

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Penelitian ini didasarkan pada berbagai literatur ilmiah dan dokumen teknis yang relevan, terutama mengenai teknologi *Blockchain*, *Smart contract*, distribusi *token (Airdrop)*, serta efisiensi dan keamanan sistem berbasis *Ethereum*. Bagian ini merangkum hasil-hasil penelitian sebelumnya dan studi kasus terkait sebagai landasan untuk merancang *Smart contract Airdrop* yang efisien.

2.1.1 *Smart contract* untuk *Airdrop*

Fröwis dan Böhme (2019) meneliti praktik distribusi *token* menggunakan kontrak pintar (*Smart contract*) di *Ethereum*, dan mengungkapkan bahwa metode push-style konvensional di mana *token* dikirim satu per satu ke setiap penerima sangat boros *gas fee*. Untuk distribusi *token* dalam skala besar, biaya *Gas fee* yang dibutuhkan bisa melampaui anggaran awal proyek. Mereka menekankan pentingnya efisiensi dalam desain *Smart contract*.

Wang *et al.* (2019) dalam kajiannya mengenai *Smart contract* berbasis *Ethereum* menyarankan penerapan *Batch processing* dan penggunaan *array* dalam *Smart contract* sebagai metode pengurangan konsumsi *gas fee*. Teknik ini memungkinkan pengiriman ke banyak penerima sekaligus dalam satu transaksi, sehingga lebih hemat dan cepat.

Contoh kasus dari implementasi *Airdrop* yang sukses adalah Uniswap (2020), yang membagikan *token UNI* secara gratis kepada pengguna awalnya. Strategi ini tidak hanya memicu loyalitas komunitas, tetapi juga menciptakan efek viral yang meningkatkan nilai proyek. Praktik ini menjadi acuan dalam desain sistem distribusi *token* berbasis komunitas.

2.1.2 Keamanan *Smart contract*

Keamanan *Smart contract* menjadi aspek penting dalam pengembangannya. Atzei, Bartoletti, dan Cimoli (2017) mengidentifikasi berbagai kerentanan yang umum ditemukan pada kontrak Solidity, seperti reentrancy, ConsenSys (2023)

menyarankan penerapan audit otomatis menggunakan alat bantu seperti *MythX* dan *Slither* dalam proses pengembangan kontrak. *Tools* ini mampu mendeteksi kerentanan sejak tahap awal pengembangan dan membantu tim pengembang menjaga integritas serta keamanan sistem kontraknya.

2.1.3 Optimasi *Gas fee* dan Efisiensi Kontrak

Nelaturu *et al.* (2021) memperkenalkan teknik *loop summarization* untuk mengoptimalkan perulangan dalam kode *Solidity*. Teknik ini dapat memangkas konsumsi *Gas fee* secara signifikan dengan tetap menjaga fungsionalitas kontrak. Studi tersebut membuktikan bahwa efisiensi *Gas fee* bisa ditingkatkan hingga 21% pada beberapa skenario distribusi massal.

Chen *et al.* (2023) juga menyoroti bahwa penggunaan struktur data yang sederhana, penghindaran penyimpanan yang tidak perlu, dan modularitas kode akan berdampak langsung terhadap efisiensi dan skalabilitas kontrak pintar.

2.1.4 Permasalahan Skalabilitas dan Solusi Teknologi Baru

Masalah skalabilitas pada *Blockchain Ethereum* telah mendorong munculnya berbagai solusi Layer 1 dan Layer 2. Das *et al.* (2025) menekankan bahwa eksekusi transaksi secara berurutan merupakan *bottleneck* utama dalam sistem *Ethereum*. Mereka mengusulkan eksekusi transaksi paralel (*parallel transaction execution*) sebagai alternatif untuk meningkatkan *Throughput* jaringan. *Monad*, sebagai *Blockchain Layer 1* baru, mengimplementasikan eksekusi paralel, *asynchronous transaction handling*, serta memiliki waktu blok yang lebih cepat dibanding *Ethereum*. Dalam whitepaper-nya (Monad Labs, 2023), jaringan ini disebut mampu menangani lebih dari 10.000 transaksi per detik dengan latensi sangat rendah. *Testnet Monad* juga telah digunakan dalam simulasi distribusi *Airdrop* kepada jutaan alamat dengan hasil yang sangat efisien (Monad Labs, 2024).

