel yang berisikan 64 elemen dengan masing-masing elemennya berisi 32-*bit string*. Kemudian program menyalin 16-*bit* pertama dari *string* tersebut ke elemen tabel yang telah dibentuk sebelumnya. 48-*bit* yang tersisa kemudian digantikan dengan *string* dari hasil algoritma yang merupakan gabungan dari algoritma *rotate* dan xor. Program lalu membuat 8 variabel sebagai tempat untuk menampung nilai 8 kode *hash* yang telah dinisialisasi di awal program. Dalam kasus ini, variable tersebut diberi nama a, b, sampai h. Kemudian program melakukan pengacakan kembali pada nilai 8 variabel diatas dengan algoritma yang berisikan *rotate* dan xor kembali. Nilai h0 sampai h7 kemudian digantikan dengan hasil penjumlahan hash tersebut dengan a sampai h. Hasil *hash* akhirnya merupakan hasil konkatenasi antara h0 sampai h7 yang telah dimodifikasi seperti yang telah disebutkan di atas.

B. Sistem *Proof-of-Work* dan *Proof-of-State*

Dalam sistem *Proof Of Work*, para *Miner* dari seluruh penjuru dunia saling berlomba untuk menebak kombinasi yang sesuai tersebut. Biasanya, *Miner* mengandalkan *Graphics Processing Unit* atau yang biasa disingkat menjadi GPU dalam komputer mereka. Hal ini dikarenakan GPU merupakan komponen komputer yang terdiri dari banyak *processor* kecil, sehingga semakin banyak pula perhitungan yang dapat dilakukan komputer tersebut dalam satu waktu. Kemudian, *Miner* yang berhasil mendapatkan kombinasi tersebut mendapat *reward* berupa *cryptocurrency*.

Sistem *Proof-of-Work* ini tidak jarang disamakan dengan sistem *Proof-of-Stake* (POS). Padahal, kedua sistem ini memiliki banyak perbedaan. Pada sistem *Proof-of-Stake*, *Miner* merupakan orang yang ditunjuk oleh sistem. Kemungkinan ditunjuknya seseorang dicerminkan dari berapa banyak *cryptocurrency* yang ia miliki, tidak seperti sistem *Proof-of-Work* dimana setiap *Miner* saling berlomba-lomba. Perbedaan yang kedua terdapat dalam sistem *rewarding*. Berbeda dengan sistem *Proof-of-Work*, dalam sistem POS, *Miner* mengambil keuntungan melalui *transaction fee* dari transaksi yang berhasil diselesaikan. Sistem POS memiliki keefektifan yang lebih tinggi jika dibandingkan dengan sistem POW, karena dalam sistem POS, hanya satu *Miner* yang bekerja dalam satu transaksi, sehingga tidak banyak tenaga komputer yang terbuang.

III. MERKLE TREE

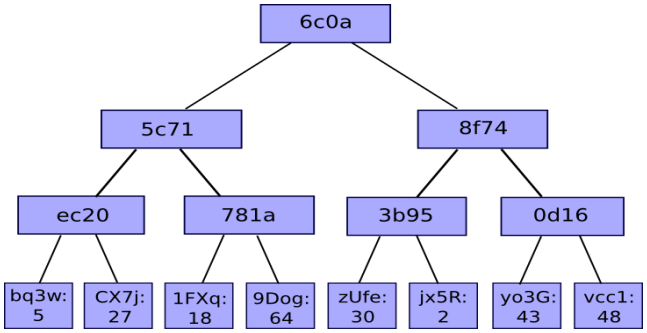
Pohon Merkle adalah sebuah sistem penyimpanan data pada dunia komputer, dimana setiap daunnya merupakan *hash*, dan setiap anggota yang bukan daun merupakan *hash* dari semua *child node* yang dimilikinya. Biasanya, pohon Merkle memiliki dua *child node* pada setiap anggota yang bukan merupakan daun. Metode ini pertama kali ditemukan oleh

Pohon Merkle efektif digunakan untuk sistem terdistribusi dalam verifikasi data. Hal ini dikarenakan sistem ini tidak perlu menyimpan semua data secara mentah, melainkan cukup dengan menyimpan *hash* dari setiap data tersebut dalam daun-daunnya. Sistem ini biasa digunakan dalam jaringan komputer yang memakai sistem *peer-to-peer networking*. Contoh jaringan yang memakai sistem ini adalah Git, Tor, dan Cryptocurrency seperti Bitcoin, Ethereum, dan lain-lain.

