BAB 2

LANDASAN TEORI

2.1 Desentralisasi Aplikasi Web

Pada bagian bab kedua terkait dengan landasan teori sebagai pedoman dalam melakukan penelitian. Lalu dalam bab ini terdapat pembahasan teori-teori mengenai system DApp web *smart contract*, teknologi Blockchain¸ jaringan Ethereum, dan metodologi pengembangan system dan juga hasil penelitian sebelumnya. Tahapan ini yaitu melakukan literature review untuk memahami apa saja yang dibutuhkan untuk implementasi blockchain dalam pengaplikasian smart contract. Selain itu juga untuk menelusuri lebih dalam dari permasalahan yang terjadi saat ini. Sumber untuk materi didapatkan dari e-book, jurnal, dan lainnya. Dari hasil literature review yang telah dilakukan seperti banyaknya(mahal) harga value dari memakai transaksi "gas" ini memang sedang terjadi. Maka dibutuhkan teknologi yang mampu menghemat value salah satunya dengan teknologi smart contract pada blockchain Ethereum.

Metode ini dapat membuat penghematan dari gas price biasanya terkadang agak mahal, salah satunya dengan teknologi algoritma hashing keccak256 yang membuat satu blok data dengan blok data lainnya saling mengunci yang membuat harga gas tidak berlebihan dalam melakukan smart contract pada blockchain Ethereum. Smart contract berfungsi sebagai tempat untuk mengatur, mengelola, dan tempat sarana melakukan pemesanan transaksi online. Blockchain Ethereum dipilih karena memiliki tingkat komputasi tinggi yaitu 167,86 TH/s dimana semakin tinggi tingkat komputasi pada blockchain membuat data yang tersimpan tidak mudah dirusak [39]. Transaksi smart contract pada blockchain Ethereum bisa dilacak history-nya, dari ID siapa yang melakukan transaksi smart contract tersebut dan juga kepada ID siapa transaksi itu dikirim [26].

Berikut juga merupakan hasil penelitian sebelumnya yaitu oleh:

1. Ting Chen, Xiaoqi Li, Xiapu Luo, Xiaosong Zhang yang merupakan bagian dari Center for Cybersecurity, University of Electronic Science and Technology di China yang berjudul "*Under-Optimized Smart contracts Devour Your Money*" berkata bahwa Gas pada Ethereum Network merupakan sebuah unit untuk membeli kekuatan menghitung dari pada miners karena smart contract dijalankan pada mesin

miners. Gas juga digunakan untuk proses pembuatan dan “pelancar” dari smart contract.

Yang disini dijelaskan bahwa operasi untuk sistem penyimpanan pada Ethereum memiliki tarif yang paling mahal diantara operasi lain. Untuk menulis data baru pada smart contract memerlukan 20.000 gas, dan jika mengganti data yang sudah ada akan memerlukan 5.000 gas [14]. Pada kenyataan dilapangan, banyak praktik - praktik kode yang masih dapat menyebabkan mahalnya harga transaksi ke smart contract dalam hal proses penyimpanan dan operasi. Salah dua kategori dari praktik-praktik tersebut adalah dead code, dan loop-related patterns [32].

Kategori dead code adalah sebuah pola dimana terdapat beberapa perintah dan fungsi yang tidak akan mungkin dijalankan oleh smart contract. Hal ini dikarenakan praktik kondisi yang kurang baik. Perintah-perintah tersebut berkontribusi pada besar tidaknya biaya dari transaksi jika ingin berinteraksi dengan smart contract. Sangat disayangkan jika membayar untuk sebuah kode yang tidak akan pernah dijalankan [32].

Dan dalam penelitian ini mereka menemukan 93,5%, 90,1% dan 80% masing-masing dari pola smart contract ini kurang teroptimasi dikarenakan dari dead code dan sifat “avalanche” yang kurang mengunci antar node blocknya, maka dari itulah penelitian itu dibuat.

2. Berikutnya disebut di artikel medium oleh [11] bahwa Ada beberapa fungsi hash bawaan dalam kontrak pintar yang dapat anda gunakan: keccak256、sha256 dan ripemd160. Semakin banyak parameter, semakin banyak gas yang dikonsumsi. Konsumsi gas: ripemd160> sha256> keccak256. Jadi jika tidak ada tujuan lain, disarankan untuk menggunakan keccak256.

3. Berikutnya yaitu oleh [40] disebutkan jika panjang byte dapat diminimalisir, gunakan jumlah serendah mungkin dari byte1 hingga byte paling terendah yaitu byte32. Dan juga tipe data byte32 lebih murah untuk digunakan dari pada tipe string.

Dari landasan teori dan hasil penelitan sebelumnya yang telah dilakukan seperti banyaknya(mahal) harga value dari memakai transaksi “gas” ini memang sedang terjadi. Maka dibutuhkan teknologi yang mampu menghemat harga value gas salah satunya dengan membangun system teknologi blockchain perangkat lunak Dapp web yang dapat membantu mengurangi harga gas pada setiap transaksi di

jaringan Ethereum. Oleh sebab itu blockchain Ethereum bisa digunakan untuk menjaga bentuk asli sebuah transaksi yang bersifat ter-desentralisasi itu tanpa mempunyai sifat terpusat, karena bisa mengetahui dari ID siapa yang melakukan transaksi maupun yang mengirim atau menerima. Namun, di blockchain sendiri ada permasalahan biaya gas yang cukup banyak value nya (mahal) dalam mengunggah sebuah data kedalam blockchain, oleh karena itu diperlukan bantuan optimasi hashing keccak256 sebagai solusi penyimpanan data.

2.2 Teknologi Blockchain

Blockchain, sebagian besar dikenal sebagai teknologi backbone di belakang Bitcoin, yang merupakan salah satu teknologi terpanas dan paling menarik saat ini di pasar. Sejak 2013 pencarian Google untuk "blockchain" telah meningkat sebanyak 1.900%. Sangat mirip dengan meningkatnya internet, blockchain memiliki potensi yang benar-benar mengganggu beberapa industri dan membuat proses lebih demokratis, aman, transparan, dan efisien. Para pengusaha, perusahaan startup, investor, organisasi global dan pemerintah semuanya telah mengidentifikasi blockchain sebagai teknologi revolusioner. Semua teknologi yang muncul mestinya mempunyai beberapa dampak positif dan negatif dalam penerapannya.Menurut [34] beberapa manfaat positif dan tantangan blockchain yaitu seperti table dibawah ini:

Manfaat Positive Blockchain	**Tantangan Blockhain**
1. Disintermediasi Dua pihak dapat melakukan pertukaran tanpa pengawasan atau intermediasi dari pihak ketiga, sangat mengurangi atau bahkan menghilangkan risiko counterparty (pihak ketiga atau peserta, baik bank atau pelanggan, dengan siapa transaksi keuangan dibuat, atau	1. Teknologi yang baru lahir Mengatasi masalah tantangan seperti kecepatan transaksi, proses verifikasi, dan batas data akan sangat penting dalam membuat blockchain yang dapat diterapkan secara luas.

pihak-pihak yang melakukan transaksi keuangan).	
2. Para penggunanya diberi kuasa Para penggunanya memegang kendali atas semua informasi dan transaksi mereka.	2. Status peraturan belum pasti Karena mata uang modern yang selalu diciptakan dan diatur oleh pemerintah nasional, blockchain dan Bitcoin menghadapi rintangan pada masalah adopsi/penerapan oleh lembaga keuangan yang sudah ada jika statusnya peraturan pemerintah belum ditetapkan.
3. Data yang berkualitas tinggi Data Blockchain secara lengkap, konsisten, tepat waktu, akurat, dan tersedia secara luas.	3. Memakan energi yang besar Para penambang Bitcoin di jaringan blockchain ini sedang mencoba 450 ribu triliun solusi per detik dalam upaya untuk memvalidasi transaksi, menggunakan sejumlah besar daya komputer.
4. Daya tahan, kehandalan, dan panjang umur Karena adanya jaringan desentralisasi, blockchain tidak memiliki titik pusat kegagalan dan	4. Kontrol, keamanan, dan privasi Selain adanya sebuah solusi, termasuk blockchain private atau permissioned dan enkripsi yang kuat, masih ada kekhawatiran keamanan

lebih mampu menahan segala serangan berbahaya	cyber yang perlu diatasi sebelum masyarakat umum mempercayakan data pribadi mereka pada solusi blockchain.
5. Proses yang berintegritas Pengguna dapat mempercayai bahwa transaksi akan dilaksanakan sama persis seperti perintah protokol dan menghilangkan kebutuhan untuk pihak ketiga.	5. Masalah integrasi Aplikasi Blockchain menawarkan solusi yang memerlukan perubahan secara signifikan atau penggantian lengkap pada sistem yang ada. Dalam rangka untuk membuat perubahan, perusahaan harus menyusun strategi transisi.
6. Transparan dan immutabilitas Perubahan pada blockchains publik dapat dilihat secara terbuka oleh semua pihak yang menciptakan transparansi, dan semua transaksi yang tidak dapat berubah, yang berarti mereka tidak dapat diubah atau dihapus.	6. Penerapan secara kultural Blockchain merupakan pergeseran lengkap untuk jaringan desentralisasi yang membutuhkan buy-in pengguna dan operator.
7. Mempermudah ekosistem Dengan semua transaksi yang ditambahkan ke Ledger umum tunggal/single publik ledger, mengurangi kekacauan dan	7. Biaya Blockchain menawarkan penghematan yang luar biasa pada masalah biaya transaksi dan waktu

komplikasi dari beberapa buku besar/ledger	namun biaya pada modal awal yang tinggi bisa menjadi penghalang.
0. 8. Transaksi yang lebih cepat Transaksi antar bank berpotensi dapat memakan waktu hingga beberapa hari untuk kliring dan penyelesaian akhir, khususnya di luar jam kerja. Namun transaksi Blockchain dapat mengurangi waktu transaksi dalam hitungan menit dan diproses selama 24/7.	
1. 9. Biaya transaksi yang lebih rendah Dengan menghilangkan perantara pihak ketiga dan biaya overhead untuk bertukar aset, blockchain memiliki potensi untuk mengurangi biaya transaksi.	

