Aplikasi Elliptical Curve Digital Signature Algorithm pada Ethereum

Daniel Mario Reynaldi (13519031)
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13519031@std.stei.itb.ac.id

***Abstract*—Cryptocurrency adalah mata uang digital yang terdesentralisasi dan diamankan oleh kriptografi. Cryptocurrency pada umumnya menggunakan teknologi blockchain yang dapat mencegah terjadinya pemalsuan, penipuan dan double spending. Ethereum adalah cryptocurrency paling banyak digunakan kedua setelah Bitcoin. Transaksi pada Ethereum diamankan dengan tanda tangan digital, algoritma tanda tangan digital yang digunakan pada Ethereum adalah Elliptical Curve Digital Signature Algorithm yang menggunakan kombinasi konsep teori bilangan dan kriptografi dengan *public key*. ECDSA mustahil untuk dipecahkan dengan komputer modern asalkan syarat-syarat tertentu dipenuhi.**

***Keywords*—cryptocurrency, Ethereum, Elliptic Curve Digital Signature, kriptografi.**

I. Pendahuluan

1.1 Latar Belakang

Teori bilangan adalah cabang ilmu matematika yang mempelajari bilangan bulat dan sifat-sifatnya. Teori bilangan telah ada sejak zaman babilonia pada tahun 1800 SM, banyak matematikawan telah berkontribusi pada teori bilangan diantaranya adalah Pythagoras, Euclid, Fermat, Euler dan Gauss. Teori bilangan sangatlah penting bagi umat manusia, sangking pentingnya matematikawan Carl Friedrich Gauss menyatakan bahwa jika matematika adalah ratu ilmu pengetahuan maka teori bilangan adalah ratunya matematika.

Konsep-konsep teori bilangan mencakup operasi aritmatika pada bilangan bulat dan sifat-sifatnya, aritmatika modulo dan bilangan spesial seperti bilangan prima. Konsep-konsep tersebut banyak digunakan dalam kehidupan manusia sehari-hari, salah satu penggunaannya yaitu pada bidang komputasi. Teori bilangan menghasilkan kriptografi, random number generator, error correcting code dan masih banyak lagi. Aplikasi teori bilangan yang sangat banyak di dunia komputasi menyebabkan teori bilangan diajarkan pada materi mata kuliah matematika diskrit untuk hampir seluruh mahasiswa teknik informatika dan ilmu komputer di seluruh universitas di dunia.

Kriptografi atau kriptologi adalah ilmu yang bertujuan mempelajari cara mengkodekan sebuah pesan atau informasi dengan konsep matematika sehingga pesan atau kode hanya diterima dan dimengerti oleh pihak yang dituju dan bukan oleh pihak lain baik secara sengaja maupun tidak sengaja. Kriptografi merupakan perpotongan cabang ilmu matematika, ilmu komputer, fisika, dan teori bahasa. Kriptografi adalah bagian penting dari kehidupan manusia pada abad ke-21 terutama setelah revolusi komputer dan internet. Kriptografi digunakan untuk mengamankan data pengguna komputer dan internet, mengamankan data penting pemerintahan, blockchain dan *cryptocurrency*.

Digital signature algorithm adalah algoritma yang digunakan untuk menghasilkan tanda tangan digital, tanda tangan digital tersebut berfungsi untuk memverifikasi keaslian dokumen atau data. Tanda tangan digital digunakan pada transaksi finansial, distribusi perangkat lunak, dan mencegah pemalsuan dan penipuan. Salah satu jenis digital signature algorithms adalah *Elliptic Curve Digital Signature Algorithm* yang menggunakan elliptic curve untuk menghasilkan tanda tangan digital. ECDSA banyak digunakan pada teknologi blockchain terutama *cryptocurrency*.

Ethereum adalah *cryptocurrency* terbesar kedua setelah Bitcoin. Ethereum ditemukan pada tahun 2013 oleh seorang programmer asal Rusia, Vitalik Buterin. Ethereum resmi diluncurkan pada tanggal 30 Juli 2015.

II. Dasar Teori

2.1 Teori Bilangan

Teori bilangan adalah cabang ilmu matematika yang mempelajari bilangan bulat dan fungsi bernilai bilangan bulat. Bilangan bulat adalah bilangan yang tidak memiliki titik desimal.