A. Pembuatan Pohon Merkle

Pembuatan *Merkle Tree* dalam *blockchain* sangatlah sederhana. Misalkan transaksi “a” menyatakan bahwa Angel membayar Toni sebesar 2 Bitcoin. Transaksi tersebut akan disimpan dalam bentuk *hash* seperti yang telah dijelaskan pada bab II. Kemudian hasil dari *hashing* tersebut akan menjadi daun paling kiri dari pohon Merkle yang akan dibuat. Lalu kemudian terdapat tiga transaksi berikutnya yang sebut saja transaksi “b”, “c”, dan “d”. Ketiga transaksi tersebut juga akan disimpan

dalam bentuk *hash* seperti yang terjadi pada transaksi pertama sehingga terbentuk daun kedua, ketiga, dan keempat. Lalu kemudian sistem secara otomatis melakukan algoritma *hashing* pada daun pertama dan daun kedua sehingga terbentuk pohon. *Hashing* juga dilakukan pada daun ketiga dan keempat. Kemudian karena sekarang terbentuk dua pohon, maka sistem akan kembali melakukan algoritma *hashing* pada kode hash yang telah terbentuk dari dua transaksi pertama dan dua transaksi kedua. Proses ini terus berulang sampai ukuran pohon mencapai ukuran yang diinginkan. Dalam sistem *blockchain* Bitcoin, setiap blok rata-rata berukuran 1 MB atau sekitar 200 transaksi.

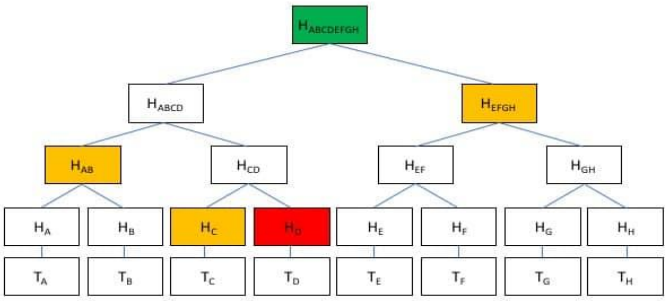


Gambar 4 *Merkle Hash Tree*
Sumber: Ethereum.org

B. Verifikasi

Jika dalam sistem *ledger* pengecekan dilakukan dengan mencari dokumen yang mengandung data mengenai transaksi tersebut, tidak demikian dengan sistem verifikasi pada *blockchain*. Setiap blok yang terbentuk mengandung kode *merkle root* pada bagian header blok tersebut. Oleh karena itu, untuk memeriksa apakah sebuah salinan blok mengandung isi yang sama, kita dapat memanfaatkan *merkle root* tersebut.

Jika terdapat perbedaan pada salah satu daun saja, maka nilai *Merkle* Root tentu akan berbeda. Sehingga jika sistem menemukan perbedaan pada *merkle root* dua buah blok, maka prosedur verifikasi harus dijalankan. Sebagai langkah awal, komputer akan mencari blok yang sudah dipercaya sebagai bahan perbandingan. Kemudian komputer akan menelusuri semua *subhash* dari kedua *merkle root* blok tersebut. Jika *subhash* kedua blok bernilai sama, program tidak akan menelusuri *child node* dari *subhash* tersebut. Tetapi apabila terdapat perbedaan kembali, maka program akan menelusuri *child node* dari kedua *subhash* yang berbeda tersebut untuk ditelusuri lebih lanjut. Algoritma ini terus berulang sampai penelusuran mencapai daun yang tidak memiliki *subhash* (kedalaman terdalam dari sebuah tree).



