BAB 2 LANDASAN TEORI

2.1 Transaction Fees

Transaction fees mengacu kepada suatu biaya yang dikeluarkan untuk menyelesaikan suatu transaksi [2] pada suatu blockchain (di dalam konteks ini adalah NEAR). Pada blockchain NEAR, pengguna akan membebankan gas fees setiap kali penggunanya melakukan mutasi terhadap data yang berada pada blockchain tersebut. Gas fees yang dibayarkan kemudian akan dibayarkan kepada para validator yang menjalankan node NEAR untuk tetap berjalan sebagai bentuk kompensasi [7]. Pada NEAR, terdapat 2 konsep terhadap gas, yaitu

- Gas units: Merupakan satuan yang digunakan untuk menghitung jumlah gas yang perlu dikeluarkan dalam suatu aksi dan tidak dihitung dalam satuan native token NEAR, melainkan satuan internal yang terdapat pada virtual machine.
- Gas price: Gas units kemudian akan dikalikan dengan gas price untuk menentukan berapa biaya yang harus dikeluarkan oleh pengguna untuk suatu aksi tertentu. Gas price akan secara otomatis dihitung kembali setiap blok berdasarkan kepadatan jaringan.

NEAR memiliki waktu blok kurang lebih satu detik yang dapat dicapai karena NEAR membatasi jumlah gas di setiap bloknya. Gas units secara sengaja dirancang oleh NEAR sehingga mudah dihitung dalam bentuk angka:

- 10¹² gas units atau 1 TGas (TeraGas)
- Sama dengan 1 milidetik waktu komputasi
- Yang sebagaimana berada pada harga minimum 100 juta yoctoNEAR yang setara dengan 0.1 milliNEAR

Karena sifat dari *gas units* NEAR yang bersifat *deterministic*, berikut adalah gambar yang dapat digunakan untuk melihat biaya gas units untuk *common actions* pada NEAR [8].

Operation	TGas	fee (mN)	fee (N)
Create Account	0.42	0.042	4.2×10 ⁻⁵
Send Funds	0.45	0.045	4.5×10 ⁻⁵
Stake	0.50	0.050	5.0×10 ⁻⁵
Add Full Access Key	0.42	0.042	4.2×10 ⁻⁵
Delete Key	0.41	0.041	4.1×10 ⁻⁵

Gambar 2.1. Cost of Common Actions [7]

Aksi komputasi yang lebih kompleks seperti pemanggilan fungsi, perhitungan *gas units* harus dilakukan secara manual dengan mencoba memanggil fungsi yang terkait. Pada penelitian ini akan dilakukan evaluasi terhadap rata-rata biaya transaksi pada setiap pemanggilan fungsi, yang rumusnya dapat didefinisikan sebagai berikut [9].

$$\overline{TxFee} = \frac{1}{n} \sum_{i=1}^{n} TxFee_i = \frac{TxFee_1 + TxFee_2 + \dots + TxFee_n}{n}$$
 (2.1)

di mana TxFee adalah biaya transaksi yang dikeluarkan untuk pemanggilan fungsi dalam satuan N (NEAR) dan i adalah jumlah transaksi yang dilakukan dalam simulasi percobaan.

2.2 Smart Contract

Smart contract dapat diartikan sebagai sebuah program yang disimpan pada suatu blockchain yang dijalankan ketika suatu kondisi yang telah ditentukan terpenuhi [10]. Smart contract biasanya digunakan untuk mengotomasi eksekusi dari sebuah perjanjian sehingga semua partisipan dapat yakin terhadap keluaran yang diberikan tanpa keikutsertaan pihak perantara maupun hilangnya waktu. Selain itu,

smart contract juga bisa digunakan untuk mengotomasi suatu alur kerja yang akan memicu aksi berikutnya ketika suatu kondisi terpenuhi.

