BAB 2
TINJAUAN PUSTAKA

2.1 Blockchain

Teknologi *blockchain* merupakan struktur data yang membentuk sebuah buku besar digital dan mendistribusikan buku besar tersebut melalui jaringan [11]. *Blockchain* dikembangkan pertama kali oleh Stuart Haber dan W Scott Storneta pada tahun 1998. Kemudian *blockchain* dikembangkan lagi oleh Satoshi Nakamoto dalam paper nya berjudul "*A Peer to Peer Electronic Cash System*" dengan mengimplementasikannya pada salah satu cryptocurrency yaitu *Bitcoin* sebagai pembayaran elektronik pada jaringan *peer to peer* yang bersifat terdesentralisasi [12].

Dalam *paper* yang berjudul "*Blockchain: Powering the Internet of Value*" menjelaskan bahwa, *blockchain* pada dasarnya merupakan *database* yang berisi data riwayat transaksi seperti catatan atau *logs* yang terdistribusi pada semua *node* yang tergabung pada jaringan tersebut. Namun, terdapat suatu perbedaan pada *blockchain* dan *database* pada umumnya. Perbedaan tersebut terletak pada sifat *immutable* yang dimiliki oleh teknologi *blockchain*, dimana data yang tercatat dan tersimpan pada suatu *block*, saling terhubung dengan *block* yang lain pada jaringan tersebut dan tidak dapat berubah [13]. Selain itu, *blockchain* juga menerapkan *consensus* atau kesepakatan bersama dalam memasukkan data ke setiap *block* yang ada dalam jaringan *blockchain* tersebut. Hal ini bertujuan agar data yang tersimpan dalam *blockchain* tidak ada yang tidak valid [14].

Penginputan data ke dalam *blockchain*, dilakukan melalui lima tahapan yaitu:

- Proses transaksi merupakan tahapan dimana pengguna akan melakukan transaksi dan transaksi tersebut terkirim melalui jaringan *blockchain*.
- Autentikasi transaksi merupakan tahapan dimana akan dilakukan *digital signature*. Setelah itu transaksi tersebut akan diletakkan pada *transactions pool*. *Transactions pool* merupakan wadah yang mengumpulkan data-data transaksi dari berbagai pengguna yang belum dimasukkan ke dalam suatu *block*.
- *Creating block* merupakan tahapan yang dimana beberapa transaksi dari

transactions pool dikumpulkan dan dimasukkan ke suatu *block* yang kemudian akan dikirim ke jaringan *blockchain* untuk divalidasi.

- Validasi *block*, merupakan tahapan dimana *block* akan divalidasi oleh *node*. Teknik memvalidasi *block* pada setiap jaringan *blockchain* berbeda-beda tergantung consensus yang ditetapkan oleh jaringan tersebut. Sebuah situs yang bernama Investopedia.com menyebutkan terdapat lima *consensus* yang saat ini digunakan yaitu *proof of work*, *proof of stake*, *proof of capacity*, *proof of activity*, dan *proof of burn*. Namun *proof of work* merupakan consensus yang saat ini umum digunakan dalam memvalidasi suatu *block* [15].
- *Block chaining* merupakan tahapan ketika *block* sudah divalidasi, maka akan ditambahkan ke dalam *blockchain* yang kemudian isi buku besar tersebut akan disebarkan pada jaringan sehingga, setiap *node* memiliki data dengan isi data yang sama.

Proof of work merupakan mekanisme kesepakatan bersama yang dilakukan oleh *node* yaitu *miner* untuk memvalidasi suatu *block* dan menambahkan ke *blockchain*. Dalam mekanisme *proof of work*, para *node* yang telah menjadi *miner* akan mencari *hash* kriptografi dalam bentuk SHA-256 yang valid, untuk menjadikan *block* tersebut valid sehingga dapat ditambahkan kedalam jaringan *blockchain*. *Hash* yang valid terbentuk melalui proses komputasi yang melibatkan *nonce*, *timestamp*, *previous hash*, dan *merkle root* atau kumpulan transaksi. *Hash* yang telah didapatkan melalui proses *proof of work*, akan digunakan kembali untuk pemvalidasian pada *block* selanjutnya yang dimana *hash* tersebut akan menjadi *previous hash* yang digunakan untuk mendapatkan *hash* yang valid pada *block* selanjutnya [16].