2.1.5 Strategi dan Dampak *Airdrop*

Allen *et al.* (2022) melakukan analisis komprehensif terhadap berbagai jenis *Airdrop* dan menunjukkan bahwa proyek yang melaksanakan *Airdrop* dengan struktur insentif yang tepat cenderung memperoleh partisipasi dan loyalitas komunitas lebih tinggi. Selain sebagai promosi, *Airdrop* juga memiliki peran strategis dalam mendistribusikan kepemilikan *token* secara adil dan memperkuat desentralisasi proyek.

2.2 Dasar Teori

Bagian ini menjelaskan konsep-konsep dasar yang menjadi fondasi dari penelitian, yaitu *Blockchain*, *Smart contract*, *Airdrop*, *token ERC-20*, *Gas fee*, dan jaringan *Monad Testnet*.

2.2.1 *Blockchain*

Blockchain adalah buku besar digital terdesentralisasi yang menyimpan catatan transaksi secara permanen dalam blok yang saling terhubung dan dienkripsi. Teknologi ini menjamin transparansi, keamanan, dan imutabilitas data. Konsensus dalam jaringan *Blockchain* seperti *Proof of Work* (PoW) atau *Proof of Stake* (PoS) digunakan untuk memvalidasi transaksi tanpa memerlukan otoritas pusat (Nakamoto, 2008; Tapscott & Tapscott, 2016).

2.2.2 *Smart contract*

Smart contract adalah kontrak digital yang dijalankan secara otomatis di atas *Blockchain*. Konsep ini diperkenalkan oleh Szabo (1997) dan diimplementasikan secara nyata oleh *Ethereum* (Buterin, 2014). *Smart contract* memungkinkan dua pihak yang tidak saling percaya untuk melakukan transaksi tanpa perantara, karena aturan kontrak tertulis dalam kode dan dijalankan tanpa campur tangan manusia.

2.2.3 *Token ERC-20*

ERC-20 adalah standar *token* yang berlaku dalam ekosistem *Ethereum*. Standar ini menentukan fungsi-fungsi dasar seperti transfer, *balanceOf*, dan *approve*, sehingga kompatibel dengan dompet, bursa, dan kontrak lainnya. *Token ERC-20* banyak digunakan dalam *ICO* dan distribusi *Airdrop* karena keumuman dan dukungan luasnya di ekosistem *Ethereum*.

2.2.4 *Airdrop*

Airdrop adalah metode distribusi *token* kripto secara gratis kepada pengguna berdasarkan kriteria tertentu. *Airdrop* digunakan untuk membangun komunitas, mempromosikan proyek baru, dan meningkatkan desentralisasi. Terdapat berbagai jenis *Airdrop* seperti standard *Airdrop* (berdasarkan kepemilikan), *bounty Airdrop* (berdasarkan aktivitas promosi), dan holder *Airdrop* (berdasarkan lamanya kepemilikan *token*) (CoinMarketCap, 2023).

2.2.5 *Gas fee*

Gas fee adalah biaya yang harus dibayarkan pengguna untuk menjalankan transaksi atau kontrak di jaringan *Blockchain*. Biaya ini dibutuhkan karena setiap instruksi yang dijalankan dalam *Ethereum Virtual Machine (EVM)* membutuhkan sumber daya komputasi. Harga *Gas fee* bersifat dinamis dan bergantung pada kondisi jaringan saat transaksi dilakukan.

2.2.6 *Monad Testnet*

Monad Testnet adalah jaringan uji coba dari *Blockchain Monad* yang dirancang kompatibel dengan *Ethereum*. *Monad* mendukung *EVM*, *RPC API*, dan memungkinkan pengembangan *Smart contract* dengan efisiensi tinggi. Dengan dukungan eksekusi paralel, waktu blok cepat, dan biaya transaksi rendah, *Monad* menjadi pilihan ideal untuk pengujian distribusi *token* massal. Fitur-fitur ini memberikan lingkungan optimal bagi pengembang untuk menguji kontrak sebelum dirilis ke jaringan utama.