Gambar 5 Verifikasi pohon
Sumber: Coincentral

Gambar di atas menunjukkan proses verifikasi seperti yang telah dijelaskan sebelumnya. Dalam pohon ini, H_d merupakan daun yang dinyatakan tidak valid karena berbeda dengan blok-blok lainnya yang dinilai sudah terpercaya. Gambar ini juga menunjukkan keefektifan dari algoritma pencarian dalam *Merkle Tree* ini. Dalam kasus ini, program hanya memeriksa *hash-hash* yang diwarnai kuning dan merah saja, sehingga program tidak perlu melakukan pemeriksaan pada seluruh daun.

IV. KOMPLEKSITAS

Dalam ilmu matematika, terdapat suatu metode untuk menentukan algoritma yang paling cocok untuk menyelesaikan sebuah masalah tertentu. Terdapat dua jenis kompleksitas, yaitu kompleksitas waktu dan kompleksitas ruang. Kompleksitas waktu menyatakan banyaknya waktu yang dibutuhkan untuk dapat menyelesaikan sebuah masalah, sedangkan kompleksitas ruang menyatakan banyaknya ukuran memori yang dibutuhkan untuk dapat memuat seluruh instruksi program.

Dalam kompleksitas waktu, dikenal sebuah istilah yang diberi nama *big-O*. Istilah ini digunakan untuk mengelompokkan berbagai jenis kompleksitas ruang dari berbagai macam program. Contoh *big-O* adalah sebagai berikut

O(1)	Waktu pelaksanaan tidak bergantung pada banyaknya data
O(n)	Laju bertambahnya waktu pelaksanaan sebanding dengan perbedaan banyaknya data
O(log n)	Laju bertambahnya waktu pelaksanaan lebih lambat dibanding dengan perbedaan banyaknya data
O(n log n)	Program memecah persoalan menjadi persoalan-persoalan kecil kemudian menyelesaikannya. Kemudian program menggabungkan hasilnya di akhir.
$O(n^2)$	Waktu bertambah secara kuadratik seiring dengan bertambah banyaknya data
$O(n^3)$	Waktu bertambah dengan laju pangkat tiga seiring dengan bertambah banyaknya data
O(n!)	Waktu meningkat sebanyak n+1 kali jika

	banyaknya data dirubah menjadi n+1
$O(2^n)$	Waktu meningkat secara eksponensial seiring dengan bertambah banyaknya data

Algoritma *Merkle Tree* merupakan algoritma yang cocok digunakan dalam sebuah sistem yang mengolah banyak data sekaligus seperti halnya sistem *Blockchain*. Hal ini dikarenakan program dirancang agar tidak perlu melakukan perbandingan pada semua daun yang dimiliki oleh pohon tersebut, seperti yang sudah dijelaskan pada bab sebelumnya. Algoritma ini bekerja dengan prinsip yang hampir sama dengan algoritma *Binary Search*. Berikut ini adalah kompleksitas waktu algoritma *Merkle Tree*

	Kasus rata-rata	Kasus terburuk
Insert	$O(\log_2(n))$	$O(\log_2(n))$
Search	$O(\log_2(n))$	$O(\log_k(n))$
Delete	$O(\log_2(n))$	$O(\log_k(n))$
Synchronization	$O(\log_2(n))$	$O(n)$

Dalam kasus rata-rata *insert*, *search*, dan *delete*, program memiliki kompleksitas $O(\log_2(n))$ yang berarti laju pertumbuhan waktu lebih lambat dibandingkan dengan laju pertumbuhan data. Pada kasus terburuk algoritma *search* dan *delete*, kompleksitas waktu yang dimiliki adalah sebesar $O(\log_k(n))$, dimana k adalah banyaknya *child node* yang dimiliki oleh masing-masing komponen non-daun dalam pohon. Tetapi karena pada *Blockchain*, masing-masing komponen non-daun memiliki dua *child node*, maka nilai k disini adalah dua. Dalam algoritma sinkronisasi, kompeksitas waktu dalam kasus terburuk adalah O(n). Hal ini terjadi pada kasus dimana setiap daun memiliki nilai yang berbeda sehingga jika banyaknya data adalah n, jumlah perbandingan yang dilakukan oleh program tersebut adalah sebanyak n data.