Keamanan pada *blockchain* terletak pada konsep kriptografi *hash* khususnya pada *previous hash* yang saling terhubung antar *block* dalam jaringan. *Hashing* merupakan metode kriptografi yang mengubah isi data kedalam suatu bentuk yang unik yang disebut sebagai *hash*. *Hash* pada setiap *block* selain mengamankan suatu informasi, juga berperan sebagai identitas pada *block* tersebut. *Hash* yang dihasilkan memiliki beberapa sifat yaitu:

1. *Deterministic*

   Deterministic merupakan salah satu sifat *hash* yang terjadi ketika terdapat suatu kalimat ataupun data, jika diberi metode kriptografi berulang kali, *hash* yang dihasilkan akan tetap sama. Sebagai contoh:

- Terdapat suatu data ‘kota’ yang diberi *hashing* menjadi “0XAKDOIBJKDIEOFF”, ketika diberi metode tersebut berulang kali hasilnya akan tetap sama menjadi “0XAKDOIBJKDIEOFF”.

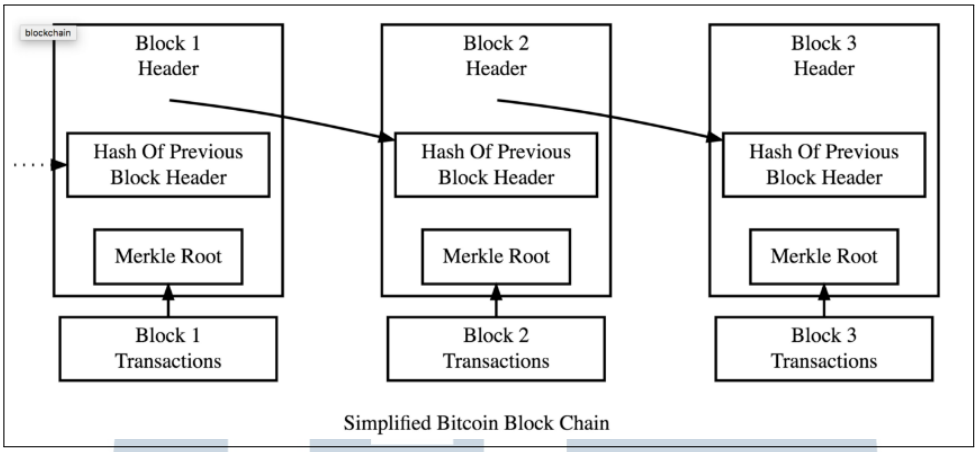
2. *Collision Resistant*
   Collision Resistant merupakan merupakan salah satu sifat *hash* yang menunjukkan bahwa tidak terdapat hasil *hash* yang sama dengan satu *hash* yang lainnya sehingga setiap hash yang dihasilkan akan selalu *unique*.

3. *Fast Computation*
   Fast Computation merupakan merupakan salah satu sifat *hash* yang menunjukkan bahwa tidak memerlukan banyak sumber daya komputasi untuk memproses suatu fungsi *hash*.

4. *Avalance Effect*
   Avalance Effect merupakan merupakan salah satu sifat *hash* yang menunjukkan jika terdapat perubahan pada data maka hasil *hash* yang dihasilkan akan berbeda dari yang sebelumnya. Sebagai contoh:

   - Terdapat suatu data yaitu ‘kota’ dengan hash “0XAKDOIBJKDIEOFF” kemudian data tersebut diubah menjadi ‘kotak’ yang menyebabkan *hash* yang dihasilkan akan berubah secara menyeluruh menjadi “0XAKDOIBJKDEEEEE”.

Pada jaringan *blockchain*, *block* sebelumnya terhubung dengan *block* selanjutnya melalui *previous hash* atau *hash* yang sudah divalidasi pada *block* sebelumnya. Sehingga ketika data dalam *block* sebelumnya diubah maka *block* selanjutnya hingga *block* seterusnya akan berubah menjadi tidak valid dikarenakan terhubung dengan *previous hash* dan karakteristik yang dimiliki oleh *hash* [17].



