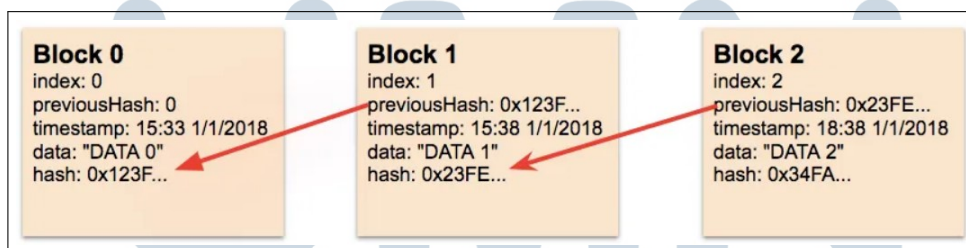


BAB 2 LANDASAN TEORI

2.1 Blockchain

Teknologi *blockchain* merupakan struktur data yang membentuk sebuah buku besar digital seperti sebuah *database* yang menyimpan seluruh rekaman transaksi secara terdistribusi dan terdesentralisasi serta divalidasi oleh seluruh komputer atau *node* yang berperan sebagai validator di dalam jaringan [7]. Struktur data pada *blockchain* terdiri dari serangkaian *block* yang saling terhubung dan setiap *block* berisi kumpulan data transaksi [8]. Setiap *block* pada *blockchain* menyimpan *hash* dan *hash* dari *block* sebelumnya yang disebut sebagai *previous hash*.

Pada jaringan *blockchain*, setiap *block* terhubung satu sama lain melalui *previous hash*. Sehingga ketika data pada suatu *block* diubah, maka *hash* pada *block* tersebut akan berubah dan *previous hash* yang disimpan pada *block* berikutnya akan berubah menjadi tidak valid. Hal ini juga menyebabkan *blockchain* bersifat *immutable* atau tidak dapat diubah [9]. Selain itu, rangkaian *block* tersebut didistribusikan ke seluruh *node* dalam jaringan *blockchain*. Hal ini meminimalkan *single point of failure*, karena setiap *node* memiliki salinan penuh dari rangkaian *block* tersebut [10].



Gambar 2.1. Hash yang Terhubung pada Setiap Block

Sumber: [8]

Blockchain menerapkan *consensus* yang dilakukan oleh seluruh komputer atau *node* dalam jaringan untuk melakukan validasi terhadap *block* yang akan dimasukkan ke dalam jaringan *blockchain* sehingga keamanan dan integritas data di dalam jaringan *blockchain* dapat dipastikan melalui proses validasi dengan *consensus* [11]. Terdapat beberapa jenis *consensus* yang dapat digunakan, namun *proof of work* dan *proof of stake* merupakan dua jenis *consensus* yang paling populer [12].

2.2 Ethereum

Ethereum merupakan jaringan *blockchain* dengan sistem *smart contract* yang dapat digunakan dalam berbagai kasus selain sebagai mata uang transaksi (ETH) [13]. *Smart contract* pada *Ethereum* dibangun dengan bahasa pemrograman *turing-complete* yaitu *Solidity*. Salah satu komponen penting dalam *Ethereum* yaitu *Ethereum Virtual Machine* (EVM) yang merupakan sebuah mesin virtual yang berjalan secara terdesentralisasi untuk melakukan komputasi dan mengeksekusi kode program *smart contract* pada jaringan *blockchain* [14]. Terdapat beberapa jaringan *blockchain* lainnya yang mendukung kompatibilitas dengan EVM yang disebut sebagai *EVM compatible blockchain*. Sehingga jaringan *blockchain* tersebut dapat menerapkan fitur *smart contract* seperti *Ethereum*. Contoh *EVM compatible blockchain* yang populer saat ini adalah *Binance Smart Chain*, *Avalanche*, *Tron*, *Arbitrum*, dan *Polygon* [15].

2.3 Polygon

Polygon merupakan salah satu jenis jaringan *EVM compatible blockchain* yang ditujukan untuk mengoptimalkan jaringan *Ethereum* dalam pemakaian *gas fee* atau biaya transaksi yang tinggi [16]. Jenis *consensus* yang digunakan oleh *Polygon* dalam melakukan validasi pada jaringan *blockchain* adalah *proof of stake*. Berbeda dengan *Ethereum* yang menggunakan *proof of work*, *proof of stake* pada *Polygon* dapat mempercepat proses transaksi sehingga mengurangi *gas fee* seminimal mungkin [17].