Table 2.0.1 Teknologi Blockchain

2.2.1 Definisi Blockchain

Sejarah awal mula penemuan Bitcoin (uang digital) pada akhir tahun 2008, yang ditemukan oleh seorang yang bernama Satoshi Nakamato, serta dalam paper yang berjudul “Bitcoin: A Peer-to-Peer Electronic Cash System”. Di mana dirinya menuliskan gagasan terkait pemanfaatan teknologi jaringan Peer-to-Peer. Menurut [30] definisi Peer-toPeer atau yang dikenal dengan P2P adalah jaringan terdistribusi yang dapat berbagi berkas media dan juga bertukar data antara dua komputer (peer) atau jenis jaringan tanpa adanya perantara. Untuk menangani transaksi elektronik

yang telah dibahas dalam paper tersebut terkait konsep cara bertransaksi dengan uang digital (Bitcoin) secara daring tanpa menggunakan pihak ketiga dan tanpa penyimpanan secara terpusat atau terdistribusi, penerapan konsep Peer-to-Peer tentu dapat dikatakan sudah sesuai untuk memberikan solusi terkait metode transaksi dengan menggunakan Bitcoin [25].

Melalui temuan cara bertransaksi Bitcoin tersebut, secara bersamaan konsep Blockchain pun pada awalnya yang hanya digunakan untuk mengamankan transaksi uang digital tersebut, hingga sekarang telah mengalami perkembangan pesat yang dapat diterapkan dalam berbagai hal, terutama pada bidang digital yang mengutamakan kepercayaan, keamanan, dan kevaliditasan sebuah transaksi data. Blockchain merupakan ledger atau buku besar digital yang terdistribusi dari transaksi yang ditandatangani secara kriptografis dan dikelompokkan ke dalam blok. Setiap blok dihubungkan secara kriptografis dengan hash blok sebelumnya setelah dilakukan validasi dan menjalani keputusan konsensus. Ketika blok baru berhasil dibuat dari proses mining, data pada blok sebelumnya akan hampir mustahil untuk diubah atau dimanipulasi [37].

Berkaitan dengan definisi Blockchain yang telah dijelaskan menurut [37] dapat ditarik kesimpulan mengenai definisi Blockchain secara umum, bahwa Blockchain merupakan database terdistribusi yang mencatat setiap terjadinya transaksi atau pertukaran dalam setiap blok dan dilindungi dengan metode keamanan kriptografi, sehingga aman dan tidak dapat mudah diubah nilainya. Namun, pada kenyataannya masih ditemukan pemikiran terkait definisi Blockchain dan cryptocurrency seperti Bitcoin merupakan hal yang sama pada masyarakat awam. Sebenarnya pemikiran tersebut tentu saja merupakan sebuah kekeliruan yang harus diluruskan. Pada dasarnya Blockchain tentu saja tidak sama dengan cryptocurrency seperti Bitcoin atau mata uang digital lainnya. Berikut terdapat uraian perbedaan antara Blockchain dan cryptocurrency pada Table 2.2:

Blockchain	**Cryptocurrency**
Blockchain merupakan teknologi seperti ledger atau basis	Cryptocurrency adalah mata uang digital yang juga merupakan

data karena menyimpan segala informasi pertukaran data, seperti mata uang digital hingga surat sertifikat kepemilikan tanah.	salah satu implementasi teknologi Blockchain, karena proses transaksinya disimpan pada Blockchain.
Tujuan dari teknologi Blockchain adalah untuk membuat biaya pertukaran nilai menjadi rendah dan memiliki lingkungan yang aman dalam transaksi peer-to-peer dengan siapa-pun.	Tujuan dari adanya cryptocurrency adalah untuk menyederhanakan dan meningkatkan kecepatan transaksi keuangan tanpa adanya batasan dari pihak tertentu.
Ranah pembahasan Blockchain sangat luas dan akan terus berkembang.	Cryptocurrency hanya terbatas pada pertukaran mata uang digital saja.

Table 2.0.2 Definisi Blockchain

Pada penjelasan singkat perbedaan pada Table 2.2, dapat diambil kesimpulan bahwa sebenarnya cryptocurrency seperti Bitcoin, Ether, Litecoin dan berbagai jenis cryptocurenncy lainnya merupakan salah satu contoh hasil penerapan atau implementasi dari teknologi Blockchain, dan dapat dikatakan bahwa Blockcain tentu saja dapat berfungsi atau digunakan tanpa menggunakan cryptocurrency. Namun, cryptocurrency tentu tidak akan dapat digunakantanpa teknologi Blockchain.

2.2.2 Jenis Blockchain

Berdasarkan jenis Blockchain terdapat tiga jenis Blockchain yang umum diketahui beserta perbedaan dan tujuannya, yaitu:

- **Public Blockhain**

Seperti namanya, Blockchain ini merupakan jaringan terdistribusi yang besar karena memiliki sifat publik yang berarti terbuka kepada semua orang yang berpartisipasi dan memiliki kode yang bersifat open-source, sehingga para komunitas dapat berdistribusi. Tujuan dari jenis Blockchain ini banyak digunakan untuk melakukan transaksi mata uang digital atau cryptocurenncy, di mana semua orang dapat melihat daftar transaksi yang pernah dilakukan dan memvalidasi transaksi.

- **Private Blockhain**

Private Blockchain adalah salah satu jenis Blockchain yang bersifat tertutup dan bertujuan untuk melakukan pertukaran informasi secara internal saja. Tentu hal tersebut dapat membuat pihak-pihak yang tidak bergabung, tidak dapat melihat proses-proses apa saja yang dilakukan pada Blockchain tersebut. Menurut Mukhopadyay[21], terdapat batasan akses pada private Blockchain. Apabila terdapat organisasi atau perusahaan yang menerapkan teknologi Blockchain secara umum. Namun, tidak terlalu nyaman dengan akses kontrol yang telah diberikan oleh jaringan publik (public Blockchain), tentu saja tujuan tersebut dapat dicapai dengan memanfaatkan Blockchain yang bersifat private ini.

- **Semi Private Blockhain**

Semi-private Blockchain atau sering disebut sebagai consortium Blockchain, merupakan jenis Blockchain yang memberikan hak akses kepada siapa saja yang berhak menggunakannya dan memiliki source

code yang tertutup. Mirip seperti dengan private Blockchain. Namun, untuk penyimpanan data yang dikirimkan melalui transaksi tetap akan tersimpan pada jaringan Blockchain public.

Table 2.0.3 Jenis Blockchain

2.2.3 Perkembangan Blockhain

Menurut perkembangannya Blockchain hingga saat ini terbagi menjadi tiga era sejak pertama kali diperkenalkan dengan penemuan Bitcoin pada sekitar akhir tahun 2008 [7], tiga era perkembangan Blockchain yaitu:

Blockchain 1.0	Generasi pertama dari Blockchain yang diawali dengan kemunculan istilah Bitcoin dan secara dasar digunakan untuk cryptocurrency atau mata uang digital, juga termasuk teknik kriptografi keuangan dalam mengamankan proses transaksi dan aplikasi secara publik.
Blockchain 2.0	Implementasi Blockchain untuk layanan keuangan dan kontrak cerdas (*smart contract*) diperkenalkan secara khusus pada generasi Blockchain 2.0 melalui platform jaringan yang bernama Ethereum. Selain itu juga berbagai macam pelayanan lainnya seperti perusahaan pasar juga mulai menggunakan layanan Blockchain. Pada generasi ini juga Blockchain lebih fleksibel terhadap kebutuhan penggunanya.
Blockchain 3.0	Pada generasi ketiga, Blockchain sudah digunakan untuk diimplementasikan padaaplikasi di luar industri jasa keuangan dan digunakan di industri yang lebih bersifat umum seperti pemerintahan, kesehatan, kepemilikkan karya seni, proses peradilan, dan lain sebagainya.

Table 2.0.4 Perkembangan Blockchain

2.2.4 Kelebihan Teknologi Blockhain

Beberapa contoh kelebihan dari teknologi Blockchain yang telah diketahui dari definisi penjelasan sebelumnya yaitu:

a. Transparansi atau keterbukaan, dalam Blockchain menerapkan sistem yang transparan supaya proses yang ada di dalamnya dapat dilihat dan dibagikan kepada semua orang.
b. Kekal atau tetap, karena hanya terjadi sekali penulisan data pada Blockchain dan apabila data tersebut diubah, akan sangat susah sekali dan hampir tidak mungkin untuk mengubah semua data yang telah tersimpan pada Blockchain. Sebab data yang akan diubah akan mempengaruhi catatan transaksi setelahnya, sehingga dengan mengubah sebuah data, diperlukan upaya untuk mengubah hampir seluruh rekaman data yang telah ada.
c. Memiliki sistem keamanan yang kuat dengan menerapkan kriptografi seperti fungsi hash untuk memverifikasi dan menjaga integritas data pada setiap block sehingga valid serta mencegah dari adanya perubahan data.
d. Memiliki kemudahan dalam melacak setiap data transaksi pada jaringan Blockchain, karena data transaksi yang disimpan pada jaringan Blockchain tentu akan merujuk pada transaksi sebelumnya, sehingga hal ini dapat mempermudah dalam proses verifikasi dan pencarian data transaksi.
e. Sifat anonymous, meskipun data yang disimpan pada jaringan publik Blockchainbersifat transaparan atau dapat dilihat oleh orang lain. Namun, terkait dengan identitas setiap pengguna yang mengirimkan maupun menerima transaksi dalam jaringan Blockchain menggunakan suatu alamat tertentu atau yang disbeut dengan public key, dan dalam hal ini, identitas sebenarnya dari setiap pengguna tidak ditampilkan pada interaksi transaksi tersebut.

2.2.5 Struktur Blockchain

Di balik bagaimana cara proses Blockchain bekerja, tentunya terdapat bagian-bagian penting yang terstruktur supaya Blockchain dapat digunakan. Menurut [19], struktur dari Blockchain terdiri dari 3 bagian komponen utama yaitu:

- **Blok (*block*)**

Blockchain tersusun dari banyaknya block yang merupakan representasi untuk sebuah daftar transaksi yang sah dan disimpan. Setiap blok memiliki sebuah hash kriptografis sebagai pointer atau sebagai identitas setiap blok supaya dapat saling terhubung antara satu dengan yang lainnya.