Gambar 2.1. Hash yang Terhubung Pada Setiap Block

Sumber: [18]

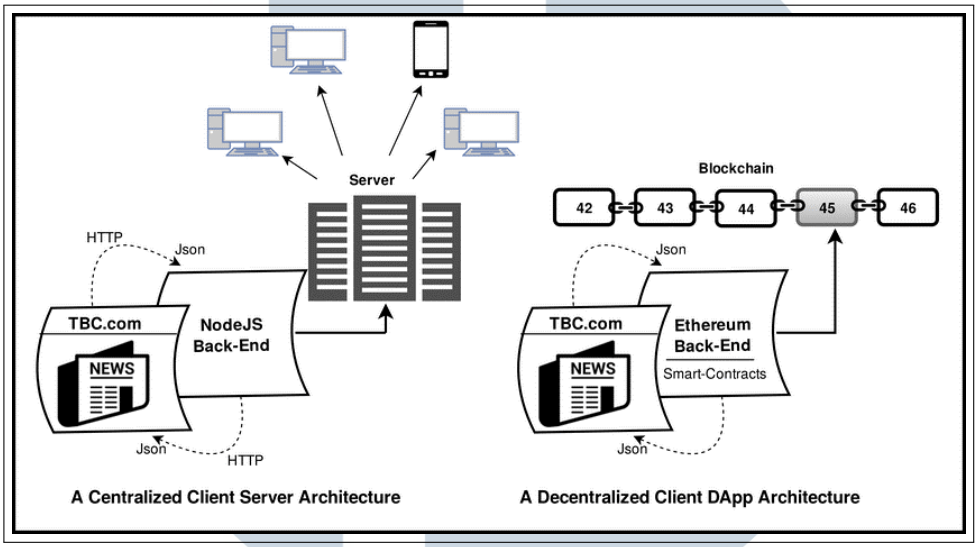
2.2 Ethereum

Ethereum merupakan salah satu jenis jaringan *blockchain* seperti *bitcoin* namun memiliki fitur tambahan yaitu *smart contract* yang digunakan untuk memverifikasi dan mengeksekusi kode aplikasi pada jaringan *blockchain*. *Ethereum* dibangun dengan bahasa pemograman *turing-complete* untuk membangun suatu *smart contract* sehingga kode atau fungsi program yang ditulis pada *smart contract* dapat berjalan diatas jaringan *Ethereum* [19]. Kode *smart contract* dieksekusi melalui sebuah *compiler machine* yang disebut sebagai *Ethereum Virtual Machine* atau disingkat EVM.

2.2.1 Smart Contract

Smart contract merupakan program sebagai *microservice* yang disimpan dalam blockchain dan berjalan ketika suatu kondisi terpenuhi. *Smart contract* merupakan suatu program kecil yang tersimpan dalam *blockchain* dan akan berjalan secara automatis apabila syarat atau kondisi yang telah diatur tercapai [20]. *Smart contract* pertama kali ditemukan oleh Nick Szabo pada tahun 1990 [21]. *Smart contract* dapat menjalankan tindakan seperti mengirimkan pemberitahuan, menambahkan data, menghapus data, dan mengubah data yang tersimpan pada smart contract yang kemudian setiap riwayat transaksi yang dilakukan akan dicatat ke *blockchain* [22]. Sehingga aplikasi atau sistem yang mengintegrasikan jaringan *blockchain* dapat menggunakan *smart contract* untuk mengolah data layaknya

server-side (back-end). Sebelum *smart contract* dapat digunakan oleh suatu sistem atau aplikasi, *smart contract* akan dilakukan *deployment* yaitu mengirimkan *smart contract* ke jaringan *blockchain* sehingga *smart contract* tersebut akan tersebar di setiap *node* yang tergabung pada jaringan tersebut [23].



Gambar 2.2. Arsitektur Penggunaan Smart Contract

Sumber: [23]

2.2.2 Ethereum Virtual Machine

Ethereum Virtual Machine (EVM) adalah sebuah mesin virtual yang dapat melakukan komputasi dan mengeksekusi perintah secara terdesentralisasi. EVM merupakan salah satu bagian terpenting dari *blockchain Ethereum*, dimana seluruh transaksi eksekusi *smart contract* yang tersimpan pada *blockchain Ethereum* dijalankan pada EVM. (Eiki, 2019). EVM mengeliminasi beberapa faktor penghambat, sehingga pengembang tidak harus memiliki perangkat keras dengan spesifikasi tinggi untuk membangun suatu sistem yang terintegrasi dengan jaringan *blockchain*. EVM tidak hanya dapat digunakan pada jaringan *Ethereum*, melainkan dapat digunakan oleh jaringan *blockchain* lainnya. *Blockchain* yang dapat menggunakan EVM disebut sebagai *EVM compatible blockchain* sehingga jaringan *blockchain* tersebut dapat menerapkan fitur *smart contract* seperti *Ethereum*. Beberapa contoh EVM compatible blockchain adalah *Binance Smart Chain*, *Avalanche*, *Tron*, *Polygon*, dan yang lainnya. (GoCrypto, 2022).