1. Pembagian Bilangan Bulat
   a habis membagi b jika terdapat bilangan bulat c sedemikian sehingga $b = ac$.

   $$a \mid b \text{ jika } b = ac,\ c \in \mathbf{Z} \text{ dan } a \neq 0 \qquad (1.1)$$

2. Teorema Euclidean
   Misalkan m dan n adalah bilangan bulat, jika m dibagi dengan n maka hasil pembagiannya adalah q dan sisanya r sedemikian sehingga:

   $$m = n \cdot q + r \qquad (1.2)$$

3. Pembagian Bersama Terbesar (PBB)
   Misalkan a dan b adalah bilangan bulat bukan nol. Pembagi bersama terbesar dari a dan b adalah bilangan bulat terbesar c sehingga $c \mid a$ dan $c \mid b$. PBB dinyatakan dalam bentuk:

   $$PBB(a, b) = c \qquad (1.3)$$

4. Aritmatika Modulo
   Misalkan a dan m adalah bilangan bulat dengan $m > 0$, maka $a \bmod m$ memberikan sisa a saat dibagi m.

$$a = m \cdot q + r \quad (1.4)$$

$$a \bmod m = r \quad (1.5)$$

5. Kongruensi
   Misalkan a dan b adalah bilangan bulat dan m adalah bilangan bulat lebih besar dari nol, maka berlaku:

$$a \equiv b \,(mod\ m) \Leftrightarrow m \mid (a - b) \quad (1.6)$$

6. Balikan Modulo
   Misalkan a dan m bilangan bulat maka balikan dari $a \bmod m$ adalah :

$$x \cdot a \equiv 1 \,(mod\ m) \quad (1.7)$$

$$a^{-1} \,(mod\ m) = x \quad (1.8)$$

7. Bilangan Prima
   Bilangan prima p adalah bilangan yang hanya bisa dibagi dirinya sendiri dan bilangan 1.

2.2 Kriptografi

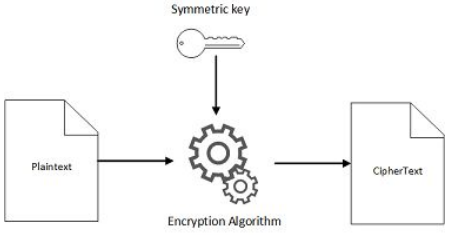
Kriptografi adalah ilmu yang mempelajari cara mengkodekan suatu data atau pesan agar terjaga kerahasiaannya, tidak dapat dimengerti pihak yang bukan pengirim atau penerima, dan tidak dapat dimanipulasi setelah dikirim.

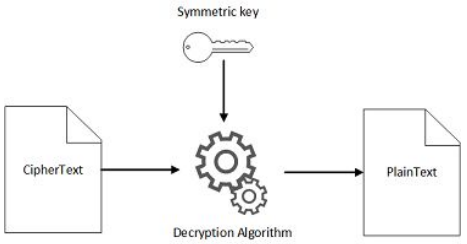
Pesan yang akan dikirim oleh pengirim (sender) disebut plaintext, melewati proses enkripsi plaintext akan diubah menjadi ciphertext yang tidak dapat dipahami oleh pihak yang tidak mengetahui algoritma enkripsi. Ciphertext kemudian dikirim ke penerima (receiver) lalu diubah kembali menjadi plaintext melalui proses dekripsi. Tidak semua algoritma enkripsi memungkinkan proses dekripsi salah satu contohnya adalah algoritma Hashing.

Kriptografi umumnya dibagi menjadi 3 jenis utama, diantaranya:

2.2.1 Kriptografi Simetris

Kriptografi simetris adalah protokol kriptografi yang menggunakan satu kunci untuk melakukan proses enkripsi dan dekripsi. Kunci yang digunakan dalam proses harus dijaga kerahasiaannya, dan biasanya adalah rangkaian bilangan acak yang dihasilkan oleh *Random Number Generator*.