V. PENGARUH DAN MANFAAT

Sistem ini merupakan salah satu terobosan paling berpengaruh dalam sistem ekonomi dunia. Bill Gates pun pernah berkata, "*BitCoin is a technological tour de force*", yang berarti Bitcoin merupakan sebuah pencapaian dalam dunia teknologi yang luar biasa. Banyak pihak pun mulai melirik *Cryptocurrency* sebagai tempat untuk berinvestasi. Hal ini dapat dibuktikan dengan sempat melonjaknya harga Bitcoin pada tahun lalu, dimana pada awal januari satu Bitcoin dihargai sekitar $1000 dan melonjak pada awal kuartal ketiga. Harga Bitcoin per satu keping pada saat itu mencapai $2500. Tidak berhenti sampai disitu, performa Bitcoin terus meningkat sampai mencapai puncaknya pada akhir tahun dimana nilai satu Bitcoin dihargai $20,000. *Cryptocurrency* pun sampai sekarang masih digunakan sebagai alat investasi.

Gambar 6 Performa Bitcoin 2017
Sumber: *Business Insider*

Manfaat lain dari *cryptocurrency* ini adalah transaksi dapat dilakukan secara *anonymous*. Ini disebabkan karena semua data transaksi disimpan dalam bentuk *hash*, sehingga identitas pengirim, dan alamat tujuan pengiriman tidak dapat diketahui secara langsung. Transaksi *cryptocurrency* juga bersifat *peer-to-peer* yang artinya pengiriman uang dilakukan secara langsung tanpa melalui perantara. Ini menyebabkan transaksi dapat dilakukan dengan cepat.

Selain itu, kelebihan lain yang ditawarkan mata uang digital ini adalah terbukanya peluang untuk melakukan transaksi antar negara yang memiliki mata uang yang berbeda. Transaksi dapat dilakukan tanpa perlu menukar mata uang terlebih dahulu, sehingga perdagangan dapat dilakukan dengan lebih mudah.

Mata uang konvensional pada umumnya bergantung pada politik dan berbagai kebijakan pemerintah pada negara yang memakai mata uang tersebut. Namun pada Bitcoin, mata uang ini tidak terikat pada pihak manapun, sehingga keadaan politik, ekonomi, dan sosial negara tidak berpengaruh pada kinerja Bitcoin. Harga murni dipengaruhi oleh perbandingan banyaknya *demand and supply* akan Bitcoin. Oleh karena itu, Bitcoin dan mata uang digital lainnya dapat digunakan sebagai sarana pelindung dari inflasi negara layaknya emas.

Akan tetapi, tidak semua negara dapat menerima Bitcoin sebagai alat pembayaran yang sah. Oleh karena itu pihak Bitcoin selalu berusaha untuk membuat mata uangnya dapat lebih mudah diterima oleh semua pihak. Salah satu cara yang ditempuh adalah dengan menetapkan regulasi sehingga perdagangannya dapat dilakukan dengan legal. Sejauh ini, pihak Bitcoin telah mengajukan sembilan buah proposal pada ETF (*Exchange Traded Fund*). ETF adalah upaya pendanaan, yang secara singkat dapat membuat Bitcoin dapat diperdagangkan seperti layaknya saham dalam bursa efek. Sayangnya, Bitcoin masih harus terus berusaha memperbaiki sistem yang dimilikinya karena semua proposal yang diajukan tersebut belum ada yang diterima. Menurut pakar ekonomi, alasan terkuat ditolaknya proposal ini adalah adanya keraguan dan ketakutan akan terjadinya manipulasi pasar dan penipuan di ETF. Resiko yang terkandung dalam Bitcoin dinilai masih terlalu besar.