2.2.3 Gas

Setiap proses komputasi yang memproses suatu transaksi melalui *smart contract* yang terhubung dengan jaringan *blockchain*, akan dieksekusi oleh *node miner*. *Miner* pada jaringan *blockchain* yang telah mengeksekusi transaksi tersebut dengan memvalidasi transaksi dan memasukkan transaksi tersebut ke jaringan *blockchain* melalu suatu *consensus*, akan diberi bayaran sebagai *reward* berupa *gas* yang telah dibayar oleh pengirim transaksi. *Gas* merupakan biaya pemrosesan yang dibayar oleh user selaku pengirim transaksi berdasarkan unit khusus dalam *Ethereum* [24]. Sehingga pengguna sistem akan terus membiayai transaksi ketika melakukan request yang berhubungan dengan *smart contract*. *Gas* merupakan komponen utama dalam keamanan aplikasi atau sistem yang mengintegrasikan jaringan *EVM compatible blockchain*. Penggunaan *gas* atau biaya transaksi dapat meminimalisir terjadinya serangan *denial of service* pada layanan sistem dikarenakan setiap kali *user* melakukan *request* ke *smart contract*, akan dikenakan biaya transaksi atau *gas* [25].

2.2.4 Account

Account atau akun merupakan komponen yang penting dalam jaringan *Ethereum*. *Account* mengindentifikasikan pengguna dalam bentuk *address* dengan ukuran 20-byte. *Account* memampukan pengguna untuk melakukan transaksi antar pengguna yang berbeda ataupun berinteraksi dengan *smart contract* pada jaringan *Ethereum*. Pada jaringan *Ethereum*, terdapat dua jenis *account* yaitu *Externally Owned Account* (EoA) dan *Contract Account* (*Smart Contract Account*) [19]. Adapun penjelasan dan perbandingan yang dijabarkan dalam bentuk tabel sebagai berikut.

Tabel 2.1. Perbedaan EoA dan Contract Account

Perbedaan	EoA	Contract Account
Definisi	Akun yang dimiliki oleh pengguna jaringan *Ethereum*.	Akun yang menyimpan beberapa kode program *smart contract* yang akan dijalankan ketika suatu kondisi terpenuhi.
Private Keys	*Private key* dimiliki oleh pengguna, sehingga pengguna tersebut memiliki akses penuh terhadap akun tersebut.	*Private key* tidak dimiliki siapapun, sehingga tidak ada pihak yang memiliki akses penuh terhadap suatu *smart contract* ketika sudah melakukan *deployment*.
Yang Dimiliki	Memiliki *balance* atau saldo	Memiliki *balance* dan kode program.
Pembuatan Akun	Pembuatan EoA tidak memungut biaya apapun	Pembuatan *contract account* memungut biaya karena menggunakan penyimpanan pada jaringan *Ethereum*.
Menjalankan Transaksi	EoA dapat memulai suatu transaksi karena memiliki akses sepenuhnya terhadap akun.	*Contract Account* tidak dapat memulai suatu transaksi sendiri. Ia hanya dapat menjalankan transaksi sebagai bentuk respon ketika suatu kondisi terpenuhi.
Jenis Transaksi	Jenis transaksi pada EoA dapat berupa mengirim dan menerima saldo dan memanggil fungsi pada *smart contract*.	Jenis transaksi pada *Contract Account* memiliki banyak fungsi berupa memanggil fungsi kontrak lain, *contract deployment*, dan sebagainya.
Kepemilikan	Dimiliki oleh pengguna di dunia nyata (semua pengguna pada jaringan *Ethereum*) *smart contract*.	Tidak dimiliki siapapun, hanya berjalan secara terdesentralisasi dalam jaringan *blockchain*.