2.4 Smart Contract

Smart contract merupakan sebuah kode program yang disimpan di dalam jaringan *blockchain* dan dijalankan secara otomatis ketika suatu kondisi terpenuhi. *Smart contract* yang sudah dikembangkan, akan dilakukan *deployment* yaitu mengirimkan *smart contract* ke dalam jaringan *blockchain* sehingga *smart contract* tersebut disebarkan ke setiap *node* yang tergabung pada jaringan tersebut. Aplikasi atau sistem yang mengintegrasikan teknologi *blockchain* dapat memanfaatkan *smart contract* sebagai *server-side (back-end)*. Setiap operasi yang dilakukan melalui *smart contract* menghasilkan perubahan status pada jaringan *blockchain*, akan menghasilkan suatu transaksi [18]. Riwayat transaksi tersebut tercatat dan

tersimpan di dalam jaringan *blockchain*, sehingga mengubah status jaringan tersebut [19].

2.5 Gas

Pengoperasian *smart contract* yang menghasilkan transaksi atau mengubah status pada jaringan *blockchain* akan dikenakan biaya transaksi atau *gas fee* berdasarkan jumlah satuan *gas* yang digunakan (*gas used*) untuk mengeksekusi operasi pada suatu fungsi *smart contract*. *Gas* adalah satuan pengukuran jumlah sumber daya komputasi yang dibutuhkan untuk mengeksekusi suatu operasi dalam jaringan *blockchain*. Pengolahan transaksi oleh *node* pada jaringan memerlukan sumber daya komputasi yang cukup, sehingga pengguna harus membayar *gas fee* setiap kali menjalankan suatu operasi sebagai kompensasi atas penggunaan sumber daya tersebut. Konsep ini berperan dalam menjaga keamanan dan kestabilan pada jaringan, dengan membebankan *gas fee* untuk setiap operasi sehingga, dapat mencegah *denial of service*. Selain itu, batasan pada *gas used* mencegah pemborosan sumber daya komputasi dalam suatu transaksi [20].

Jumlah *gas* yang dibutuhkan bergantung pada kompleksitas operasi yang dijalankan. Semakin rumit operasi tersebut, semakin besar *gas* yang diperlukan, sehingga total *gas fee* yang dikenakan akan semakin besar untuk mengeksekusi fungsi operasi dari *smart contract* tersebut. Sedangkan, *gas fee* ditentukan oleh ukuran *block* yang digunakan untuk menyimpan hasil transaksi tersebut. Semakin besar kapasitas *block* yang digunakan, semakin tinggi biaya satuan *gas* yang ditetapkan. Ukuran *block* dapat berubah tergantung pada *traffic* pada jaringan *blockchain* [21]. Kecepatan penyelesaian operasi tidak dipengaruhi oleh kompleksitas fungsi itu sendiri, melainkan oleh *node* pada jaringan *blockchain* yang memproses hasil transaksi tersebut [22].

2.6 Account

Account merupakan komponen penting dalam jaringan *Ethereum* sebagai pengidentifikasi pengguna dalam bentuk *address* dengan ukuran 20-byte. Dengan *account*, pengguna dapat melakukan transaksi antar pengguna yang berbeda atau berinteraksi dengan *smart contract* yang tersimpan pada jaringan *blockchain*. Pengguna dapat membuat suatu *account* dan menggunakannya untuk melakukan suatu transaksi melalui sebuah *wallet* seperti *metamask* [23]. Terdapat dua jenis

account yaitu *Externally Owned Account* (EoA) dan *Contract Account* (sebuah akun pada *smart contract*) [13]. Berikut adalah penjelasan dan perbandingan kedua jenis akun dalam bentuk tabel:

Tabel 2.1. Perbedaan EoA dan Contract Account

Perbedaan	EoA	Contract Account
Definisi	Akun yang dimiliki oleh pengguna jaringan untuk melakukan transaksi antar pengguna ataupun <i>smart contract</i> .	Akun yang menyimpan kode program <i>smart contract</i> dan berjalan secara otomatis ketika suatu kondisi yang telah ditetapkan terpenuhi.
Private Keys	Setiap pengguna memiliki <i>private key</i> yang memberikan akses penuh terhadap <i>account</i> tersebut.	Tidak ada pengguna yang memiliki <i>private key</i> untuk akses penuh terhadap <i>smart contract</i> .
Yang Dimiliki	Memiliki <i>balance</i> atau saldo	Memiliki <i>balance</i> atau saldo dan kode program.
Pembuatan Akun	Pembuatan EoA tidak memungut biaya (<i>gas fee</i>)	Pembuatan <i>contract account</i> memungut biaya (<i>gas fee</i>) karena membutuhkan penyimpanan pada jaringan <i>Ethereum</i> .
Menjalankan Transaksi	Bisa memulai suatu transaksi karena memiliki akses sepenuhnya terhadap akun.	<i>Contract Account</i> tidak bisa memulai suatu transaksi, hanya bisa menjalankan transaksi sebagai bentuk respon ketika suatu kondisi terpenuhi.
Jenis Transaksi	EoA dapat menghasilkan transaksi berupa mengirim dan menerima saldo serta memanggil fungsi pada <i>smart contract</i> .	<i>Contract Account</i> dapat menghasilkan lebih banyak jenis transaksi berupa memanggil fungsi <i>smart contract</i> lain, <i>contract deployment</i> , dan sebagainya.

Tabel 2.2. Perbedaan EoA dan Contract Account (Lanjutan)

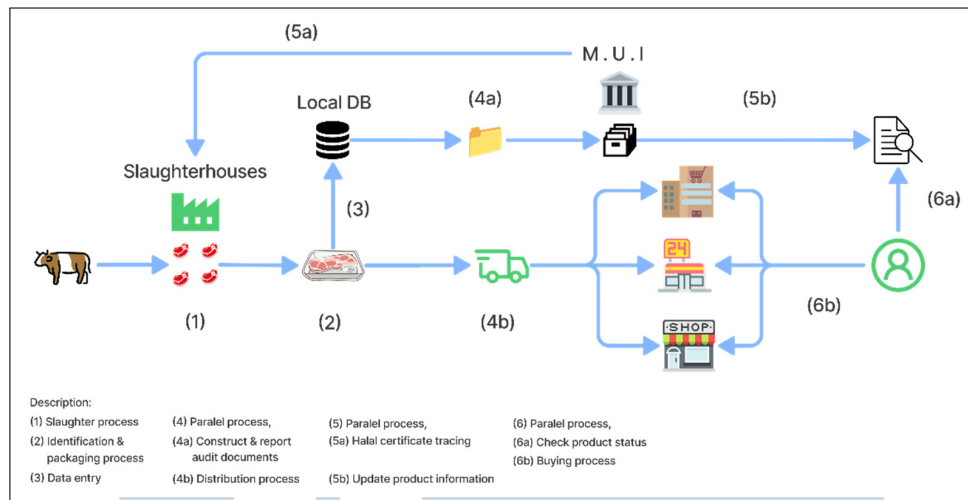
Perbedaan	EoA	Contract Account
Kepemilikan	Setiap pengguna pada jaringan <i>Ethereum</i> memiliki EoA yang berfungsi sebagai akun.	<i>Contract account</i> tidak dimiliki oleh siapapun, hanya berjalan secara terdesentralisasi dalam jaringan <i>blockchain</i> .

2.7 Decentralized Application

Decentralized application (DApp) merupakan aplikasi yang berjalan di atas jaringan *blockchain* tanpa adanya perantara terpusat. DApp memanfaatkan jaringan *blockchain* sebagai penyimpanan data melalui penggunaan *smart contract* sebagai *backend* yang terhubung dengan jaringan *blockchain* [24]. Dalam penggunaan DApp, pengguna memerlukan *crypto wallet* untuk dapat mengakses sebuah fitur DApp yang memanggil fungsi dari *smart contract* sehingga dapat melakukan pembayaran berupa *gas fee* untuk menjalankan sebuah transaksi. Selain berinteraksi terhadap *smart contract*, *crypto wallet* memungkinkan pengguna untuk melakukan transaksi antar pengguna lain melalui *crypto wallet* pengguna tersebut. Setiap transaksi yang dilakukan pada *crypto wallet* akan tercatat dan tersimpan pada jaringan *blockchain*. Salah satu contoh *crypto wallet* yang sangat populer adalah *MetaMask* [25].