Menurut [2] struktur dari sebuah blok terdiri dari header, diikuti dengan metadata dan daftar transaksi yang disimpan. Pada Gambar 2.1, terdapat representasi struktur sebuah blok yang memiliki berbagai komponen. Berikut penjelasan terkait komponen yang ada pada setiap blok pada jaringan Blockchain:

Block Size merupakan bagian pertama dari struktur blok yang menyimpan informasi terkait dengan ukuran sebuah blok dalam bytes.

Block Header merupakan bagian dari sebuah blok yang memiliki ukuran 80 bytes dan menyimpan sekumpulan metadata, seperti:

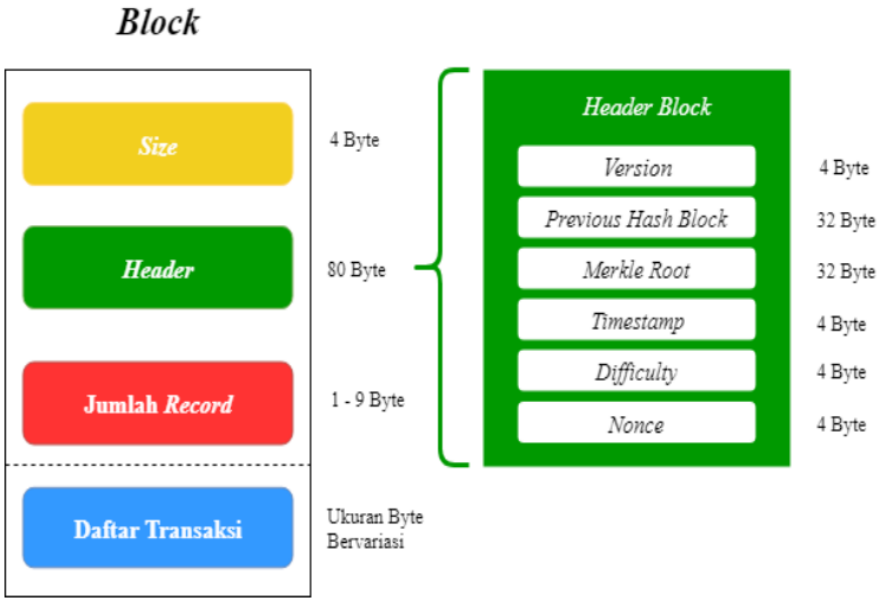
Version: Menyimpan informasi versi sebuah blok dan memiliki ukuran sebesar 4 bytes.
Previous Block Hash: Metadata yang menyimpan hash pada blok sebelumnya, juga berfungsi sebagai "rantai" yang menghubungkan dengan blok tersebut dengan blok sebelumnya dan memiliki ukuran sebesar 32 bytes.
Merkle Root:

Merupakan sekumpulan informasi dari semua transaksi yang telah dilakukan hash pada blok tersebut dengan memiliki ukuran sebesar 32 bytes dan bertujuan untuk memberikan kesimpulan dari semua transaksi yang dilakukan blok tersebut.
Timestamp: Menyimpan informasi terkait timestamp atau kapan waktu blok tersebut dibuat dengan memiliki ukuran sebesar 4 bytes.
Difficulty Target: Menyimpan informasi terkait tingkat kesulitan algoritma PoW (*Proof of Work*) yang digunakan dan memiliki ukuran sebesar 4 bytes.
Nonce: Merupakan angka acak yang disimpan dengan ukuran sebesar 4 bytes dan digunakan dalam proses penambangan blok baru.

Table 2.0.5 Block Header

Jumlah Record adalah bagian dari blok yang menghitung sebarapa banyak transaksi yang dilakukan dan biasanya memiliki ukuran 1-9 bytes.

Daftar transaksi merupakan bagian yang menyimpan kumpulan data transaksi yang telah dilakukan pada sebuah blok tersebut dengan ukuran data yang bervariasi.



Gambar 2.0.1 Struktur Sebuah Block

- **Rantai (*chain*)**

Supaya setiap block pada Blockchain saling terhubung, diperlukanlah "rantai" dalam bentuk hash yang menghubungkan antara satu block dengan block lainnya. Mekanisme hash merupakan salah satu konsep yang rumit secara matematis untuk diterapkan pada Blockchain. Meskipun Blockchain dianggap merupakan inovasi teknologi terbaru. Namun, tidak dengan hash. Konsep hashing tentunya sudah ada sejak sekitar 30 tahun yang lalu, dan digunakan pada konsep Blockchain karena hash hanya dapat membuat fungsi satu-arah yang tidak dapat dilakukan dekripsi. Fungsi sebuah hashing menciptakan algoritma matematis yang memetakan ata dengan segala ukuran ke dalam karakter bit yang biasanya memiliki panjang sebanyak 32 karakter, yang mana panjang ukuran bit tersebut mempresentasikan data yang telah di-hash.

Secure Hash Algorithm (SHA) merupakan salah satu fungsi hash yang digunakan oleh Blockchain, sedangkan algoritma yang biasa digunakan untuk melakukan hash pada Blockchain menggunakan algoritma SHA-256 yang dapat mengubah panjang ukuran data apapun menjadi sebuah karakter hash dengan ukuran 256 bits (32 bytes), sehingga pada Blockchain, hash bisa dianggap sebagai kumpulan dari nilai transaksi yang digabung dan bersifat unik dari data pada sebuah

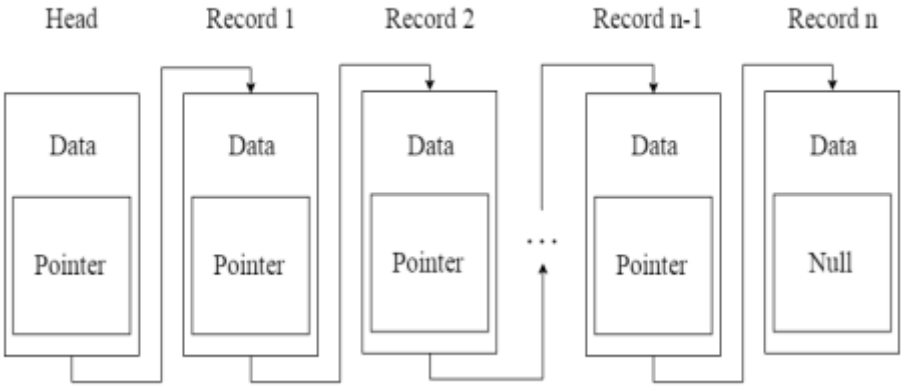
block untuk mengunci block supaya tetap menjadi 1 nilai address di dalam Blockchain.

- **Jaringan (*network*)**

Istilah jaringan atau network pada Blockchain merupakan representasi dari banyaknya nodes atau komputer yang saling terhubung satu sama lain dan menjalankan sebuah algoritma untuk mengamankan jaringan. Pada setiap node memiliki rekaman dari seluruh transaksi yang terekam pada Blockchain. Para node tersebut berlokasikan di seluruh dunia dan dikelola oleh setiap orang yang tergabung dalam jaringan Blockchain. Sudah sangat jelas terkait dengan topologi jaringan yang digunakan oleh Blockchain yaitu Peer-to-Peer, yang mana dari seluruh node dapat saling berkomunikasi antar satu node dengan node yang lain untuk menerima maupun mengirim pesan.

2.2.6 Kerja Blockchain

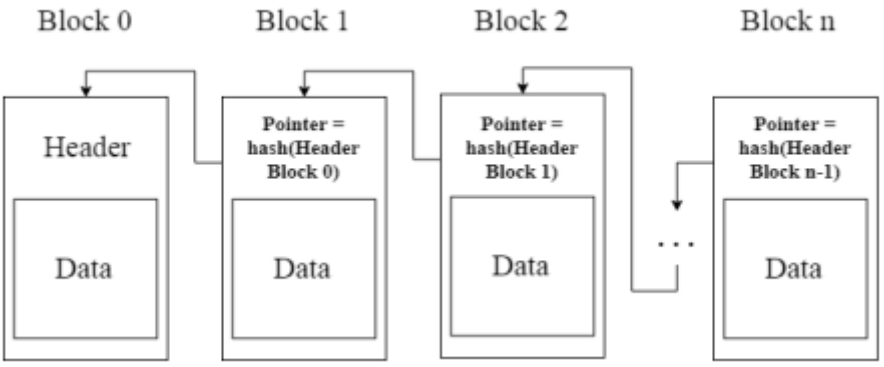
Blockchain dapat digambarkan seperti sekumpulan banyak blok yang tersambung dan membentuk seperti rantai. Pada hakikatnya Blockchain memiliki persamaan dengan cara kerja salah satu sebuah koleksi linear dari struktur data yaitu linked list atau sernarai berantai, dimana pada linked list terdiri dari struktur data yang digambarkan sebagai node tersebut berisi data yang disimpan dan akan merujuk pada node lain dengan bantuan pointer Gambar 2.2. Untuk mencari data dalam linked list, perlu dilakukan penelusuran dari node pertama hingga node yang menyimpan data yang dicari itu ditemukan.



Gambar 2.0.2 Diagram Skema Linked List

Sama halnya seperti pada Blockchain, di mana sebuah blok pada Blockchain tersebut terdiri dari data yang terstruktur dan memiliki tujuan untuk menyimpan sekumpulan data atau daftar transaksi, serta mendistribusikannya kepada seluruh node atau komputer pada jaringan. Pada proses transaksi setiap blok yang telah melakukan transaksi akan disimpan pada Blockchain dan setiap transaksi tersebut terdapat nilai hash yang didapatkan dari nilai hash blok sebelumnya kemudian dimasukan ke dalam blok, selanjutnya untuk menghitung nilai hash-nya yang baru, sehingga hash tersebut dapat dianggap sebagai pointer atau penghubung dari setiap blok tersebut. Namun, untuk nilai hash yang didapat harus memenuhi persyaratan tertentu yang disebut dengan difficulty supaya mendapat blok yang valid. Proses pencarian hash yang menghasilkan blok yang valid disebut juga sebagai PoW (*Proof of Work*).