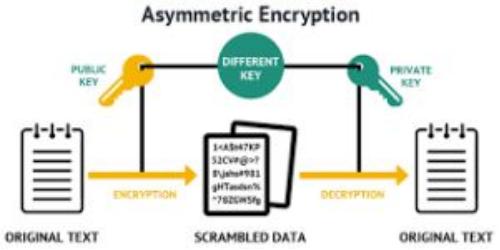




Gambar 1.1 Skema Kriptografi Simetris
(sumber:https://docs.aws.amazon.com/crypto/latest/userguide/concepts-algorithms.html)

2.2.2 Kriptografi Asimetris

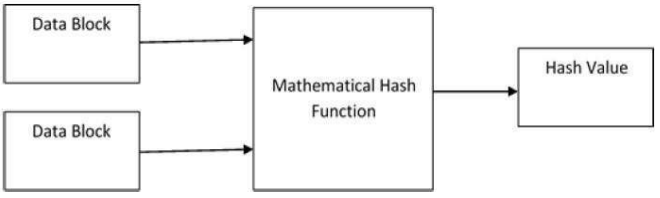
Kriptografi Asimetris adalah protokol kriptografi yang menggunakan dua kunci berbeda dalam proses enkripsi dan dekripsi. Kunci yang digunakan untuk proses enkripsi disebut kunci publik yang tidak perlu dijaga kerahasiaannya. Kunci yang digunakan untuk proses dekripsi disebut kunci privat, kerahasiaan kunci ini harus dijaga oleh pengirim dan penerima. Kunci privat umumnya dipilih secara random lalu diproses dengan operasi matematika untuk menghasilkan kunci publik. Operasi matematika yang digunakan harus *irreversible* seperti mencari faktor prima bilangan yang sangat besar, hal tersebut berguna untuk mencegah pesan diintervensi oleh pihak yang ketiga. Contoh algoritma kriptografi asimetris adalah RSA dan Elliptic Curve Cryptography.



Gambar 1.2 Skema Kriptografi Asimetris
(sumber: https://teachcomputerscience.com/asymmetric-encryption/)

2.2.3 Fungsi Hash

Fungsi hash adalah fungsi yang mengubah input berupa bilangan dengan panjang tertentu menjadi sebuah bilangan lain dengan panjang yang tetap. Fungsi hash adalah one-way encryption yang artinya pesan atau plaintext yang sudah di enkripsi tidak dapat lagi didekripsi. Contoh fungsi hash yang sering digunakan adalah fungsi MD5, SHA-1, SHA-2 dan SHA-256. Berikut adalah skema fungsi hash:

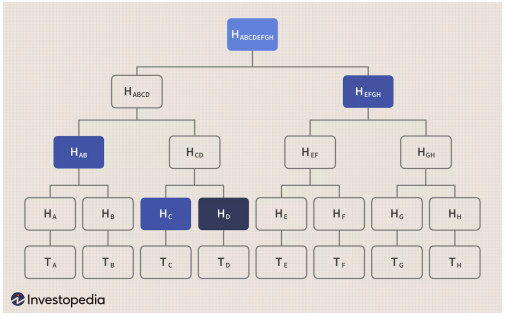


Gambar 1.3 Skema Hash Function
(sumber:www.tutorialspoint.com/cryptography/images/)

2.3 Merkle Tree

Merkle tree adalah suatu struktur data yang dipatenkan oleh

Ralph Merkle pada tahun 1979. Karakteristik Merkle tree adalah setiap daun dari pohon merkle berisi hash suatu data dan setiap simpul berisi hash dari node anaknya. Pohon merkle digunakan dalam cryptocurrency seperti Bitcoin dan Ethereum untuk mengorganisasi kumpulan transaksi dalam suatu block. Transaksi dalam suatu block menjadi daun dari pohon merkle lalu isi dari akar pohon tersebut dimasukan kedalam block, hal ini berguna untuk mempercepat validasi dan verifikasi transaksi pada setiap block. Selain itu, pohon merkle juga digunakan dalam Git, dan beberapa sistem NoSQL.



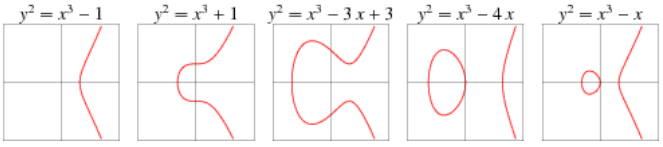
Gambar 1.4 Merkle Tree
(sumber: www.investopedia.com/terms/m
/merkle-root-cryptocurrency.asp)

2.4 Elliptic Curve Cryptography

Elliptic Curve Cryptography adalah kriptografi asimetris yang dikembangkan oleh Neal Koblitz dan Victor S. Miller pada tahun 1985. Dibandingkan algoritma seperti RSA, ECC membutuhkan lebih sedikit storage dan komputasi dengan tingkat keamanan yang hampir serupa. Kunci ECC 160 bit memiliki keamanan yang hampir sama dengan kunci RSA 1024 bit.