Kelemahan lain yang dimiliki oleh uang ini adalah karena sifatnya yang *anonymous*, transaksi ini tidak dapat dilacak secara langsung. Uang ini sering kali dimanfaatkan sebagai alat tukar dalam perdagangan gelap di internet. Selain itu, tidak

jarang juga ditemukan kejahatan lain seperti judi. Pada Juli 2018 lalu, ketika sedang digelarnya piala dunia, kepolisian Guangdong selatan, Republik Rakyat Cina, menemukan judi sepakbola yang melibatkan 330,000 pengguna dan sekitar 8,000 agen. Jumlah perputaran uang dalam kasus kali ini diyakini mencapai 1.5 juta dolar amerika.

VI. KESIMPULAN

Bitcoin dan mata uang digital lainnya menawarkan sistem baru dalam penyimpanan catatan transaksi. Sistem desentralisasi yang diterapkan dalam *cryptocurrency* ini menawarkan banyak keuntungan. Salah satu fitur yang selalu menjadi unggulannya adalah fitur keamanan. Melalui sistem *Blockchain* dan enkripsi SHA-256, dapat dikatakan hampir tidak ada celah yang dapat dimanfaatkan untuk meretas sistem ini. Semua orang yang memiliki akses internet dapat mengontrol dan mengawasi jumlah uang yang beredar dan kemana saja uang-uang itu mengalir.

Jika dilihat dari kelebihan-kelebihan yang ditawarkan, mata uang digital dapat menciptakan pola pikir baru mengenai sistem pencatatan transaksi yang baik. Selain itu, *cryptocurrency* ini juga memulai sebuah era dimana kepemilikan aset digital mulai dipertimbangkan oleh orang-orang sebagai alat tukar, maupun sebagai alat investasi. Terlepas dari bagaimana reaksi pasar akan munculnya fenomena baru ini, inovasi dalam *Blockchain* ini tidak dapat dipandang sebelah mata. Uang ini dianggap dapat menggantikan mata uang konvensional pada masa yang akan datang.

VII. UCAPAN TERIMA KASIH

Saya ingin berterima kasih kepada Tuhan yang Maha Esa atas karunia-Nya yang telah mengizinkan saya untuk menyelesaikan makalah ini. Saya juga ingin berterima kasih kepada Dr. Judhi Santoso, Dr. Rinaldi Munir, dan Harilili M.Sc sebagai dosen yang telah memberi pengarahan dalam menyelesaikan makalah Matematika Diskrit ini. Terima kasih juga saya ucapkan pada teman-teman saya yang telah mendukung saya dalam menyelesaikan makalah ini.

REFERENCES

[1] https://www.tutorialspoint.com/cryptography/cryptography_hash_functions.htm diakses pada tanggal 30 November 2018 pukul 13.58

[2] https://www.youtube.com/watch?v=bBC-nXj3Ng4 diakses pada tanggal 30 November 2018 pukul 11.25

[3] https://lifehacker.com/what-is-blockchain-1822094625 diakses pada tanggal 30 November 2018 pukul 15.50

[4] https://en.bitcoin.it/ diakses pada tanggal 7 Desember 2018 pada pukul 13:45

[5] https://brilliant.org/wiki/merkle-tree/ diakses pada tanggal 7 Desember 2018 pada pukul 16:50

[6] https://coincentral.com/merkle-tree-hashing-blockchain/ diakses pada tanggal 7 Desember 2018 pada pukul 18:25

[7] https://www.cnnindonesia.com/teknologi/20170914121558-185-241681/menakar-masa-depan-bitcoin-dan-mata-uang-digital-di-indonesia diakses pada tanggal 7 Desember 2018 pada pukul 22.40

[8] https://tools.ietf.org/html/rfc6234 diakses pada tanggal 9 Desember 2018 pada pukul 17.03

[9] https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf diakses pada tanggal 9 Desember 2018 17.05

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, 8 Desember 2018

Timothy 13517087