Selain untuk menghasilkan blok yang valid, hash juga berfungsi sebagai identitas unik yang dimiliki oleh setiap blok, dan bermanfaat untuk menjaga integritas data supaya tidak mudah diubah. Apabila terdapat sedikit saja perubahan data pada blok, nilai hash pada blok tersebut akan berubah dan mempengaruhi nilai previous hash pada blok-blok selanjutnya, karena Blockchain menerapkan fungsi algoritma hash seperti SHA-256 yang telah dibahas pada penjelasan sebelumnya. Seperti yang diilustrasikan pada Gambar 2.3, dalam blokchain selalu diawali dengan blok ke-0 atau yang disebut dengan genesis block, sedangkan untuk menghasilkan blok ke-1 dan seterusnya atau blok baru, diperlukanlah miner. Istilah miner dalam Blockchain merupakan suatu pihak khusus yang memvalidasi transaksi dan menyimpan hasil transaksi pada Blockchain. Miner akan melakukan proses mining dengan menggunakan peralatan komputasi hash untuk menghasilkan blok baru. Meskipun ketika sebuah blok baru berhasil dihasilkan oleh miner, terkadang juga memungkinkan apabila terdapat miner lain yang menghasilkan blok baru secara bersamaan akan membuat sebuah fork atau percabangan pada Blockchain [16].

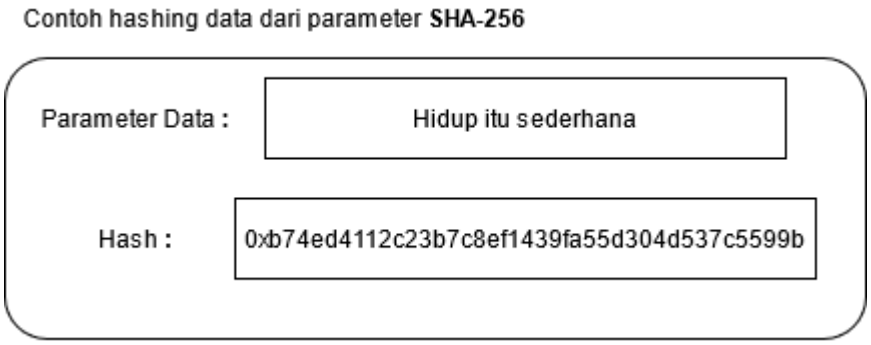


Gambar 2.0.3 Diagram Skema Blockchain

2.2.7 Fungsi Hash

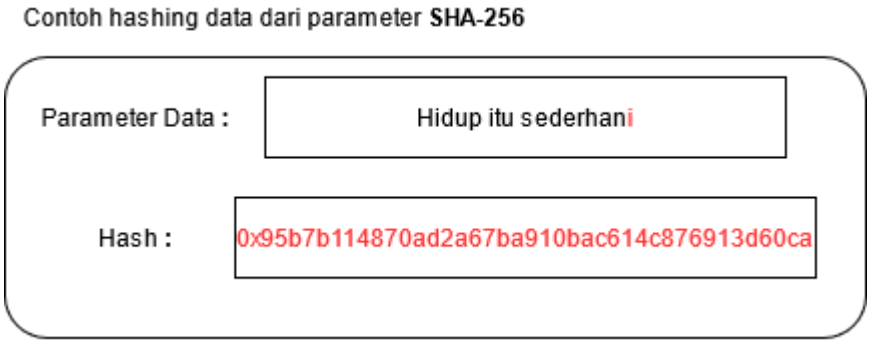
Hashing merupakan proses untuk mengubah segala data input menjadi data output yang terdiri susunan karakter acak dengan panjang yang telah ditentukan, dan dapat ditentukan sebagai karakter yang unik terhadap masing-masing data yang telah diproses. Berapa pun panjang string yang dimasukan, hasil keluaran data tersebut memiliki panjang yang tetap, dan juga proses hashing memastikan bahwa apabila terdapat sedikit perubahan pada data yang telah dimasuka akan mengubah dan mempengaruhi hasil data output tersebut.

Sebagai contoh dalam Blockchain diterapkan salah satu algoritma hash yaitu SHA-256 yang akan melakukan hashing terdapat data input menjadi data output dengan panjang 256 bit atau 64 karakter, sehingga kevalidan dan integritas data akan tetap terjaga dan tidak mudah dimanipulasi seperti pada Gambar 2.4 dan Gambar 2.5, yang mendemonstrasikan hasil hashing terhadap data input dan output apabila terdapat perubahan pada data aslinya.



Gambar 2.0.4 Fungsi Hash sebelum dtambah parameter

Gambar Data yang dilakukan dengan hashing SHA-256



Gambar 2.0.5 Fungsi Hash setelah ditambah parameter

Gambar: Perubahan pada data yang dilakukan dengan hashing SHA-256

Selain itu hash pada Blockchain juga dijadikan sebagai pointer atau penghubung antar blok dan digunakan untuk menghasilkan dan memvalidasi blok baru seperti yang sudah dijelaskan pada bagian sebelumnya, sedangkan pada Ethereum juga menggunakan teknik hashing yang disebut dengan Keccak256 dengan cara yang rumit dan melakukan hash pada setiap transaksi yang terjadi, dari proses hash tersebut juga menghasilkan transaction hash atau sebagai bukti bahwa pihak tertentu telah melakukan transaksi melalui Blockchain Ethereum.

2.2.8 Pemanfaatan Teknologi Blockchain

Telah diketahui dari pembahasan perkembangan teknologi Blockchain, khususnya dimulai dari era Blokchain 2.0, di mana teknologi Blockchain pada kenyataannya sudah mulai diimplementasikan pada beberapa sektor selain bidang keuangan, dan juga pada bidang lain seperti kesehatan, industri, hukum, dan lainnya untuk memberikan solusi dalam pelayanan yang berkaitan dengan integritas atau keaslian data [19]. Berikut terdapat beberapa contoh pemanfaataan dari teknologi Blockchain adalah sebagai berikut:

a. Bidang hukum tentu dapat menerapkan teknologi Blockchain terutama saat dibutuhkan pada proses peradilan, dengan memanfaatkan Blockchain, informasi mengenai barang bukti tetap terjaga integritasnya dan mencegah adanya pemalsuan data kasus.
b. Bidang kesehatan juga dapat menerapkan Blockchain yang dapat diimplementasikan terutama terkait dengan kepentingan kerahasiaan data riwayat kesehatan pasien melalui rekam medis elektronik.
c. Bidang rantai persediaan seperti Walmart yang merupakan perusahaan dari Amerika Serikat yang mengoperasikan jaringan departement store. Bekerjasama dengan IBM, Walmart telah mengimplementasikan Blockchain sebagai bagian dari persyaratan keamanan pangan baru untuk para pemasoknya. Hal tersebut didasari supaya Walmart dapat melacak informasi bahan pangan dari pertanian ke toko dalam waktu dekat, dengan menggunakan sistem ledger terdistribusi Blockchain dan menghindari pemalsuan maupun kerugian lainnya.
d. Bidang keuangan terkait dengan melakukan transaksi mata uang digital (cryptocurrency) yang bisa dilakukan pada berbagai platform apa pun dengan syarat harus dapat terhubung dengan jaringan internet. Contoh penerapannya seperti Bitcoin, Litecoin, dan Ripple.

2.3 Ethereum

Ethereum (ETH) pertama kali diluncurkan pada tahun 2013 oleh salah satu pengembangnya, Vitalik Buterin, seorang penulis dan programmer di komunitas Bitcoin. Menurut definisi, Ethereum adalah implementasi dari blockchain, yang memperkenalkan kekuatan komputasi untuk membangun kembali penggunaan blockchain. Blockchain hanya dapat mengubah mata uang digital menjadi transaksi nilai, terutama antar pengguna melalui bahasa skrip yang melewati perdagangan aset digital [9]. Latar belakang ditemukannya Ethereum adalah pada saat itu, karena banyaknya transaksi bernilai rendah berdasarkan penerapan blockchain Bitcoin, terjadi perdebatan tentang jaringan blockchain yang "membengkak" [19]. Kekhawatiran utamanya adalah bahwa setiap aplikasi yang dibangun dengan protokol Bitcoin, akan memiliki permasalahan utama yaitu dalam masalah penskalaan volume transaksi yang terjadi, karena Bitcoin tidak dibangun untuk menangani jumlah transaksi yang dibutuhkan oleh setiap aplikasi yang dibangun. Berdasarkan permasalahantersebut Vitalik dan rekan-rekannya melihat hal tersebut sebagai sebuah kesempatan supayaorang-orang dapat membangun aplikasi terdesentralisasi dalam Blockchain, sehingga dari hal tersebut dikembangkanlah sebuah platform Blockchain baru yang bernama Ethereum. Menurut latar belakang tersebut Ethereum dapat dianggap sebagai pengembangan dari Bitcoin yang mampu membangun aplikasi berbasis teknologi Blockchain.

Secara komponennya Ethereum memiliki dua komponen penting yaitu prosesor virtual turing (Turing-complete virtual processor) yang disebut sebagai Ethereum Virtual Machine(EVM) yang memungkinkan prosesor Turing dalam Ethereum tersebut menjalankan script atau bahasa pemrograman yang disebut Solidity untuk membangun aplikasi terdesentralisasi dan juga nilai token yang disebut dengan Ether, supaya dapat digunakan sebagai mata uang atau cryptocurrency yang disahkan oleh jaringan untuk melakukan transaksi antar pengguna atau sebagai kompensasi bagi para miner [13] , sehingga dapat terlihat jelas, perbandingan antara Bitcoin dan Ethereum. Bitcoin merupakan implementasi dari Blockchain yang bersifat public dan memiliki batasan terhadap bahasa script untuk membangun sebuah aplikasi berbasis Blockchain. Namun, dengan Ethereum pengembang aplikasi dapat lebih fleksibel untuk mengembangkan aplikasi berbasis

Blockchain yang dapat diatur berdasarkan konsensusnya maupun jenis Blockchain yang digunakan yaitu private atau public Blockchain [7].

2.3.1 Penyimpanan Variabel pada Ethereum

Ketika sebuah smart contract dideploy di Ethereum Network, mereka akan memesan memori-memori yang dibutuhkan oleh seluruh variabel yang akan dipakai oleh smart contract tersebut. Besar kecilnya memori ditentukan oleh tipe data apa yang digunakan. Semakin besar tipe datanya, maka akan semakin besar memori yang harus dipesan oleh Ethereum. [35] Tetapi Ethereum akan selalu memesan memori sebesar bytes32 seberapa kecil apapun besar variabel yang akan digunakan. Hal ini dilakukan untuk memastikan bahwa seluruh memori memiliki standarisasi tersendiri sehingga lebih mudah untuk melakukan proses pencarian data. Sebagai contoh misalkan sebuah smart contract memiliki sebuah variabel boolean yang hanya mengembalikan nilai benar atau salah. Seharusnya variabel tersebut memiliki besar uint4, tetapi Ethereum akan memesan memori sebesar bytes32 untuk variabel tersebut. Setiap besaran memori yang dipesan oleh Ethereum memiliki harga masing-masing. Didasari hal inilah maka penulis untuk mengoptimasi penggunaan memori dan harga yang digunakan pada smart contract [5].