Elliptic curve adalah kurva yang didefinisikan oleh persamaan sebagai berikut:

$$y^2 = x^3 + ax + b \qquad (1.9)$$
$$4a^3 + 27b^2 \neq 0$$



Gambar 1.4 Elliptic Curve
(sumber: https://mathworld.wolfram.com
/EllipticCurve.html)

Elliptic Curve Cryptography didasari pada konsep *Elliptic Curve Discrete Logarithm Problem* atau disingkat ECDLP. ECDLP menyatakan bahwa untuk suatu kurva E dan untuk dua buah titik P dan Q pada kurva tersebut dan sebuah skalar k dimana berlaku:

$$kP = Q \qquad (1.10)$$

Q mudah dihitung jika diketahui k dan P, namun akan sangat sulit menghitung k dari Q dan P, terutama jika k adalah bilangan yang besar. Oleh karena itu, k adalah kunci privat pada ECC dan Q akan menjadi kunci publik.

ECC diimplementasikan pada Medan Galois untuk hasil yang optimum. Persamaan elliptic curve pada Medan Galois adalah seperti berikut:

$$y^2 = x^3 + ax + b \bmod p \qquad (1.11)$$
$$p \varepsilon P$$

Terdapat dua operasi elementer pada Medan Galois yang dapat dimanfaatkan untuk ECC yaitu operasi penjumlahan dua titik pada elliptic curve serta operasi penggandaan titik. Operasi penjumlahan dua titik pada elliptic curve dapat dinyatakan dalam persamaan:

$$P(x\square, y\square) \; dan \; Q(x_q, y_q) \qquad (1.12)$$
$$P + Q = R$$
$$x_r = m^2 - x\square - x_q \bmod p$$
$$y_r = m(x\square - x_r) - y_q \bmod p$$
$$m = \frac{y\square - y_q}{x\square - x_q} \bmod p$$

Operasi penggandaan titik pada elliptic curve dapat dinyatakan dalam persamaan sebagai berikut:

$$2P = R \qquad (1.13)$$
$$x_r = m^2 - 2x\square \bmod p$$
$$y_r = m(x\square - x_r) - y_p \bmod p$$
$$m = \frac{3x\square^2 + a}{2y\square} \bmod p$$

2.5 Digital Signature Algorithm

Digital Signature Algorithm adalah sekumpulan algoritma yang bertujuan menghasilkan tanda tangan digital dari sebuah data atau pesan. Tanda tangan digital adalah sebuah konsep kriptografi yang berguna menguji dan memverifikasi keaslian dan keamanan suatu pesan atau data.

Digital Signature Algorithm meliputi 4 langkah penting, yaitu:

1. Key Generation
   Pada tahap ini algoritma akan menghasilkan kunci publik dan kunci privat. Kunci privat x dipilih secara random berdasarkan parameter seperti fungsi hash yang dipilih, panjang kunci, dan lain-lain. Kunci publik y dihasilkan dari operasi aritmatika modulo terhadap kunci privat.

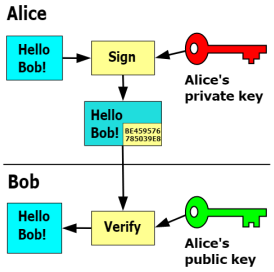
2. Key Distribution
   Kunci publik y dikirim oleh pengirim ke penerima, kunci publik tidak perlu dirahasiakan. Sebaliknya, kunci privat harus dirahasiakan.

3. Signing
   Pada tahap ini algoritma menghasilkan tanda tangan digital melalui tahap-tahap berikut:
   1. Pilih sebuah bilangan k random.
   2. $r \coloneqq (g^k \bmod p) \bmod q$
   3. $s \coloneqq (k^{-1}(Hash(m) + xr)) \bmod q$

   p dan q adalah bilangan prima yang dipilih pada tahap key generation, dan m adalah pesan yang hendak dikirim. Jika r atau s menghasilkan bilangan nol, maka proses diulang dengan k yang baru. Pasangan bilangan (r,s) adalah tanda tangan digital.