Smart contract bekerja dengan cara mengikuti pernyataan seperti "if/when...then..." sederhana yang dibuat dalam bentuk kode yang nantinya akan dimasukkan ke dalam blockchain [11]. Kemudian, sekumpulan komputer yang berfungsi sebagai miner atau validator akan mengeksekusi perintah ketika suatu kondisi yang telah ditentukan sudah terpenuhi. Perintah-perintah ini dapat berupa memberikan uang ke pihak tertentu, memasukkan data ke dalam suatu basis data, mengecek suatu state, mengirim notifikasi, dan lain-lain. Blockchain akan memperbarui state dari smart contract tersebut setelah selesai dieksekusi. Karena ini, smart contract bersifat kekal (immutable), dan hanya pihak tertentu yang sudah diberikan akses dapat melihat hasil akhirnya. Karakteristik-karakteristik dari smart contract adalah sebagai berikut [10]:

- Kecepatan, efisiensi, dan akurasi: ketika suatu kondisi telah terpenuhi, contract tersebut akan langsung mengeksekusinya, dan karena *smart contract* bersifat digital dan otomatis maka tidak ada perlu proses manual untuk menyelesaikan kegiatan dari *smart contract* tersebut.
- Terpercaya dan transparan: Karena tidak ada pihak ketiga yang terlibat, dan karena juga catatan terenkripsi dari transaksi dibagikan ke para partisipan, tidak ada pihak yang perlu untuk mempertanyakan apakah informasi yang dikirimkan telah diubah secara diam-diam.
- Keamanan: Catatan transaksi pada blockchain telah dienkripsi sehingga membuat sulit untuk diretas. Selain itu, karena setiap catatan terhubungi pada catatan sebelumnya dan sesudahnya pada blockchain, peretas harus mengubah state di seluruh blockchain untuk mengubah sebuah catatan.

2.3 NEAR Protocol VERSITAS

NEAR Protocol (kemudian hanya disebut sebagai "NEAR" saja) adalah sebuah *platform blockchain* pengembangan terdesentralisasi yang dapat digunakan untuk hosting aplikasi *serverless* dan *smart contract*. Blockchain merupakan sebuah *decentralized ledger* yang melakukan transaksi dalam komputer-komputer yang tersebar di banyak tempat [12]. Elemen utama dari blockchain [12] adalah sebagai berikut.

- *Distributed Ledger Technology*: Semua partisipan dalam jaringan memiliki akses terhadap ledger terdistribusi dan catatan transaksi yang abadi (*immutable*). Transaksi hanya direkam sekali saja dan tidak dapat diubah, yang mengeliminasi potensi duplikasi.
- *Immutable Records*: Tidak ada partisipan yang dapat mengubah atau merusak sebuah transaksi setelah transaksi tersebut masuk ke dalam ledger.
- *Smart Contracts*: Sekumpulan aturan dapat disimpan di dalam blockchain dan dieksekusi secara otomatis. Smart contract dapat didefinisikan sebagai program yang berjalan di atas blockchain untuk melakukan suatu perintah dan bersifat immutable.

NEAR sendiri dibangun sedari awal dengan filosofi untuk menjadi platform blockchain yang paling mudah digunakan dari sisi pengembang dan pengguna [7]. Secara spesifik, NEAR dirancang agar untuk melakukan hal-hal berikut dengan mudah [13]:

- 1. Mudah untuk mengembangkan aplikasi terdesentralisasi meski pengembang hanya memiliki pengalaman dalam mengembangkan aplikasi web tradisional;
- 2. Mudah untuk melakukan onboarding pengguna dengan pengalaman yang sebaik mungkin meskipun mereka tidak ada pengalaman terhadap konsep kripto, token, kunci, dompet, dan istilah-istilah blockchain lainnya;
- 3. Mudah untuk melakukan scaling aplikasi secara mulus karena NEAR sendiri sudah mengatur kapasitas kemampuan melalui teknik sharding tanpa biaya tambahan atau usaha lebih dari pengembang aplikasi di atas platform NEAR.

Sesuai dengan filosofi NEAR yang mudah untuk digunakan, NEAR menggunakan *human readable account ID* sebagai identitas akun, alih-alih menggunakan *public key hash*. Identitas akun NEAR memiliki persyaratan sebagai berikut:

- Panjang minimum adalah 2 karakter
- Panjang maksimum adalah 64 karakter
- Account ID terdiri atas "Account ID parts" yang dipisahkan dengan tanda titik"."
- "Account ID part" terdiri atas simbol alfanumerik huruf kecil yang dipisahkan dengan garis bawah "_" atau strip "-"