2.3.2 Optimasi pada Ethereum pada lokasi variabel

Pada penelitian oleh Chen Yi-Cyuan, penulis lebih menjurus kepada penempatan deklarasi lokasi variable. Hal tersebut didasari oleh sistem penyimpanan dari Ethereum sendiri, yaitu setiap bytes32 memori, ia akan menyimpan sebuah nilai. Walaupun nilai tersebut memiliki besar yang lebih kecil dibandingkan dengan bytes32, Ethereum akan tetap memesan memori sebesar bytes32. Didasarkan hal tersebut, maka letak atau susunan deklarasi variabel sangatlah penting. Pada penelitian dikatakan bahwa sangat penting untuk mengetahui setiap variabel memiliki besar berapa, sehingga kita sebagai programmer dapat mengoptimisasi penggunaan memori pada smart contractnya. Misalkan dengan meletakkan variabel-variabel, yang jika ditambahkan besar memorinya tidak melebihi bytes32, dekat satu sama lain. Jika menggunakan mode tersebut, maka tidak akan ada memori yang hilang atau disia-siakan. Sehingga seluruh memori yang dipesan dapat digunakan dengan sebaik-baiknya [12].

2.3.3 Optimasi yang menggabungkan beberapa variabel pada suatu memori

Peneliti Lucas membuktikan bahwa penggabungan variabel yang ditotal masih belum mengisi memori bytes32 dapat mengurangi memori yang digunakan pada smart contract. Hal ini didasari karena sistem penyimpanan variabel pada Ethereum yang selalu memesan bytes32 memori. [14] Sehingga jika kita menggabungkan beberapa variabel menjadi sebuah bytes32 memori, maka akan menambah efisiensi memori pada smart contract [5].

2.3.4 Akun pada Ethereum

Akun pada Ethereum merupakan hal yang mendasari bagaimana cara kerja Blockchain Ethereum. Akun-akun tersebut digunakan untuk menyimpan dan menelusuri informasi pengguna dalam jaringan. Menurut [20] ada platform Ethereum terdapat dua jenis akun, yaitu:

a. *User Account* (*externally owned accounts*) merupakan akun yang dimiliki oleh pengguna pada Ethereum. Ketika akun pengguna Ethereum dibuat, pada saat itu juga menghasilkan sebuah kunci public dan kunci private. Kunci private disimpan aman secara individu, dan kunci public digunakan sebagai alamat akun atau identitas User account. Kunci public umumnya terdiri dari 256 karakter. Namun, untuk Ethereum hanya menggunakan 42 karakter pertama untuk mewakili identitas akun, dan ditulis dalam bentuk hexadesimal seperti: "**0x19afA8C970AaB2D3a24F42d85244e8756d23293a**" Akun ini juga sering disebut sebagai akun eksternal yang menyimpan nilai (balance) dalam satuan Ether. Karena kedudukan akun ini bersifat eksternal, akun ini dapat melakukan transaksi dengan menjalankan fungsi yang sudah didefenisikan pada kontrak.

b. *Contract Accounts* (*contract address*) terkait pembahasan pada Ethereum, akun kontrak merupakan jenis akun yang mirip dengan *User account* (*externally owned accounts*), karena dapat diidentifikasi dengan menggunakan kunci publik (public key) namun, tidak memiliki private key. Sebuah akun kontrak memiliki nilai Ether yang mirip dengan akun pengguna atau User account yang digunakan apabila terdapat fungsi yang dijalankan, karena pada akun kontrak memiliki script kode untuk menjalankan fungsi dan variable seperti yang dituliskan pada *smart contract* yang digunakan.

2.3.5 *Smart contract*

Perkembangan blockchain hingga saat ini ternyata sudah banyak dimanfaatkan dan diterapkan pada berbagai bidang seperti yang telah dibahas sebelumnya. Supaya pihak pengembang dapat menerapkan aplikasi yang dibangun berbasis Blockchain, solusi yang tepat adalah dengan menggunakan smart contract. Kontrak cerdas atau smart contract merupakan penerapan dari platform Blockchain yang memiliki tujuan untuk menentukan kesepakatan (consensus) antara beberapa pihak berdasarkan jenis konsensus yang digunakan dan diaplikasikan
pihak sensus penerapan yang dalam berbentuk skrip atau kode sebagai logika bisnis yang terkait dalam penggunaan sistem atau aplikasi berbasis teknologi Blockchain [19].

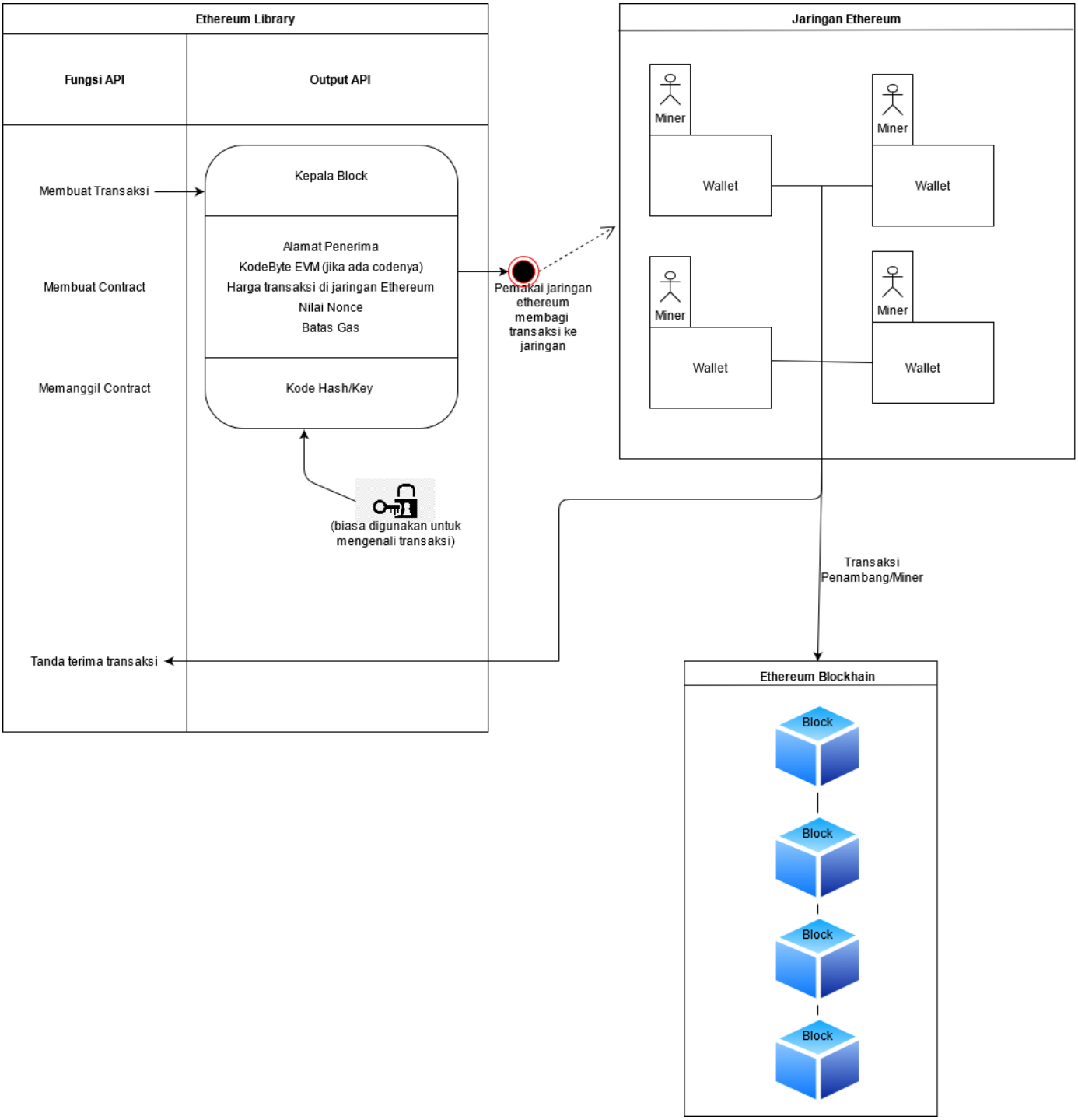
Implementasi smart contract tentunya dapat dibangun sesuai dengan kebutuhan yang diinginkan dan digunakan secara aktif melalui platform Blockchain manapun seperti Ethereum dengan menggunakan bahasa pemrograman yang bernama Solidity. Untuk merancang *smart contract*, serta melakukan transaksi tanpa adanya pihak ketiga dengan menggunakan satuan Ether di jaringan Ethereum, seperti yang terlihat pada Gambar 2.6, dengan adanya kesepakatan dalam bentuk *smart contract* seharusnya dapat meningkatkan transparansi serta kepercayaan terhadap aplikasi Blockchain bagi penggunanya.

2.3.6 Cara Kerja Ethereum

Penerapan dari arsitektur Blockchain terhadap Ethereum tentunya memiliki persamaan arsitektur yang terdiri dari berbagai komponen yang saling berinteraksi dan bekerja sama untuk menjalankan fungsinya. Terkait dengan bagaimana Ethereum bekerja dalam Blockchain, tentunya terdapat beberapa komponen penting di balik proses tersebut. Komponen Ethereum yang dimaksud meliputi Ethereum Virtual Machine (EVM), miner, blok, mining, Ether, dan gas [13]. Seperti yang telah dijelaskan sebelumnya bahwa jaringan Blockchain terdiri dari sekumpulan node yang dimiliki oleh para miner (penambang) dan juga beberapa node yang tidak dimiliki oleh siapa pun namun tetap membantu pengaplikasian smart contract..