2.2.5 Metamask

Metamask merupakan salah satu aplikasi *wallet* yang terhubung dengan jaringan *blockchain*. Aplikasi tersebut tersedia pada *browser* sehingga memudahkan pengguna untuk berinteraksi dengan web yang mengintegrasikan teknologi *blockchain* serta melakukan transaksi pada *smart contract* [26]. *Metamask* memiliki beberapa fitur utama sebagai berikut [27].

- *Metamask* menyediakan fitur untuk membuat *account* seperti *EoA*.
- *Metamask* memampukan dalam melakukan transaksi antar *account*.
- *Metamask* menyediakan beberapa jaringan *blockchain* seperti *Ethereum*, EVM *Compatible Blockchain* yang dapat dihubungkan dengan suatu *account* untuk melakukan suatu transaksi.
- *Metamask* memampukan dalam melakukan import atau eksport *account* melalui *private key*.
- Setiap transaksi yang dilakukan pada *Metamask* juga akan tercatat pada situs resmi pencatatan riwayat transaksi seperti *Etherscan*, *Polygonscan*, dll.

2.3 Confidentiality, Integrity, dan Availability

Dalam keamanan informasi, terdapat tiga aspek penting yaitu yaitu *Confidentiality* (kerahasiaan), *Integrity* (integritas), dan *Availability* (ketersediaan). Ketiga attribut tersebut lebih dikenal dengan sebutan CIA [28]. Masing-masing aspek memiliki penilaian yang berbeda dalam suatu keamanan informasi.

2.3.1 Confidentiality

Confidentiality merupakan salah satu aspek penting dalam keamanan informasi yang memastikan kerahasiaan pada data sehingga data hanya dapat diakses oleh orang berwenang saja [28]. Tingkat *confidentiality* akan semakin tinggi bila data yang tersimpan semakin sulit untuk diakses atau hanya orang yang berkepentingan saja yang dapat mengakses data tersebut. Sedangkan tingkat *confidentiality* akan bernilai rendah bila data yang tersimpan mudah untuk diakses oleh siapa saja [29].

2.3.2 Integrity

Integrity merupakan salah satu aspek penting dalam keamanan informasi yang menjamin keakurasian pada data sehingga memastikan data tidak berubah. Data harus terjaga keutuhan dan keakuratannya [28]. Tingkat *Integrity* akan rendah bila data tidak sesuai dengan data asli atau data tidak lengkap. Terjadinya perubahan pada data juga dapat menurunkan tingkat Integrity pada data yang tersimpan. [29].

2.3.3 Availability

Availability merupakan salah satu aspek penting dalam keamanan informasi yang memastikan bahwa data tersedia bagi siapa saja yang berwenang dan membutuhkannya [28]. Tingkat *availability* akan tinggi bila data yang tersimpan dapat diakses oleh siapa saja. Sedangkan, tingkat *availability* akan rendah bila data yang tersimpan hanya dapat diakses sebagian individu atau kelompok. Tingkat *availability* bertolak belakang dengan *confidentiality*. Bila tingkat *availability* tinggi maka tingkat *confidentiality* menjadi rendah begitu pula sebaliknya [29].

2.3.4 Base Metrics

Base metrics merupakan metode yang digunakan untuk menguji suatu keamanan. Pada pengujian tersebut, *base metrics* terbagi menjadi dua pengujian yaitu *base impacts* dan *base exploitability* [29] [30]. Berikut merupakan penjelasan mengenai dua pengujian tersebut.

2.3.5 Base Impacts

Terdapat tiga aspek yang diukur pada pengujian ini yaitu *Confidentiality Impact*, *Integrity Impact*, dan *Availability Impact*. Nilai pada setiap aspek dibagi menjadi *none*, *partial*, dan *complete*. *Confidentiality Impact* mengukur dampak potensi terhadap kerahasiaan bila eksploitasi atau penyerangan berhasil dilakukan. Pada penilaian *Confidentiality Impact*, nilai akan menjadi *none* bila tidak ada dampak terhadap kerahasiaan data. Nilai akan menjadi *partial* bila dampak berupa sebagian kerahasiaan data dapat diakses. Nilai akan menjadi *complete* bila dampak berupa kerahasiaan data dapat terlihat oleh siapa saja atau individu yang tidak memiliki akses [29] [30].