Account pada NEAR juga mendukung untuk pembuatan *subaccount*, di mana seperti jika ada account dengan nama example.near maka mungkin untuk membuat *subaccount* dengan nama subaccount1.example.near, subaccount2.example.near, dan seterusnya dengan syarat hanya parent account yang memiliki izin untuk membuat *subaccount* tersebut. NEAR menggunakan algoritma consensus baru dan sebuah arsitektur sharding untuk mencapai performa yang relatif tinggi saat penggunaanya [13]. Kunci utama dari teknologi fondasi NEAR adalah:

- *Dynamic Resharding*: NEAR menggunakan teknik *sharding* yang merupakan bagian dari *horizontal scaling*. *Sharding* merupakan pendekatan untuk meningkatkan skalabilitas pada basis data yang melibatkan menyebarkan komputasi dan kapasitas penyimpanan pada beberapa server secara paralel. Karena sharding pada NEAR bersifat dinamis, NEAR akan secara berkala menyesuaikan jumlah *shards* yang aktif sesuai dengan kepadatan jaringan.
- *Nightshade*: Nightshade menggunakan teknik *sharding* yang baru bernama "Nightshade" untuk mencapai konsensus dan menyelesaikan transaksi antar *shard* yang aktif. Nightshade menawarkan model yang berbeda dibandingkan dengan teknik *sharding* yang sudah ada dengan cara mengubah abstraksi *sharding*. Nightshade mengasumsikan bahwa setiap *shard* bekerja Bersama untuk memproduksi potongan data yang secara kolektif akan membuat sebuah blok utuh.
- *Doomslug*: Doomslug merupakan teknik baru yang digunakan oleh NEAR Protocol untuk memproduksi sebuah blok pada *blockchain*. Menurut NEAR, Doomslug memungkinkan jaringan untuk mencapai *practical finality* setelah berkomunikasi hanya dalam satu periode, dengan *finality gadget* menyediakan finalitas BFT setelah periode kedua. *Practical finality* pada Doomslug (juga dikenal sebagai Finalitas Doomslug atau *Doomslug Finality*) adalah ketika sebuah blok tidak dapat diubah kecuali setidaknya ada satu partisipan yang rusak.
- Fees as Developer Rewards: NEAR akan mendistribusikan 30% dari biaya transaksi kepada sebuah contract yang dipanggil fungsinya. Pemilik contract dapat menentukan bagaimana mereka akan mengalokasikan atau menggunakan biaya yang diterima. Biaya ini ditujukan untuk memberikan insentif terhadap pengembangan aplikasi yang dilakukan pada NEAR.

• *Token-Based Storage*: Pemegang token NEAR memiliki hak untuk menyimpan data dalam *blockchain*. Sebagai contoh, jika seseorang memiliki saldo sebanyak 1 NEAR, maka berarti orang tersebut dapat menyimpan data sebesar hamper 10KB pada akun yang dimilikinya.

2.4 User Acceptance Testing

Tingkat penerimaan pengguna digunakan untuk mengukur intensi pengguna terhadap penggunaan aplikasi sosial media terdesentralisasi yang diukur melalui skala Likert yang dikumpulkan melalui kuesioner. Skala Likert digunakan untuk mengukur sikap, pendapat, dan persepsi seorang atau kelompok orang tentang fenomena sosial [14]. Jawaban pada skala Likert dapat dikonversikan menjadi sebuah nilai yang dapat dilihat pada Tabel 2.1.

Tabel 2.1. Tabel Skala Likert

Jawaban	Nilai/Skor
Sangat Setuju	5
Setuju	4
Netral	3
Tidak Setuju	2
Sangat Tidak Setuju	1

Kemudian dari data kuesioner yang telah dikumpulkan, tingkat penerimaan dapat diukur melalui rumus berikut [15]:

Skor Penerimaan =
$$\frac{\text{Jumlah skor jawaban kuesioner}}{\text{Total skor tertinggi}} * 100\%$$
 (2.2)

Sedangkan untuk interpretasi tingkat penerimaan pengguna dapat dirujuk kepada Tabel 2.2.

Tabel 2.2. Interpretasi Tingkat Penerimaan [15]

Range Skor	Keterangan	
0% - 19,99%	Sangat Tidak Setuju	
20% - 39,99%	Tidak Setuju	
40% - 59,99%	Netral	
60% - 79,99%	Setuju	
80% - 100%	Sangat Setuju	