Proses eksekusi terhadap smart contract dan transaksi tersebut dijalankan pada Ethereum Virtual Machine (EVM) yang merupakan perangkat Turing complete

yang berjalan pada jaringan Ethereum seperti yang digambarkan pada gambar dibawah, EVM juga digunakan sebagai tempat penyimpanan bagi *smart contract* untuk dapat membantu dalam memperluas Ethereum dengan menuliskan fungsionalitas atau logika bisnis dalam pengembangan aplikasi berbasis Blockchain. Tahapan berikutnya adalah *smart contract* dapat dieksekusi sebagai bagian dari transaksi dan dalam proses mining, sedangkan Ether digunakan sebagai satuan cryptocurrency dalam jaringan Ethereum yang memiliki fungsi seperti halnya untuk melakukan transaksi dari satu akun ke akun yang lain atau juga dapat digunakan dalam menjalankan fungsi-fungsi yang telah ditetapkan pada *smart contract*.



Gambar 2.0.6 Skema Cara Kerja Jaringan Ethereum

Selain itu juga terdapat komponen yang disebut dengan gas, yaitu sebuah bagian internal yang menjaga nilai Ethereum atau merupakan bentuk biaya untuk melakukan transaksi secara mikro supaya dapat menjaga proses komputasi pada Blockchain. Gas akan dibayarkan ketika terjadi proses operasi eksekusi pada fungsi yang ada pada *smart contract*. Terkait dengan berapa jumlah besaran gas yang

dibayarkan pada setiap melakukan eksekusi sebuah kodeterbilang sangat kecil karena mengadopsi sifat transaksi mikro. Apabila gas tersebut habis, pengguna tentu tidak dapat melanjutkan operasi transaksi. Dalam proses transaksi menggunakan Ethereum, transaksi juga harus ditandatangani secara digital dengan menggunakan private key pemilik akun. Hal tersebut untuk memastikan bahwa identitas pengirim dapat digunakan pada saat melakukan proses verifikasi akun [20].

2.3.7 Penambang/Miner

Sebuah proses transaksi pasti membuat sebuah hub blok yang akan ditambahkan kedalam Jaringan Blockchain Ethereum, dan sebuah transaksi juga memerlukan waktu dan spesifikasi tinggi dari sebuah barang VGA yang merupakan hasil dari daya komputasi perangkat lunak sebagai mekanisme konsensus bukti pengerjaan, maka hal ini suka disebut dengan Miners/Penambang [18].

Para mineworkers ini juga yang mengumpulkan beberapa transaksi yang belum terkonfirmasi, lalu melakukan komputasi yang telah ditetapkan oleh arrange agar dapat dimasukkan kedalam blockchain. Ketika komputasi telah selesai, maka kumpulan transaksi akan terkonfirmasi dan dimasukkan ke dalam blockchain. Mineworker juga secara terintegrasi memberitahu kepada diggers lainnya tentang kumpulan transaksi yang telah terkonfirmasi agar mereka juga tahu bahwa ia telah menyelesaikan tugasnya. Proses inilah yang mengakibatkan para pemakai smart contract untuk membayar Gas Cost hasil komputasi berikut kepada mineworkers dengan transaksi-transaksi yang akan dikonfirmasi.

2.3.8 Gas

Untuk membayar biaya transaksi pada Jaringan Ethereum, diperlukan tarif yang harus dibayarkan oleh pengirim transaksi. Hal itu adalah gas, yaitu salah satu satuan pada Ethereum untuk mengukur penggunaan komputasi pada transaksi. Pengeluaran biaya transaksi tentu dapat diubah-ubah, sesuai kebutuhan transaksi, tetapi jika ternyata gas yang dibutuhkan ternyata lebih banyak dibandingkan yang disediakan, maka transaksi akan gagal dan tidak dapat diterima di dalam blockchain. Nilai 1 gas adalah identik dengan 0,000001 ETH [23].

Semakin besar komputasi yang dilakukan didalam smart contract, semakin besar pula gas yang dibutuhkan untuk membayar para mineworkers untuk melakukan konfirmasi dari transaksinya. Gas cost yang diperlukan berasal dari seluruh operasi yang telah diubah menjadi bahasa mesin, dimana setiap operasi memiliki biayanya masing- masing. Konsep ini berlaku pula bila adanya memori yang bertambah, semakin besar memori yang digunakan pada smart contract, semakin besar gas yang diperlukan karena proses menyimpan pada jaringan Ethereum cukup memakan banyak biaya [15].

Adapula seperti yang Chen Yi-Cyuan, yaitu terlihat bahwa beberapa kode pada smart contract ternyata belum terlalu teroptimasi dari beberapa sisi. Hal ini tentu juga membuat besar gas yang diperlukan juga terpengaruh. Sehingga dapat disimpulkan juga bahwa semakin teroptimasinya sebuah smart contract akan membantu penghematan biaya gas pada pengembangan smart contract [11].

2.3.9 Keccak256

Keccak256 adalah algoritma hashing yang memiliki sifat oneway. Dimana hasil hashing tidak dapat didekripsi kembali ke nilai awal. Tetapi hasil dari hashing dapat dipastikan memiliki hasil yang sama jika nilai awalnya adalah sama persis. Sebenarnya banyak algoritma hashing diluar sana, contohnya adalah sha256 dan MD5 seperti yang sudah dijelaskan pada struktur blockchain sebelumnya, tetapi tim Ethereum memutuskan untuk menggunakan algoritma hashing keccak256 pada ethereum network. Keccak256 adalah sebuah fungsi yang akan mengambil beberapa parameter dengan tipe data apapun dan melakukan hashing one-way yang akan membentuk sebuah kata yang unik dari parameter yang diinput. Urutan dari parameter juga berpengaruh pada keunikan dari data yang dihashing. Hasil dari hashing tersebut ialah sebuah kata unik yang tidak beraturan yang memiliki 66 huruf termasuk "0x" sebagai penanda bahwa kata tersebut adalah sebuah hexadesimal [17]. Ethereum menggunakan keccak256 karena sangat cocok dengan kinerja jaringan Ethereum dalam hal penyimpanan data pada *smart contract*. Sebuah slot penyimpanan pada Ethereum juga berupa 64 karakter [14].

2.3.10 ERC-780

ERC merupakan kepanjangan dari "*Ethereum Request for Comment*" sedangkan *780* merupakan proposal dari identitas tersebut yang pada umumnya *ERC-20* yang merupakan keoriginalan yang dipakai secara umum di jaringan Ethereum.

Mengapa *ERC-780*?, dikarenakan *780* ini yang dibuat oleh tim Ethereum yang di desain untuk membuktikan subjek apapun dalam tampilan antarmuka *smart contract.* Dengan kata lain, ini memungkinkan Anda untuk mengetes subjek tertentu didalam ekosistem jaringan blockchain Ethereum Anda juga dapat mengklaim atau memverifikasi alamat Anda sendiri. (Thornstensson, 2017). Maka dari ini saya akan mencoba metode keccak256 melalui proposal jaringan Ethereum ERC-780.

2.4 Sublime Text

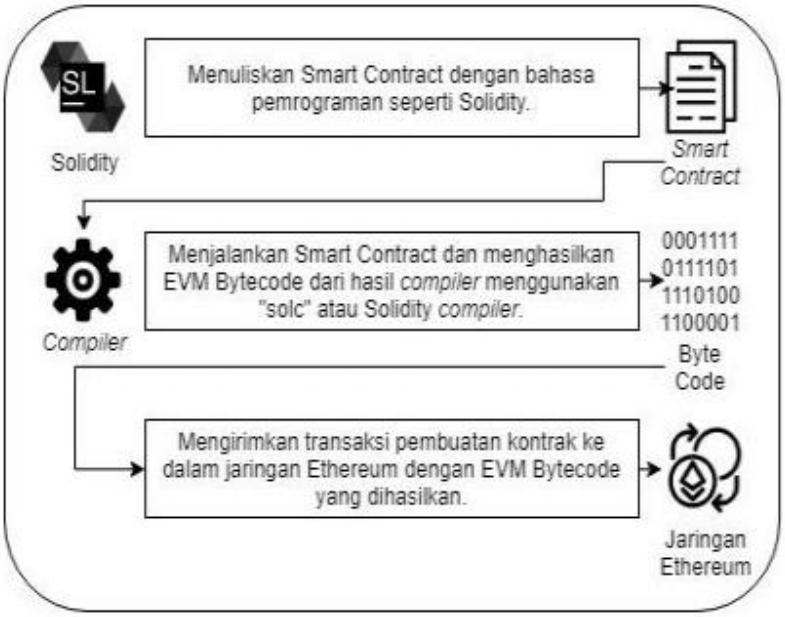
Sublime Text adalah editor kode sumber lintas platform shareware dengan antarmuka pemrograman aplikasi (API) Python. Ini secara native mendukung banyak bahasa pemrograman dan bahasa markup, dan fungsi dapat ditambahkan oleh pengguna dengan plugin, biasanya dibuat oleh komunitas dan dipelihara di bawah lisensi perangkat lunak bebas.

2.5 Bahasa Pemrograman

Bahasa program adalah sekumpulan instruksi yang diberikan kepada komputer untuk dapat melaksanakan tugas-tugas tertentu dalam menyelesaikan suatu permasalahan. Bahasa program berfungsi untuk memerintah komputer agar dapat mengolah data sesuai dengan langkah-langkah penyelesaian yang telah ditentukan oleh programmer. Bahasa ini memungkinkan seorang programmer untuk menentukan secara persis data mana yang akan diolah oleh komputer, bagaimana data ini akan disimpan/diteruskan, dan jenis langkah apa yang akan diambil dalam berbagai situasi secara persis [33].

2.5.1 Solidity

Solidity merupakan bahasa pemrograman berorientasi objek (kontrak) yang memiliki tujuan untuk merancang *smart contract* supaya dapat berjalan pada Ethereum Virtual Machine (EVM), serta disimpan dalam sebuah file dengan ekstensi (.sol). Segala kode yang dituliskan dalam bahasa pemrograman Solidity akan dikompilasi menggunakan Solidity compiler atau biasa disebut dengan "solc" yang menghasilkan bytecode (sekumpulan fungsi yang telah diencode), supaya dapat dijalankan dan dieksekusi pada EVM seperti yang digambarkan pada Gambar 2.7



Gambar 2.0.7 Proses Compile dan Deploy Kontrak

Pada proses encoding dengan compiler supaya menghasilkan Bytecode yang digunakan sebagai referensi fungsi dan kontrak untuk diekseskusi pada EVM. Proses encoding tersebut dibantu dengan menggunakan ABI (Application Binary Interface) yang merupakan daftar definisi fungsi dalam kontrak dan beberapa argumen yang ditulis dalam format Javascript Object Notation (JSON). Daftar fungsi dan argument tersebut diubah dengan hash menjadi ABI, kemudian dapat diolah oleh EVM. ABI sangat diperlukan supaya dapat menentukan fungsi mana yang ada pada kontrak untuk dijalankan, serta menjamin fungsi tersebut akan mengembalikan data dalam format yang sudah ditentukan [13].

