BAB 2
TINJAUAN PUSTAKA

Dalam Bab ini berisi berbagai konsep dasar dan teori-teori yang berkaitan dengan pembangunan "Sistem Pengarsipan Ijazah dan Sertifikat Uji Kompetensi Berbasis Blockchain dan IPFS untuk Menjaga Integritas Data".

2.1 Arsip

Arsip berasal dari Bahasa asing, orang Yunani mengatakan *"Archivum"* yang artinya tempat untuk menyimpan, sering juga kata tersebut ditulis *"Archeon"* yang berarti balai kota (tempat untuk menyimpan dokumen) tentang masalah pemerintahan [16].

Arsip adalah setiap catatan (record atau warkat) yang tertulis, tercetak, atau ketikan, dalam bentuk huruf, angka atau gambar, yang mempunyai arti dan tujuan tertentu sebagai bahan komunikasi dan informasi, yang terekam pada kertas (kartu formulir), kertas film (slide, film-strip, mikro film), media komputer (pita tape, piringan, rekaman, disket), kertas photo copy dan lain-lain [16], [17].

2.2 Ijazah

Ijazah merupakan surat tanda tamat belajar sehingga dapat diperoleh seseorang jika telah menyelesaikan proses pembelajaran pada jenjang pendidikan tertentu. Pada kehidupan sehari-hari ijazah digunakan untuk berbagai hal, diantaranya yaitu syarat melamar pekerjaan ataupun syarat untuk melanjutkan ke pendidikan selanjutnya, dan juga menjadi bukti intelektualitas seseorang [15].

2.3 Sertifikat Uji Kompetensi

Sertifikat Uji Kompetensi adalah pengakuan kompetensi atas prestasi siswa yang sesuai dengan keahlian dalam cabang ilmunya, atau memiliki prestasi di luar program studinya yang telah lulus ujian kompetensi. Sertifikat kompetensi dapat digunakan sebagai syarat untuk memperoleh pekerjaan yang membutuhkan keahlian tertentu. Pihak yang diperbolehkan menerbitkan sertifikat adalah pihak sekolah dengan organisasi profesi, lembaga pelatihan, atau lembaga yang terakreditasi. Informasi yang tertera di dalam sertifikat kompetensi adalah nomor sertifikat, logo dan penerbit sertifikat, nama program studi, nama pemilik, tempat

tanggal lahir, tanggal, bulan, dan pustaka kelulusan uji kompetensi, sistem pengujian, dan area kompetensi kelulusan [18].

Adapun manfaat dari sertifikat kompetensi diantaranya adalah [19]:

1. Meningkatkan kepercayaan diri
2. Mengukur kemampuan
3. Akses pengembangan diri
4. Penyaringan calon karyawan
5. Meningkatkan produktivitas

2.4 Ketertelusuran

Ketertelusuran merupakan kemampuan untuk mengikuti jejak atau asal usul suatu informasi atau produk dari awal hingga akhir. Dalam konteks teknologi informasi, ketertelusuran digunakan untuk melacak aliran data atau informasi seperti transaksi atau aktivitas yang dilakukan oleh suatu entitas atau aktor [20], [21].

Tujuan dari ketertelusuran adalah untuk mengidentifikasi dan memahami sumber atau aliran data yang terjadi dari sebuah sistem yang beroperasi, sehingga memberikan keyakinan dan kepercayaan bagi entitas yang terlibat [22].

2.5 UML *(Unified Modelling Language)*

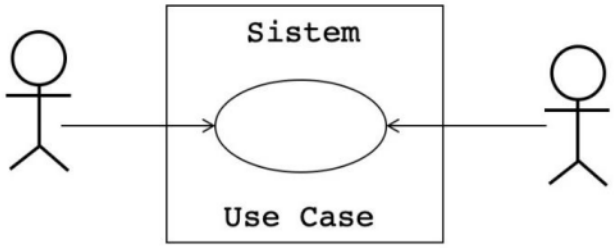
Unified Modelling Language (UML) adalah sebuah "Bahasa" yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem perangkat lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem. Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada perangkat keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan *class* dan *operation* dalam konsep dasarnya, maka lebih cocok untuk penulisan perangkat lunak dalam bahasa- bahasa berorientasi objek seperti C++, Java, C# atau VB.NET. Walaupun demikian, UML tetap dapat digunakan untuk modeling aplikasi prosedural dalam VB atau C [23]. UML juga mempunyai kelebihan yang

dapat memudahkan seorang pengembang sistem perangkat lunak dalam merancang sistem yang akan dibuat karena sifatnya yang berorientasi pada objek [24].

Terdapat beberapa diagram yang biasanya digunakan untuk memodelkan analisis fungsional dalam rangka pengembangan perangkat lunak. Berikut diantaranya diagram yang umum digunakan:

a) *Use Case Diagram*

Menggambarkan sejumlah *external actor* dan hubungannya ke *use case* yang diberikan oleh sistem. *Use case* adalah deskripsi fungsi yang disediakan oleh sistem dalam bentuk teks sebagai dokumentasi dari *use case symbol*. *Use case* digambarkan hanya yang dilihat dari luar oleh *actor* dan bukan bagaimana fungsi yang ada di dalam sistem. Ilustrasi dari *actor*, use case, dan *boundary* dapat dilihat pada Gambar 2.1.