Integrity Impact mengukur dampak potensi terhadap integritas data bila eksploitasi atau penyerangan berhasil dilakukan. Pada penilaian *Integrity Impact*, nilai akan menjadi *none* bila tidak ada dampak integritas data yang tersimpan. Nilai akan menjadi *partial* bila dampak berupa ada kemungkinan perubahan integritas pada data. Nilai akan menjadi *complete* bila dampak berupa integritas pada data yang tersimpan dapat berubah sepenuhnya [29] [30].

Availability Impact mengukur dampak potensi pada ketersediaan data bila eksploitasi atau penyerangan berhasil dilakukan. Pada penilaian *Availability Impact*, nilai akan menjadi *none* bila tidak ada dampak pada ketersediaan data. Nilai akan menjadi *partial* bila dampak tersebut berupa pengurangan ketersediaan data, namun data masih dapat diperlihatkan. Nilai akan menjadi *complete* bila dampak berupa data tidak tersedia sama sekali [29] [30]. Berikut merupakan rumus yang digunakan untuk menghitung *base metrics*.

$$I = 10,41 * (1 - (1 - CI) * (1 - II) * (1 - AI)) \tag{2.1}$$

Berikut adalah *value* dari nilai yang didapatkan yang dijabarkan dalam bentuk tabel berikut ini.

Tabel 2.2. Tabel Penilaian *Base Impacts*

Nilai	Value
None	0
Partial	0,275
Complete	0,66

Sumber: [29] [30]

Base Exploitability Terdapat tiga aspek yang diukur pada pengujian ini yaitu *Access Vector*, *Authentication*, dan *Access Complexity*. *Access Vector* mengukur tingkat akses yang dibutuhkan sehingga dapat dilakukan eksploitasi atau penyerangan. Penilaian pada *Access Vector* berupa *Adjacent Network*, *Network*, dan *Local*. Nilai akan menjadi *Local* bila eksploitasi dapat dilakukan dengan mendapatkan akses sistem secara fisik. Nilai akan menjadi *Adjacent Network* bila eksploitasi dapat dilakukan dengan mengakses jaringan lokal. Nilai akan menjadi *Network* bila eksploitasi dapat dilakukan dengan mengakses di luar jaringan lokal [29] [30].

Authentication mengukur berapa autentikasi yang diperlukan untuk dapat

melakukan eksploitasi. Penilaian pada *Authentication* berupa *Multiple*, *Single*, dan *None*. Nilai akan *Multiple* bila dibutuhkan dua atau lebih authentikasi. Nilai akan *Single* bila dibutuhkan satu authentikasi. Nilai akan *None* bila tidak membutuhkan authentikasi untuk melakukan eksploitasi [29] [30].

Access Complexity mengukur tingkat kesulitan akses yang diperlukan untuk melakukan eksploitasi. Penilaian pada *Access Complexity* berupa *High*, *Medium*, dan *Low*. Nilai akan *High* bila akses yang didapatkan cukup sulit. Nilai akan *Medium* bila akses tidak mudah untuk didapatkan, namun bisa didapatkan. Nilai akan menjadi *Low* bila akses didapatkan dengan mudah [29] [30]. Berikut merupakan rumus yang digunakan untuk menghitung *base exploitability*.

$$E = 20 * AV * Au * AC \tag{2.2}$$

Berikut adalah *value* dari nilai yang didapatkan pada *base exploitability*.

AV	*Local*	0,395
	Adjacent Network	0,646
	Network	1
Au	*Multiple*	0,45
	Single	0,56
	None	0,704
AC	*High*	0,35
	Medium	0,61
	Low	0,71

Gambar 2.3. Value *Base exploitability*

Sumber: [29] [30]

2.3.6 Base Score

Ketika hasil *base exploitability* dan *base impact* telah dihitung, maka *base metrics* dapat diperoleh dengan menghitung *base score*. Berikut merupakan rumus perhitungan *Base Score*.