4. Signature Verification
   Pada tahap ini tanda tangan digital yang dihasilkan pada tahap *signing* diverifikasi dengan serangkaian

operasi aritmatika modulo. Jika tanda tangan melewati tahap verifikasi maka tanda tangan dapat dikirim ke penerima.



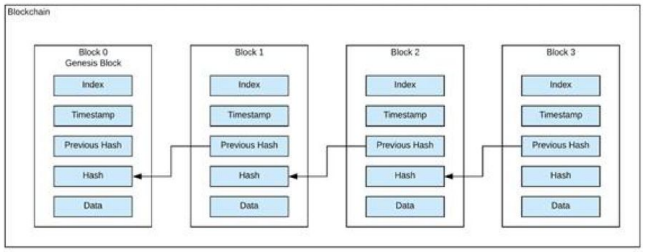
Gambar 1.5 Digital Signature Algorithm
(sumber:en.wikipedia.org/wiki/Digital_signature#/media/File:Private_key_signing.svg)

2.6 Blockchain dan Ethereum

Blockchain adalah serangkaian struktur data yang disebut block yang tiap waktu terus meningkat panjangnya. Block dihubungkan dengan block yang lain dengan kriptografi, tiap block mengandung hash dari block sebelumnya. Block berisi serangkaian transaksi yang dikumpulkan oleh node blockchain, jumlah transaksi pada suatu block bisa mencapai ratusan sampai ribuan transaksi. Block ini tidak langsung digabung ke rantai utama blockchain namun harus melalui suatu proses yang bernama *mining*.

Mining adalah proses yang berguna untuk memvalidasi dan memverifikasi tiap transaksi pada node. Mining membutuhkan komputasi yang tinggi, oleh karena itu sulit bagi pihak untuk merusak, merubah, atau memanipulasi rantai utama blockchain. Para penambang mendapat imbalan berupa Ether untuk setiap penambangan yang berhasil atau untuk setiap block yang ditambahkan ke rantai utama blockchain. Konsep dimana transaksi harus terlebih dahulu melalui proses mining disebut *proof-of-work*.

Transaksi dilakukan oleh sebuah akun, akun adalah sebuah entitas yang memiliki Ether dan dapat berupa milik user, *externally owned*, atau *smart contracts*. Setiap akun memiliki kunci privat dan kunci publik yang dihasilkan oleh Digital Signature Algorithm, kunci privat ini berguna untuk menandatangani setiap transaksi yang dilakukan oleh akun dan kunci publik berfungsi memverifikasi keaslian tanda tangan dan permintaan transaksi. Ethereum menggunakan skema ECDSA untuk menghasilkan kunci publik dari kunci privat.



Gambar 1.5 Blockchain
(sumber: https://www.spheregen.com/blockchain-technology-basics/)

III. Elliptic Curve Digital Signature Algorithm

Elliptic Curve Digital Signature Algorithm adalah salah satu jenis algoritma DSA yang memanfaatkan elliptic curve. ECDSA digunakan dengan memanfaatkan elliptic curve pada galois field. Terdapat 4 tahap penting pada DSA yaitu pembuatan kunci, pembuatan tanda tangan, pengiriman kunci dan verifikasi tanda tangan oleh penerima.

Pembuatan kunci pada ECDSA dimulai dengan pemilihan elliptic curve, elliptic curve pada galois field dinyatakan dengan persamaan (1.11), setelah itu kita dapat menghitung beberapa poin yang ada pada kurva. Dari sekumpulan titik pada kurva tersebut pilihlah sebuah titik G yang akan dijadikan basis kriptografi, dipilih juga bilangan n sedemikian sehingga n prima dan berlaku persamaan

$$nG = O \qquad (1.14)$$

O adalah titik kurva pada *infinity*. Setelah itu pilih juga fungsi hash yang akan digunakan untuk mengubah pesan menjadi bentuk hexadecimal biasanya fungsi yang dipilih adalah SHA-1, SHA-2, SHA-256 dan Keccak-256.

Pembuatan kunci dijelaskan dengan algoritma sebagai berikut:

1. Pilih secara acak sebuah bilangan prima d, $1 \leq d \leq n-1$.
2. Hitung $Q = dG$.
3. Q adalah kunci publik dan d adalah kunci privat yang dijaga kerahasiaannya.