2.5.2 Web3.js Library

Web3.js atau javascript merupakan bahasa pemrograman yang digunakan dalam pengembangan website agar lebih dinamis dan interaktif. Untuk mempermudah berinteraksi dengan Ethereum network, maka diperlukan web3 sebagai kumpulan library javascript yang dapat membuat program berinteraksi dengan blockchain Ethereum. Ini mewakili pengikatan bahasa JavaScript untuk antarmuka JSON RPC Ethereum, yang membuatnya dapat digunakan secara langsung dalam teknologi web, karena JavaScript secara native didukung di hampir semua browser web. Web3.js juga biasa digunakan di sisi server di aplikasi Node.js [28].

2.5.2.1 Node Js
Karena menggunakan blockchain pribadi atau lokal yang berjalan, kami perlu mengonfigurasi aplikasi untuk mengembangkan *smart contract* [9]. Untuk itu, akan memerlukan Node Package Manager atau NPM, yang mencakup Node Js.
2.5.2.2 React Js
React Js merupakan sebuah front-end library JavaScript yang di buat oleh facebook. React adalah library yang bersifat composable user interface, yang artinya kita dapat membuat berbagai UI yang bisa dibagi menjadi beberapa komponen yang menjadikan Node js sebagai servernya [22]. Untuk itu disini saya menggunakannya sebagai tampilan front end yang cepat dan efisien.

Table 2.0.6 Javascript

2.6 HTML & CSS

Kepanjangan dari HTML adalah hypertext markup language, dan HTML bertujuan untuk menampilkan konten pada web. HTML merupakan suatu teknologi yang penting karena semua website menggunakan HTML untuk menampilkan kontennya.

Kepanjangan dari CSS adalah cascading style sheet, dan CSS bertujuan untuk membuat tampilan web lebih menarik. CSS adalah suatu teknologi yang penting karena semua website menggunakan CSS untuk membuatnya lebih menarik.

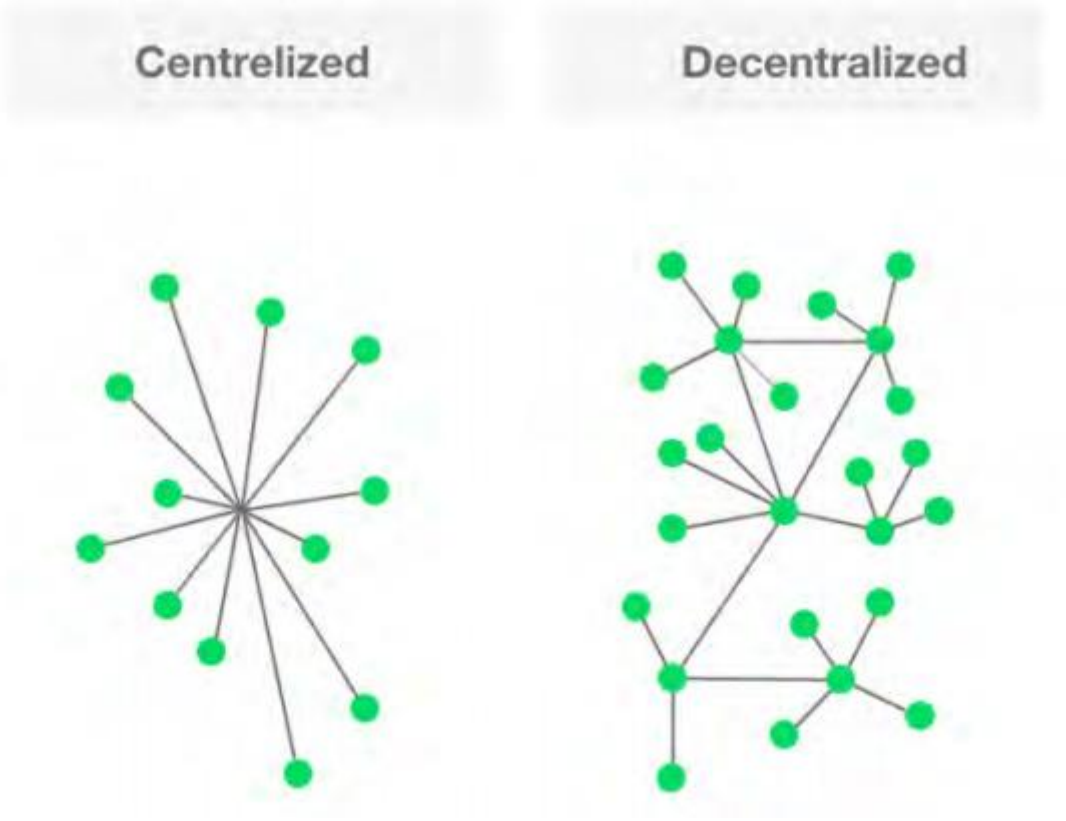
Di satu sisi, HTML bisa bekerja secara maksimal dengan dua bahasa frontend: CSS (Cascading Style Sheets) dan JavaScript. Jika digabungkan, kedua bahasa frontend ini bisa meningkatkan pengalaman user dan mengaktifkan fungsi yang lebih canggih.

- CSS erat kaitannya dengan styling, seperti background, warna, layout, spacing, dan animasi.
- JavaScript memungkinkan Anda menambahkan fungsionalitas yang dinamis, seperti slider, pop-up, dan galeri foto.

Kira-kira gambaran sederhana perbedaan HTML dan CSS dan JavaScript seperti ini: HTML adalah orang yang tidak memakai satu helai benang pun, CSS adalah bajunya, dan JavaScript adalah aktivitas dan sikapnya [3].

2.7 Decentralized Application (DApp)

Decentralized Application (DApp) merujuk ke aplikasi yang dijalankan oleh banyak pengguna melalui jaringan blockchain yang terdesentralisasi [36]. Ethereum blockchain memberikan kemampuan komputasi dan penyimpanan melalui mekanisme *smart contract*. Oleh karena itu, Ethereum DApps dapat menggunakan *smart contract* yang disediakan oleh Ethereum untuk mengimplementasikan logika bisnis [36].



Gambar 2.0.8 Jaringan Centralized dan Decentralized

2.8 Ganache

Ganache adalah jaringan blockchain pribadi, yang merupakan blockchain pengembangan server lokal yang dapat digunakan untuk bertindak seperti blockchain publik [8]. Ganache digunakan untuk deploy *smart contract* dan untuk menjalankan tes. Ganache menyediakan 10 akun dengan 100 Ethereum untuk menguji *smart contract* yang ada di blockchain lokal.

2.9 Truffle Framework

Truffle Framework merupakan tool penting untuk mengembangkan sebuah *smart contract* yang nantinya diunggah ke dalam jaringan blockchain Ethereum. Karena di dalam framework Ini menggunakan bahasa pemrograman Solidity untuk mengembangkan *smart contract*. Kerangka kerja truffle menyediakan beberapa fungsi untuk menunjang pembuatan *smart contract*, diantaranya sebagai berikut [8]:

1. Client Side Development
2. Script Runner
3. Network Management

4. Development Console
5. Deployment & Migrations
6. *Smart contract* Management
7. Automated Testing

2.10 Metamask

Karena browser saat ini belum mendukung koneksi langsung ke jaringan blockchain. maka perlu ditambahkan *extension* metamask ke browser, mengonversi browser biasa menjadi browser blockchain. Selain itu juga bisa menghubungkan jaringan ganache untuk mengembangkan aplikasi. Metamask juga bisa digunakan untuk mengelola akun pribadi Ethereum, yang digunakan untuk terhubung ke blockchain dan tujuan transaksi [8].

MetaMask adalah *extension* atau plugin yang memungkinkan untuk mengunjungi web versi DApp di browser. Hal ini memungkinkan untuk menjalankan aplikasi Terdesentralisasi Ethereum tepat di browser Anda tanpa menjalankan blockchain Ethereum penuh. MetaMask juga menyertakan brankas identitas yang aman, menyediakan antarmuka pengguna untuk mengelola identitas pengguna di berbagai situs dan menandatangani transaksi blockchain [8].

2.11 Diagrams.net

Diagrams.net adalah tumpukan teknologi open source untuk membangun aplikasi diagram, dan perangkat lunak diagram pengguna akhir berbasis browser yang paling banyak digunakan di dunia.

2.12 Unified Modeling Language

UML atau Unified Model Language adalah sebuah bahasa pemodelan yang dapat digunakan pada pembuatan software. Ini adalah bahasa pemodelan general yang dapat digunakan untuk mendesain sebuah sistem atau aplikasi. Dengan adanya model sebuah system, maka programmer dan designer dapat sepakat apa aplikasi/sistem yang ingin dibuat. UML juga dapat digunakan untuk menunjukan sistem yang sedang dirancang kepada stakeholder, sehingga orang-orang terpenting yang terlibat dalam pembuatan aplikasi/sistem dapat mengerti apa yang sedang dibuat.

2.12.1 Use Case Diagram

Use-case diagram merupakan model diagram UML yang digunakan untuk menggambarkan requirement fungsional yang diharapkan dari sebuah sistem. Usecase diagram adalah diagram usecase yang digunakan untuk menggambarkan secara ringkas siapa yang menggunakan sistem dan apa saja yang bisa dilakukannya. Use case class digunakan untuk memodelkan dan menyatakan unit fungsi/layanan yang disediakan oleh sistem (or bagian sistem: subsistem atau class) ke pemakai. Diagram use case tidak menjelaskan secara detail tentang penggunaan usecase, namun hanya memberi gambaran singkat hubungan antara usecase, aktor, dan sistem.