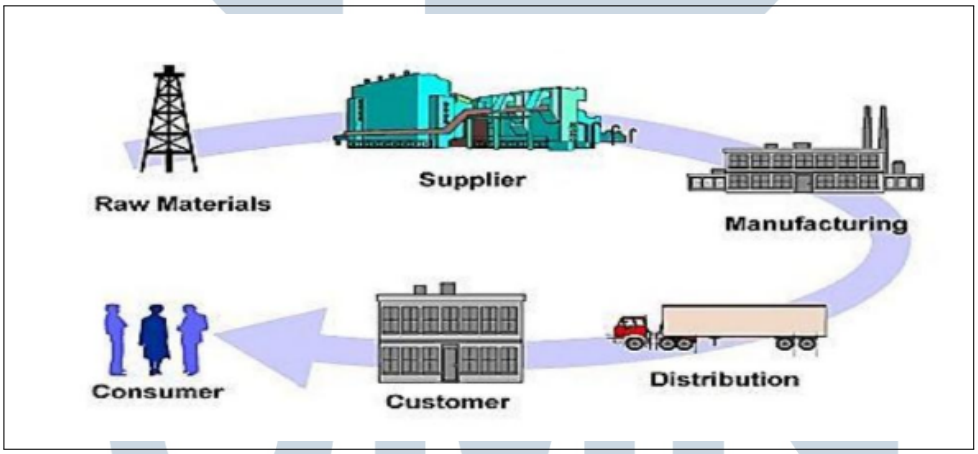
$$BaseScore = round\_1\_to\_dec(((0,6 * 1) + (0,4 * E) - 1,5) * f(Impact)) \tag{2.3}$$

Nilai pada *f(impact)* didapatkan melalui hasil penilaian pada *base metrics*.

Bila hasil *base metrics* bernilai 0, maka *f(impact)* akan bernilai 0. Jika *base metrics* tidak bernilai 0, maka *f(impact)* akan bernilai 1,176 [29] [30].

2.4 Supply Chain Management

Supply chain management merupakan mekanisme yang melibatkan optimalisasi 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. [31]. Istilah *supply chain management* pertama kali dikemukakan oleh Oliver dan Weber pada tahun 1982. Dalam jaringan *supply chain management*, proses tersebut ditunjukkan dalam bentuk rantai yang panjang dimana terdapat keterlibatan setiap *stakeholders* dalam jaringan tersebut [32].



Gambar 2.4. Alur Proses *Supply Chain Management*

Sumber: [31]

Berikut adalah *stakeholders* yang terlibat dalam proses *supply chain*:

1. *Suppliers*

   Merupakan *stakeholder* yang berperan pada rantai pertama *supply chain* yaitu menyediakan bahan pertama seperti bahan baku, mentah, suku cadang, dan sebagainya. Rantai ini memulai proses pertama pada *supply chain management*.

2. *Manufacture*

   Merupakan *stakeholder* yang berperan pada rantai kedua *supply chain* yaitu mengolah bahan baku atau bahan mentah yang didapat oleh *supplier* menjadi produk yang sudah jadi yang nantinya akan digunakan untuk memenuhi kebutuhan *customer*. Rantai ini melanjutkan proses dari rantai yang pertama pada *supply chain*.

3. *Distributor*

   Merupakan *stakeholder* yang berperan pada rantai ketiga *supply chain* yaitu mengirim produk yang sudah jadi dari *manufacture* kepada *retailer* atau pedagang eceran. Walaupun ada beberapa produk yang disalurkan langsung kepada *customer*, namun secara umum menyalurkan produk tersebut ke *retailer* terlebih dahulu. Rantai ini melanjutkan proses dari rantai yang kedua pada *supply chain*.

4. *Retailers*

   Merupakan *stakeholder* yang berperan pada rantai keempat *supply chain* yaitu menjual produk yang disalurkan oleh distributor kepada konsumen. Rantai ini melanjutkan proses dari rantai yang ketiga pada *supply chain*.

5. *Costumer*

   Merupakan *stakeholder* yang berperan pada rantai terakhir supply chain sebagai konsumen atau pembeli produk yang disediakan oleh *retailers*. Rantai ini melanjutkan proses dari rantai yang keempat dan merupakan rantai yang terakhir dari *supply chain*.