Tahap selanjutnya adalah tahap pembuatan tanda tangan digital. Proses pembuatan tanda tangan digital dapat dijelaskan dengan algoritma berikut:

1. Pilih secara acak sebuah bilangan k, $1 \leq k \leq n-1$
2. Hitung $kG = R$, $R(x_r, y_r)$.
3. $r = x_r \bmod n$, jika $r = 0$ ulangi dari langkah 1.
4. r adalah satu dari dua bagian tanda tangan digital.
5. Hitung $k^{-1} \bmod n$.
6. Hitung e = Hash(m), m merupakan pesan yang akan ditanda tangani.
7. $s = k^{-1}(e + dr) \bmod n$.
8. Tanda tangan digital pada pesan m dengan kunci publik Q dan kunci privat d adalah pasangan (r,s).

Kunci dan tanda tangan kemudian dikirim ke penerima. Setelah itu penerima dapat memverifikasi keaslian tanda tangan digital yang diterima dengan algoritma sebagai berikut:

1. Pastikan pasangan (r,s) adalah bilangan bulat pada selang $[1, n-1]$
2. Hitung e = Hash(m)
3. Hitung $w = s^{-1} \bmod n$
4. Hitung $u1 = ew \bmod n$
5. Hitung $u2 = rw \bmod n$
6. Hitung $X = u1G + u2Q$
7. Jika X = O maka dipastikan tanda tangan digital atau data adalah palsu.
8. Jika $X \neq O$, ubah koordinat x menjadi integer kemudian hitung $v = x \bmod n$
9. Jika $v = r$, maka tanda tangan dan data yang dikirim

adalah valid.

Beberapa hal yang perlu diperhatikan pada penggunaan ECDSA adalah sebelum digunakan untuk menghasilkan tanda tangan digital, pihak penerima dan pengirim harus menyetujui terlebih dahulu beberapa parameter yang akan digunakan pada ECDSA, parameter tersebut adalah kurva yang akan digunakan fungsi hash dan metode pengiriman kunci. Selain itu, bilangan seperti *k* dan *d* harus benar-benar dipilih secara acak, jika tidak keamanan sistem tidak dapat terjamin. Oleh karena itu diperlukan *Random Number Generator* yang telah tersertifikasi. Pemilihan bilangan pada ECDSA yang tidak random pernah menyebabkan kasus hacking oleh kelompok hacker bernama fail0verflow dan seorang hacker asal Amerika Serikat bernama George Hotz terhadap Sony di tahun 2010.

IV. APLIKASI ECDSA PADA ETHEREUM

Ethereum sebagai sebuah cryptocurrency bergantung pada algoritma dan prinsip kriptografi untuk bekerja. Salah satu prinsip yang digunakan adalah elliptic curve digital signature algorithm yang berguna untuk menghasilkan tanda tangan digital untuk mengamankan setiap transaksi pada network Ethereum. Terdapat dua aplikasi ECDSA pada Ethereum yaitu sebagai *address* suatu akun dan untuk menandatangani transaksi yang kemudian akan dimasukan ke dalam block.

Address adalah identitas suatu akun, address A(p_r) diperoleh dari sebuah kunci privat akun yang dipilih secara random p_r, kemudian diubah menjadi kunci publik p_b melewati ECDSA, setelah itu kunci publik tersebut di hash dengan algoritma *keccak-256.* 160 Bit terakhir dari hasil fungsi hash adalah address akun Ethereum. Berikut adalah contoh pembuatan address akun Ethereum:

p_r =bba7b944f2f794820b5403fbdc5de5f12785fbf30c5b88d4c038fe03e25f0886

p_b = ECDSA(p_r)

p_b =047b5e1258f8b3024bdbe254cdbb8396cb2662269ecc8de2fbd0bdb49253590fc595b574feccdf86614471732648413664d42bb3d142dc2663f097322a30fcfb74

A(p_r) = hash(p_b)