2.8 Supply Chain

Supply chain management merupakan mekanisme yang mengoptimalkan waktu, lokasi dan aliran kuantitas bahan untuk meningkatkan produktivitas rantai suplai suatu perusahaan. *Supply chain management* bertujuan dalam meningkatkan efektivitas dan efisiensi dari suatu proses sehingga menciptakan koordinasi yang baik antara *stakeholders* yang dapat menghindari kerugian yang cukup besar pada suatu perusahaan. Dalam jaringan *supply chain management*, proses tersebut ditunjukkan dalam bentuk rantai yang panjang dimana terdapat keterlibatan setiap *stakeholders* seperti *supplier*, *slaughterhouses*, *distributor*, *store*, dan *customer* dalam jaringan tersebut [26].



Gambar 2.2. Alur Proses *Supply Chain Management*

Sumber: [26]

2.9 Sistem Jaminan Halal

Sistem Jaminan Halal (SJH) merupakan sistem yang menjamin kehalalan produk oleh suatu perusahaan, telah memenuhi persyaratan kehalalan dengan ketentuan yang telah ditetapkan oleh LPPOM MUI [27]. Regulasi yang digunakan oleh LPPOM MUI dalam sistem jaminan halal adalah HAS 23000 yang berisi 11 kriteria SJH yang wajib dipenuhi untuk mendapatkan sertifikat halal. Berikut 11 kriteria SJH pada HAS 23000:

1. Kebijakan Halal

Hal - hal tertulis yang harus ditetapkan dan disosialisasikan oleh manajemen puncak perusahaan kepada seluruh pihak (stakeholder) perusahaan untuk menghasilkan produk halal secara konsisten.

2. Tim Manajemen Halal

Sekelompok anggota yang ditetapkan oleh manajemen puncak perusahaan dengan bukti tertulis, bertanggung jawab terhadap perencanaan, implementasi, evaluasi, dan perbaikan SJH di perusahaan.

3. Pelatihan

Pelatihan merupakan kegiatan untuk meningkatkan pengetahuan, keterampilan, dan sikap guna mencapai tingkat kompetensi yang diinginkan.

Pelatihan ini wajib diadakan oleh perusahaan dan dilaksanakan minimal setahun sekali.

4. Bahan

Bahan yang harus diperhatikan dalam jaminan kehalalan produk meliputi bahan baku, bahan tambahan, bahan penolong, kemasan, pelumas, sanitizer, dan bahan pembersih. Bahan tersebut dapat dibagi menjadi bahan tidak kritis yaitu bahan yang terdapat dalam daftar positif halal dan bahan kritis. Bahan kritis merupakan bahan yang berstatus tidak halal atau haram dan harus dilengkapi dokumen pendukung yang memadai.

5. Fasilitas Produksi

Fasilitas produksi seperti bangunan, ruangan, mesin, dan peralatan harus memenuhi persyaratan yang ditetapkan.

6. Produk

Dilarang menciptakan produk yang memiliki rasa atau aroma yang mengarah pada sesuatu yang diharamkan, serta kemasan dan label produk tidak boleh memiliki deskripsi bersifat erotis atau porno.

7. Prosedur Tertulis Aktivitas Kritis

Perusahaan harus memiliki prosedur tertulis yaitu SOP (*Standard Operating Procedure*) untuk menjalankan aktivitas kritis. Aktivitas kritis merupakan aktivitas yang dapat mempengaruhi status kehalalan produk misalnya penggunaan bahan baru untuk produk yang sudah disertifikasi, transportasi bahan, penyimpanan bahan, dan sebagainya.

8. Kemampuan Telusur

Produk harus dapat dilacak atau ditelusuri kembali hingga bahan yang disetujui oleh LPPOM MUI dan diproduksi di fasilitas yang memenuhi kriteria fasilitas.

9. Penanganan Produk yang Tidak Memenuhi Kriteria

Perusahaan harus memiliki prosedur tertulis mengenai produk yang tidak memenuhi kriteria. Produk tersebut harus dimusnahkan dan tidak boleh diproses ulang serta tidak boleh dijual ke konsumen yang membutuhkan produk halal. Produk yang sudah dijual namun tidak memenuhi kriteria harus ditarik kembali dari pasar.

10. Audit Internal

Perusahaan wajib memiliki prosedur tertulis mengenai audit internal pelaksanaan SJH. Audit internal dilakukan minimal satu kali dalam setahun.

11. Kaji Ulang Manajemen

Perusahaan harus memiliki prosedur tertulis mengenai pengkajian ulang manajemen yang dilakukan minimal sekali dalam setahun.