2.5 Sertifikasi Halal

Sertifikasi halal merupakan suatu kegiatan dalam pemberian fatwa secara tertulis oleh MUI berupa sertifikat halal dari suatu produk hingga proses internal perusahaan sehingga dapat memberikan kepastian status kehalalan pada produk tersebut [6]. Pemeriksaan kehalalan produk dan sertifikasi halal dilakukan oleh Lembaga Pengkajian Pangan Obat-obatan dan Kosmetika Majelis Ulama Indonesia (LPPOM MUI) yang didirikan pada tanggal 6 Januari 1989 [33]. Dalam melakukan sertifikasi halal, LPPOM MUI menerapkan regulasi *Halal Assurance System* (HAS) atau Sistem Jaminan Halal (SJH) sehingga Sistem Jaminan Halal harus telah diterapkan oleh perusahaan sebelum melakukan pendaftaran sertifikasi halal. Hal

ini bertujuan proses produksi dapat menjaga status kehalalannya sesuai ketetapan LPPOM MUI [34].

Saat ini, regulasi HAS yang digunakan oleh LPPOM MUI sebagai persyaratan sertifikasi halal adalah HAS 23000. Regulasi ini berisi 11 kriteria Sistem Jaminan Halal (SJH) dan persyaratan lain, seperti kebijakan dan prosedur sertifikasi halal. Perusahaan wajib memenuhi 11 kriteria SJH yang terdapat pada HAS 23000 untuk mendapatkan sertifikat halal [35]. Berikut 11 krtieria SJH pada HAS 23000:

1. Kebijakan Halal
   Kebijakan halal merupakan komitmen 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
   Tim manajemen halal merupakan sekelompok anggota yang ditetapkan oleh manajemen puncak perusahaan yang disertai bukti tertulis dan bertanggung jawab terhadap perencanaan, implementasi, evaluasi, dan perbaikan sistem jaminan halal di perusahaan.

3. Pelatihan
   Perusahaan harus membuat prosedur pelatihan dan melaksanakan pelatihan tersebut minimal setahun sekali.

4. Bahan
   Bahan dapat meliputi: bahan baku, bahan tambahan, bahan penolong, kemasan, pelumas, bahan pembersih, dan media validasi. Bahan dikelompokkan menjadi dua yaitu bahan tidak kritis yaitu bahan yang termasuk ke dalam daftar halal dan kritis yaitu bahan yang diluar daftar bahan halal. Perusahaan yang memiliki bahan tidak kritis harus dilengkapi dengan dokumen pendukung yang cukup.

5. Fasilitas Produksi
   Fasilitas produksi terbagi menjadi 3 tempat yaitu:

   (a) Industri Pengolahan

   - Segala fasilitas industri pengolahan harus didaftarkan.

- Industri pengolahan harus memastikan produk tidak terkontaminasi dengan bahan kritis.
- Fasilitas yang terlibat dalam sharing facility, harus bersifat bebas dari bahan kritis.

(b) Restoran

- Segala fasilitas industri pengolahan harus didaftarkan.
- Restoran harus menjamin bahwa fasilitas dan peralatan baik ataupun fasilitas yang digunakan untuk sharing facility, terbebas dari bahan kritis.

(c) Rumah Potong Hewan (RPH)

- Fasilitas RPH hanya digunakan untuk produksi daging hewan halal. Tidak boleh tercampur dengan pemotongan daging hewan tidak halal.
- Lokasi RPH daging halal harus terpisah dengan jarak minimal 5 km dari RPH daging babi.
- Alat penyembelih baik secara manual ataupun mekanis harus memenuhi persyaratan penyembelihan halal.

6. Produk
   Produk tidak diperbolehkan memiliki rasa atau baru yang mengarah pada sesuatu yang diharamkan. Bentuk, kemasan, atau label produk tidak menggambarkan sifat erotis atau porno.
7. Prosedur Tertulis Aktivitas Kritis
   Prosedur tertulis tata cara pelaksanaan aktivitas kritis yaitu aktivitas yang dapat mempengaruhi status kehalalan produk harus dimiliki oleh perusahaan.
8. Kemampuan Telusur
   Perusahaan wajib memiliki prosedur tertulis mengenai ketelusuran produk yang disertifikasi sehingga menjamin produk berasal dari bahan dan fasilitas yang disetujui oleh LPPOM MUI.
9. Penanganan Produk Yang Tidak Memenuhi Kriteria Perusahaan wajib memiliki prosedur tertulis dalam menangani produk yang tidak memenuhi kriteria. Jika produk terlanjur dijual, maka produk harus segera ditarik.

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 wajib memiliki prosedur tertulis mengenai kaji ulang manajemen. Kaji ulang manajemen dilakukan minimal sekali dalam setahun.