A(p_r)=d7f2101d33c05e50aa8c7a3dee63a8a1449fc4b72243e0f93cd2d8a228836835

Address berguna sebagai pengidentifikasi sebuah akun, sementara kunci privat berguna menandatangani setiap transaksi yang dilakukan oleh akun. Pada block akan terdapat data transaksi yang berisi jumlah ether pada transaksi serta kedua pihak yang terlibat, kedua pihak diwakilkan dengan kunci publik masing-masing. Kunci publik ini yang akan digunakan untuk memvalidasi validitas transaksi yang dilakukan. Berikut kita lihat skema pemberian tanda tangan digital serta verifikasi pada suatu data, anggap transaksi diwakilkan sebuah string “Matematika diskrit itu asik dan tidak membosankan”

p_r =0299815a85e01ef6cba31f57ac7be11d48c579e759613b1174234bc5394592b8

p_b =042e8488b9d9407a65ca8e4dc573a4aa0170ea2dcc84e1790cd202caba71de17f81a56f0279cfd8ce28a53505c28d806218781b7b60be63db14c9f041a7838d505

m = “Matematika diskrit itu asik dan tidak membosankan”

signature=3044022036282c7d7c45039eb1bbaede312388b221457cf68df336c0b6ac672a130b906f02207ec666eaf1e167ebe55de8b7646911333880833f182a78cbd9e67da593667b28

Tanda tangan digital selain berfungsi mengamankan transaksi pada Ethereum juga dapat mempercepat proses validasi dan verifikasi transaksi. Saat block melalui tahap mining untuk divalidasi kebenarannya, miner tinggal mengecek apakah data transaksi pada block sesuai dengan tanda tangan digital pada block tersebut, jika tidak sesuai maka ada indikasi pemalsuan atau penggandaan mata uang.

V. KESIMPULAN

Elliptical Curve Digital Signature Algorithm adalah algoritma yang memanfaatkan dasar-dasar teori bilangan serta kriptografi. ECDSA memungkinkan adanya produk-produk *cryptocurrency* seperti Ethereum dan Bitcoin yang merupakan masa depan pembayaran. ECDSA juga merupakan algoritma kriptografi yang sangat efektif karena sulit dipecahkan oleh komputer modern, oleh karena itu potensi aplikasi ECDSA sangat luas. Aplikasi ECDSA tidak hanya di bidang *cryptocurrency* tapi juga dapat digunakan di *cybersecurity*, *finance*, *e-commerce*, dan mengamankan data-data sensitif pemerintahan dan pribadi.

Ethereum sebagai *cryptocurrency* terpopuler ke-2 setelah bitcoin memiliki potensi yang sangat luas. Ethereum tidak dikendalikan oleh suatu pihak dengan agenda tertentu dan relatif aman dari penipuan jika dibanding mata uang lainnya. Selain itu, karena menggunakan teknologi blockchain tindakan penipuan dapat dengan mudah terdeteksi oleh network Ethereum. Namun, terdapat sisi negatif dari mata uang yang tidak diawasi dan tidak dikendalikan suatu otoritas, yaitu dapat digunakan dalam transaksi kejahatan tanpa bisa dilacak oleh pihak berwajib.

VI. UCAPAN TERIMA KASIH

Puji syukur penulis ucapkan kepada Tuhan Yang Maha Esa, karena atas rahmat-Nya penulis dapat menyelesaikan penulisan makalah ini. Penulis juga berterima kasih kepada tim dosen dan asisten yang telah dengan sabar membimbing penulis selama masa perkuliahan IF2120 semester 1 2020/2021. Penulis juga mengucapkan terima kasih kepada segenap teman-teman IF 2019 yang turut membantu penulis dalam perkuliahan dan dalam penyelesaian makalah ini.

REFERENSI

[1] http://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2020-2021/ECC-2020-Bagian1.pdf diakses pada 4 Desember 2020.

[2] D. Johnson, *The Elliptic Curve Digital Signature Algorithm*. Waterloo: University of Waterloo, pp.24-26.

[3] http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Teori-Bilangan-2020-Bagian1.pdf diakses pada 4 Desember 2020

[4] http://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2020-2021/Pengantar-Kriptografi-(2020).pdf diakses pada 4 Desember 2020
[5] https://www.investopedia.com/terms/c/cryptocurrency.asp diakses pada 4 Desember 2020
[6] https://teachcomputerscience.com/asymmetric-encryption/ diakses pada 4 Desember 2020
[7] https://wstein.org/edu/2007/spring/ent/ent-html/node89.html diakses pada 5 Desember 2020
[8] http://news.bbc.co.uk/2/hi/technology/8478764.stm diakses pada 11 Desember 2020.
[9] https://gavwood.com/paper.pdf diakses pada 5 Desember 2020.

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.

Bogor, 11 Desember 2020

Daniel Mario Reynaldi 13519031