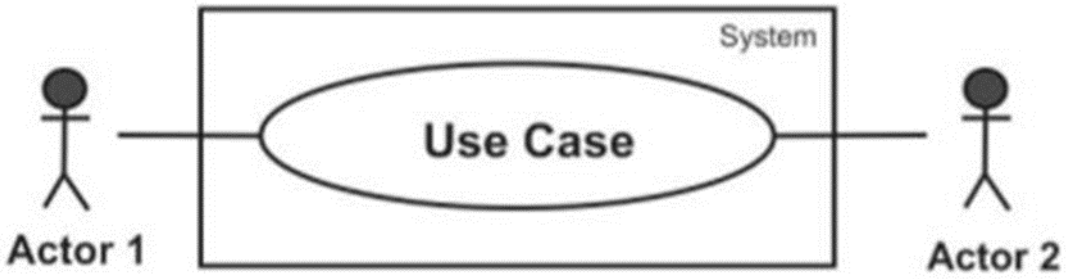
Table berikut merupakan elemen-elemen dari use case diagram:

Actor :	Mempresentasikan seseorang atau sesuatu (seperti perangkat,sistem lain) yang berinteraksi dengan sistem. Actor hanya berinteraksi dengan use case tetapi tidak memiliki kontrol atas use case.
Use Case :	Adalah gambaran fungsionalitas dari suatu sistem, sehingga customer atau pengguna sistem paham dan mengerti mengenai kegunaan sistem yang akan dibangun.
Association :	Menghubungkan link antar element.
	Yaitu kelakuan yang harus terpenuhi agar sebuah event dapat terjadi, dimana pada

<<Include>> :	kondisi ini sebuah use case adalah bagian dari use case lainnya.

Table 2.0.7 Elemen-elemen pada use case diagram

Berikut merupakan contoh gambar use case diagram:



Gambar 2.0.9 Contoh use case diagram

2.12.2 Activity Diagram

Diagram aktivitas adalah bentuk visual dari alir kerja yang berisi aktivitas dan tindakan, yang juga dapat berisi pilihan, pengulangan, dan concurrency. Dalam Unified Modeling Language, diagram aktivitas dibuat untuk menjelaskan aktivitas komputer maupun alur aktivitas dalam organisasi. Diagram aktivitas menggambarkan alur kontrol secara garis besar.

Diagram aktivitas memiliki komponen dengan bentuk tertentu, dihubungan dengan tanda panah. Panah mengarahkan urutan aktivitas terjadi, dari awal sampai akhir.

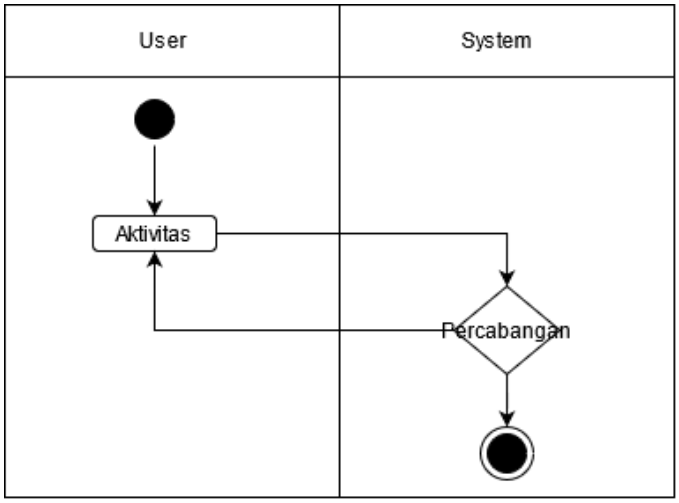
Diagram aktivitas dapat dianggap sebagai jenis alir kerja. Umumnya alir kerja tidak memiliki cara untuk menampilkan concurrency. Simbol penggabungan dan pemecahan pada diagram aktivitas dapat menjadi solusi untuk pemakaian yang sederhana.

Berikut merupakan table dari elemen-elemen activity diagram:

Simbol	Nama	Keterangan
	Status awal	Sebuah diagram aktivitas memiliki sebuah status awal.
	Aktivitas	Aktivitas yang dilakukan sistem,aktivitas biasanya diawali dengan kata kerja.
	Percabangan / Decision	Percabangan dimana ada pilihan aktivitas yang lebih dari satu.
	Penggabungan / Join	Penggabungan dimana yang mana lebih dari satu aktivitas lalu digabungkan jadi satu.
	Status Akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
	Swimlane	Swimlane memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

Table 2.0.8 Elemen-elemen pada activity diagram

Berikut merupakan contoh gambar dari activity diagram

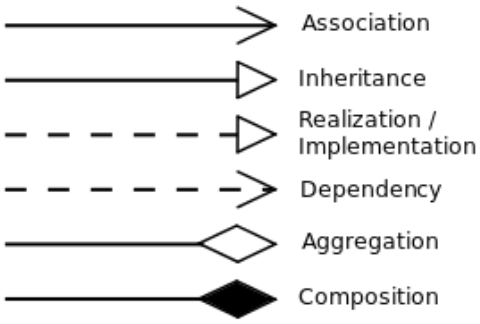


Gambar 2.0.10 Activity Diagram

2.12.3 Class Diagram

Class diagram disebut jenis diagram struktur karena menggambarkan apa yang harus ada dalam sistem yang dimodelkan dengan berbagai komponen. Berbagai komponen tersebut dapat mewakili class yang akan diprogram, objek utama, atau interaksi antara class dan objek.

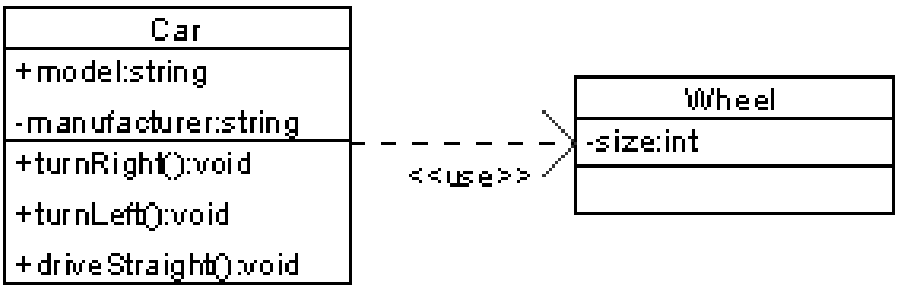
Gambar berikut merupakan elemen-elemen dari class diagram :



0	Tidak ada kejadian (jarang)
0.1	Tidak ada contoh, atau satu contoh
1	Tepat satu contoh
1.1	Tepat satu contoh
0..*	Nol atau lebih banyak instance
*	Nol atau lebih banyak instance
1..*	Satu atau lebih contoh

Gambar 2.0.11 Elemen Class Diagram

Gambar berikut merupakan contoh dari class diagram :



Gambar 2.0.12 Class Diagram

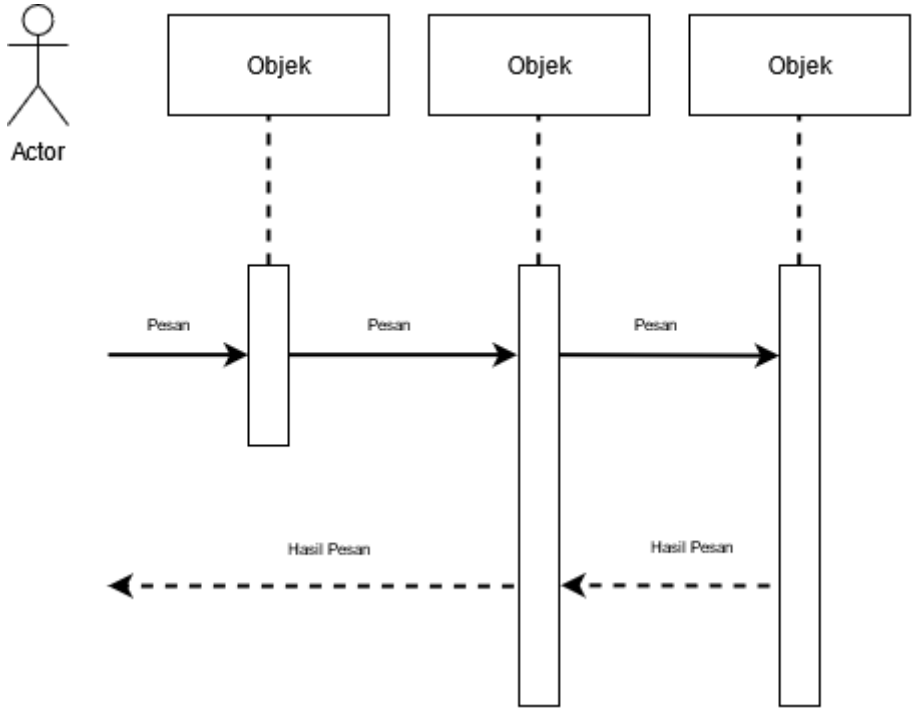
Sumber : https://en.m.wikipedia.org/wiki/Class_diagram

2.12.4 Sequence Diagram

Sequence diagram adalah diagram yang menggambarkan kolaborasi dinamis antara sejumlah object. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara object juga interaksi antara object. Sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem.

Dalam sequence diagram, setiap object hanya memiliki garis yang digambarkan garis putus-putus ke bawah. Pesan antar object digambarkan dengan anak panah dari object yang mengirimkan pesan ke object yang menerima pesan.

Berikut merupakan contoh gambar dari sequence diagram:



Gambar 2.0.13 Sequence Diagram

2.13 Extreme programming

Extreme programming adalah sebuah teknik pembuatan software yang mementingkan code dibandingkan dokumentasi. Untuk membuat hal tersebut terjadi, orang yang menggunakan teknik extreme programming akan lebih mementingkan aktifitas code dibandingkan menulis dokumentasi yang lengkap. Setelah membuat software tersebut dengan cepat, software tersebut akan ditest dan apabila software tersebut telah berjalan dengan baik, akan diberikan ke user untuk mendapatkan feedback.

2.14 White Box Testing

Dalam percobaan ini akan lebih cocok menggunakan white box testing karena pihak yang melakukannya lebih ke software developer, hal ini dikarenakan white box testing bisa mengetahui code dan semua struktur internal sebuah program. Hal ini karena dalam white box testing, saya perlu mengetahui apakah ada komponen dari software yang harus diperbaiki atau tidak. White box testing juga perlu membutuhkan pengetahuan teknis karena memang pengujiannya lebih berfokus ke arah sana [1].