Pemanfaatan Algoritma Brute Force dalam Penambangan *Bitcoin*

Michael Owen - 13519055
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail (gmail): 13519055@std.stei.itb.ac.id

***Abstract*—Di masa pandemi ini *cryptocurrency* semakin dikenal masyarakat dan menjadi alternatif investasi yang menggantikan saham, emas serta investasi lainnya. Trend cryptocurrency ini juga dipicu oleh beberapa alasan seperti kenaikan harga bitcoin yang merupakan salah satu cryptocurrency awal dan saat ini memiliki volume perdagangan terbesar yang mencapai 15 hingga 16 kali lipat dibandingkan harganya pada pertengahan tahun 2020. Penyebab lainnya antara lain karena investasi yang dilakukan secara besar-besaran oleh tokoh penting seperti Elon Musk dan cuitannya di twitter tentang Doge coin yang berhasil membuat harganya meningkat hingga 16 kali lipat dibandingkan tahun lalu.**

Ada banyak cara untuk medapatkan bitcoin ini salah satunya adalah dengan melakukan "penambangan", makalah ini akan membahas bagaimana cara melakukan penambangan cryptocurrency menggunakan algoritma brute force dan kenapa algoritma brute force yang dipilih bukan algoritma lain.

Keywords—SHA-256; Brute Force; Cryptocurrency; Hash

I. Pendahuluan

Cryptocurrency berasal dari kata cryptography dan currency. Currency berarti mata uang sedangkan cryptography berarti tulisan yang tersembunyi. Cryptocurrency adalah mata uang digital yang diperoleh melalui proses kriptografi atau penyembunyian data. Cryptocurrency terdesentralisasi pertama adalah bitcoin yang ditemukan oleh Satoshi Nakamoto yang kemudian diikuti oleh kemunculan cryptocurrency lainnya.

Pada awalnya harga bitcoin kurang dari 1 USD dan terhitung pada bulan April 2021 harganya mencapai 64.000 hingga 65.000 USD. bayangkan anda membeli bitcoin di harga 1 dollar pada tahun 2010 atau menambangnya dan menyimpannya hingga saat ini jika anda membeli sebanyak 1 dollar saja bisa dibayangkan kekayaan anda dalam 11 tahun yang mencapai 64 sampai 65 ribu kali lipat atau anda terus menambangnya sejak tahun 2010 hingga saat ini mungkin anda sudah menjadi seorang miliuner.

Untuk mendapatkan reward bitcoin dari proses mining, para penambang harus menebak angka yang jika dikombinasikan dengan data didalam block akan menghasilkan sebuah hash yang berhasil melewati hash function yang telah ditentukan. Hash yang dimaksud biasanya dimulai dengan beberapa angka 0. Angka yang harus ditebak disebut sebagai "nonce" (number used once) yang berada direntang 0 sampai 4,294,967,296.

karena menggunakan prinsip Proof of Work maka tingkat kesulitan (jumlah 0 yang diperlukan diawal sebuah hash string, semakin banyak 0 semakin sulit penambangan) dari kalkulasi di sesuaikan berdasarkan hashrate saat penambangan dilakukan, semakin banyak komputer yang digunakan untuk menambang maka semakin sulit juga untuk melakukan kalkulasinya sehingga diperkirankan untuk menyelesaikan verifikasi sebuah blok diperlukan waktu sekitar 10 menit.

II. Dasar Teori

A. *Algoritme Brute Force*

Algoritme brute force adalah algoritme yang dapat menghasilkan solusi yang selalu optimal terhadap suatu permasalahan dengan cara mencoba semua kemungkinan solusi yang ada. Algoritma ini selalu dijadikan dasar untuk mendapatkan solusi optimal meskipun tidak semangkus algoritme lain. Hampir semua permasalahan dapat diselesaikan menggunakan algoritme brute force termasuk dalam menemukan nonce untuk melakukan validasi terhadap sebuah block dalam blockchain.

Algoritme brute force sering disebut sebagai algoritme naif karena mengecek semua solusi yang ada untuk memilih solusi optimal dan memiliki waktu yang sebanding dengan langkah atau pengecekan solusi yang dilakukan.

Contoh Algoritme Brute Force dalam algoritma searching normal dalam bahasa python

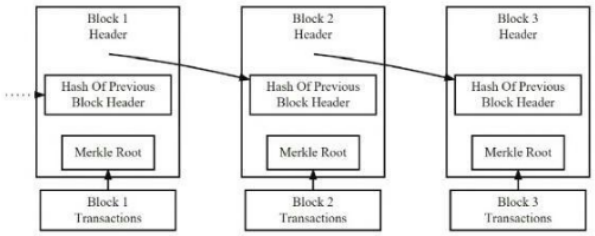
```
def search(x,value): #x is array of valueType
    for i in range (len(x)):
        if (x[i] == value):
            printf(i)
    printf("value tidak ditemukan")
```

dari contoh tersebut bisa diketahui bahwa semakin banyak isi dari array x dan semakin besar index posisi value didalam array maka waktu komputasinya juga akan semakin lama.

sudah jadi sifat dari algoritme brute force ini untuk mencari solusi dari suatu permasalahan dengan cara yang tidak mangkus karena mengecek semua kemungkinan, namun ada beberapa persoalan yang hanya bisa diselesaikan dengan algoritme ini seperti brute force attack untuk mendapat nonce tepat dalam memvalidasi sebuah block dengan tujuan mendapat reward bitcoin nantinya.

B. Bitcoin

Bitcoin adalah mata uang kripto terdesentralisasi pertama yang ditemukan oleh Satoshi Nakamoto (hanya nama samaran yang sampai saat ini belum diketahui keberadaannya), Bitcoin menyimpan semua informasi transaksinya didalam blockchain yang terdiri atas blok blok yang saling terkait seperti rantai dan memiliki nomor berurutan. Blok-blok dalam blockchain dapat terhubung karena nilai hash dari sebuah blok akan dimasukkan kedalam proses pembuatan blok selanjutnya. karena hal tersebut setiap blok dalam blockchain saling terhubung dan blok pertama adalah blok spesial yang tidak menyimpan previous hash block header dan disebut sebagai genesis block



gambar 1 struktur blockchain (Harding, 2015)

Struktur dari blockchain ini sendiri berbeda dibandingkan basis data pada umumnya, blockchain tidak memiliki fitur untuk mengedit data didalamnya karena semua informasi yang disimpan kedalam bitcoin merupakan informasi yang bersifat final dan tidak akan diubah lagi. Untuk menambahkan informasi baru ke dalam bitcoin diperlukan sebuah blok baru yang memenuhi kriteria tertentu.

Blok-blok yang diperlukan dalam sistem ini ditambang oleh para miner dalam proses menambang menggunakan komputer atau alat mining mereka

Bitcoin menggunakan algoritme SHA-256 (Secure Hash Algorithm) untuk mengenkripsi data sehingga jumlah hash yang dihasilkan akan selalu menghasilkan 64 karakter atau memiliki panjang 256 bit. Algoritme SHA ini sendiri dirancang oleh The National Institute of Standards and Technology (NIST). SHA sendiri memiliki beberapa variasi namun SHA-256 merupakan algoritme hashing yang cukup mumpuni dan sulit ditembus atau dilakukan benturan hingga saat ini. Algoritme SHA merupakan algoritme hash satu arah yang memiliki beberapa variasi seperti SHA-0, SHA-1, SHA-224, SHA-512, dan lain-lain. SHA-256 memanfaatkan beberapa operasi didalam melakukan enkripsi seperti And, Xor, Rot, Add(mod 2^{32}), Or, Shr

- RotR(A, n) adalah fungsi circular right shift of n bits of the binary word A.
- ShR(A, n) adalah fungsi untuk menggeser binary word A ke kanan sebanyak n bit
- A||B melambangkan konkatenasi binary word A dan B.

Fungsi dan Konstanta:

1. $Ch(X, Y, Z) = (X \wedge Y) \oplus (\bar{X} \wedge Z)$,
2. $M_{aj}(X, Y, Z) = (X \wedge Y) \oplus (X \wedge Z) \oplus (Y \wedge Z)$,
3. $\Sigma_0(X) = RotR(X, 2) \oplus RotR(X, 13) \oplus RotR(X, 22)$,
4. $\Sigma_1(X) = RotR(X, 6) \oplus RotR(X, 11) \oplus RotR(X, 25)$,
5. $\sigma_0(X) = RotR(X, 7) \oplus RotR(X, 18) \oplus ShR(X, 3)$,
6. $\sigma_1(X) = RotR(X, 17) \oplus RotR(X, 19) \oplus ShR(X, 10)$

64 binary words K_i adalah 32 bit pertama dari bagian fractional dari akar kuadrat 64 bilangan prima seperti dibawah ini:

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

Mengenkripsi sebuah pesan melalui Algoritme SHA-256 akan dilakukan dalam beberapa tahapan seperti di bawah ini:

1. Message Padding

Message padding atau penyisipan angka 0 sebanyak k pada pesan yang telah diubah kedalam bentuk binary yang telah disisipkan sebuah angka 1 diakhir hingga panjang pesan awal (L) + 1 + k + 64 bernilai sama dengan kelipatan 512, lalu setelah menyisipkan angka 0 tambahkan panjang teks awal dalam bentuk binary 64 bit. Akhirnya pesan akan memiliki panjang 512 bit

2. Parsing
   Pesan yang telah disisipkan kemudian di bagi kedalam 16 buah word 32 bit: $W_1, W_2, W_3, \ldots, W_{16}$

3. Message expansion
   Pesan yang telah diparsing kemudian ditambah dari W_{17} hingga W_{64}menggunakan formula

   $W_i = \sigma_1(W_i-2) + W_i-7 + \sigma_0(W_i-15) + W_i-16,\ 17 \leq i \leq 64$

4. Message Compression
   setelah mendapatkan W1 hingga W64 dari proses Message expansion selanjutnya kita lakukan kompresi menjadi 256 bit dengan formula
   1. set 8 variabel dengan nilai awal 32 bit pertama dari bagian fractional akar kuadrat 8 bilangan prima pertama
      H (0) 1 = 0x6a09e667
      H (0) 2 = 0xbb67ae85
      H (0) 3 = 0x3c6ef372
      H (0) 4 = 0xa54ff53a
      H (0) 5 = 0x510e527f
      H (0) 6 = 0x9b05688c
      H (0) 7 = 0x1f83d9ab
      H (0) 8 = 0x5be0cd19
   2. blocks M_1 hingga M_n diproses satu persatu:
      For t = 1 to N :
      susun 64 blocks Wi dari M(t)
      set (a, b, c, d, e, f, g, h) = (H (t−1) 1 , H(t−1) 2 , H(t−1) 3 , H(t−1) 4 , H(t−1) 5 , H(t−1) 6 , H(t−1) 7 , H(t−1) 8)
      do 64 rounds consisting of:
      $T_1 = h + \Sigma_1(e) + Ch(e, f, g) + K_i + W_i$
      $T_2 = \Sigma_0(a) + M_{aj}(a, b, c)$
      h = g
      g = f
      f = e
      $e = d + T_1$
      d = c
      c = b
      b = a
      $a = T_1 + T_2$
      hitung nilai baru dari $H_j(t)$
      $H_1 (t) = H_1 (t−1) + a$
      $H_2 (t) = H_2 (t−1) + b$
      $H_3 (t) = H_3 (t−1) + c$
      $H_4 (t) = H_4 (t−1) + d$
      $H_5 (t) = H_5 (t−1) + e$
      $H_6 (t) = H_6 (t−1) + f$
      $H_7 (t) = H_7 (t−1) + g$
      $H_8 (t) = H_8 (t−1) + h$

hash dari pesan adalah konkatenasi dari variabel H_i (n) setelah block terakhir diproses

$H = H_1 (N) \| H_2 (N) \| H_3 (N) \| H_4 (N) \| H_5 (N) \| H_6 (N) \| H_7 (N) \| H_8 (N)$

Dari proses hashing yang kompleks tersebut dapat dilihat bahwa akan ada banyak sekali kombinasi yang perlu dicoba untuk menghasilkan sebuah hash yang sama dan memungkinkan sebuah data sulit ditembus keamanannya. bahkan dengan perbedaan huruf kecil dan besar dampaknya akan sangat besar terhadap hash yang dihasilkan

III. Pemecahan masalah

Dalam proses penambangan diperlukan input angka nonce yang tepat untuk menghasilkan hash yang valid sehingga block dapat divalidasi dan kita bisa mendapatkan reward bitcoin. untuk itu diperlukan algoritme brute force attack untuk mencoba segala kemungkinan angka untuk mendapatkan hash yang valid.

Algoritme yang bisa dicoba yakni melakukan iterasi dari 0 hingga 4,294,967,296 jika sudah ditemukan angka yang tepat hentikan proses iterasi dan lanjutkan ke block berikutnya

contoh algoritme untuk pencarian nonce dalam bahasa python:

```
for i in range (4294967296):
    if (nonce(i)==hashValue):#nonce adalah fungsi yang memerlukan input untuk menghasilkan hash, jika nilainya sama hentikan proses pencarian
        print(i)
        endfor
```

semua kemungkinan angka harus diperiksa karena sedikit saja perbedaan angka akan menyebabkan hash mengalami perubahan yang signifikan. Karena hal tersebut algoritme hashing menggunakan brute force adalah algoritme yang tepat untuk digunakan dalam proses penambangan bitcoin ini.

References

[1] S. Joseph, “Pencarian transaksi terbaik pada bitcoin dengan program dinamis”

[2] W. Dimaz Ankaa, Mengenal Bitcoin dan Cryptocurrency

[3] Tammam. Aditya Gusti, dkk, Fungsi hash dan Algoritma SHA-256

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.

Lubuklinggau, 11 Mei 2021

Michael Owen 13519055