Gambar 2.1 *Use Case Model*

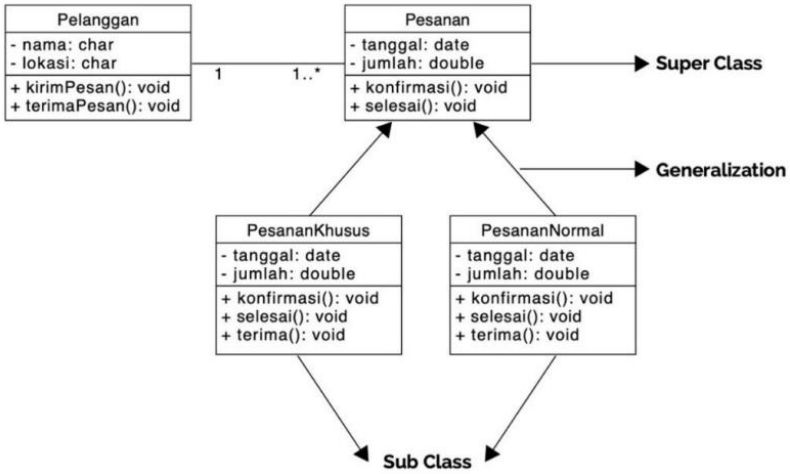
b) *Activity Diagram*

Menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktivitas yang dibentuk dalam suatu operasi. Activity diagram dibuat sebanyak aktivitas yang digambarkan pada use case diagram.

c) *Class Diagram*

Menggambarkan struktur statis class di dalam sistem. Class mempresentasikan sesuatu yang ditangani oleh sistem. Class dapat berhubungan dengan yang lain melalui berbagai cara: *associated* (terhubung satu sama lain), *dependent* (satu class tergantung/menggunakan class yang lain), *specialed* (satu class merupakan spesialisasi dari class lainnya), atau *package* (grup bersama sebagai satu unit). Sebuah sistem

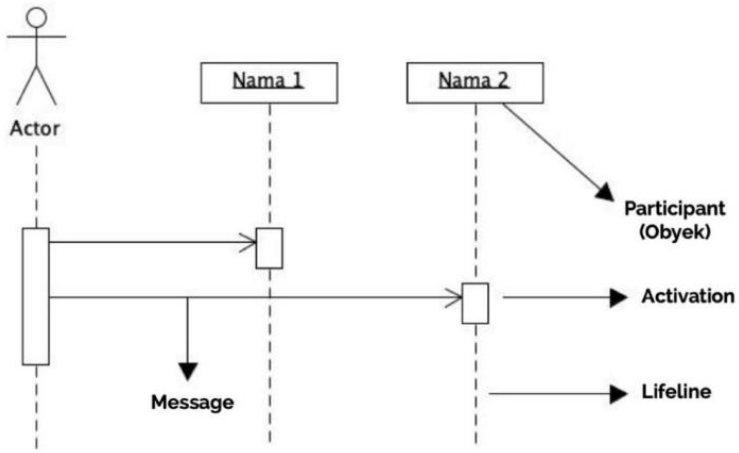
biasanya mempunyai beberapa class diagram dapat dilihat pada Gambar 2.2.



Gambar 2.2 Contoh Class Diagram pada Sistem Pemesanan

d) Sequence Diagram

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. Simbol-simbol yang ada pada *sequence diagram* dapat dilihat pada Gambar 2.3.



Gambar 2.3 Simbol Yang Terdapat Pada *Sequence Diagram*

2.6 Aplikasi Web

Aplikasi Web adalah program aplikasi yang disimpan di server jarak jauh dan dikirimkan melalui internet melalui antarmuka browser. Aplikasi web dapat dirancang untuk berbagai kegunaan dan dapat digunakan oleh siapa saja dari

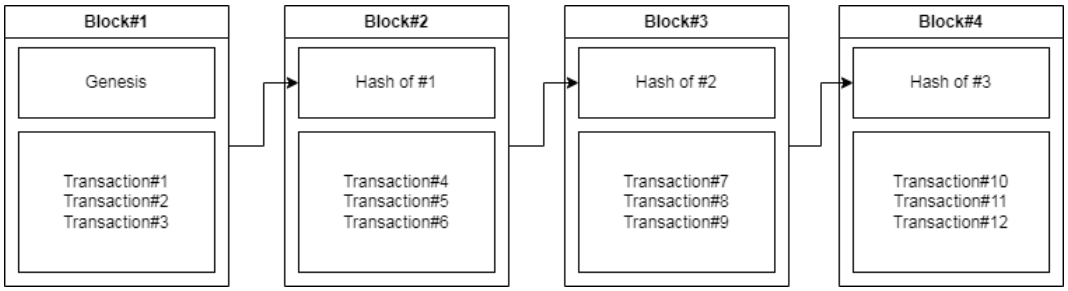
organisasi ke individu karena berbagai alasan [25]. Umumnya aplikasi web dapat digunakan mencakup webmail, toko *e-commerce* atau lainnya. Beberapa aplikasi web dapat diakses melalui browser tertentu. Namun, sebagian besar dapat diakses melalui browser apapun.

Menurut Educhannel [26], ada beberapa keuntungan dan kemudahan jika menggunakan aplikasi berbasis web, diantaranya adalah:

1. Dapat diakses dari manapun tanpa perlu menginstal aplikasi karena telah terpasang pada server.
2. *Multi-platform* atau dapat digunakan pada sistem operasi apapun, baik pada komputer dengan sistem operasi Windows, Mac OS ataupun Linux, yang terpenting pada komputer tersebut telah terhubung ke internet dan sudah terpasang web browser.
3. Terkait dengan isu lisensi (hak cipta), telah menjadi tanggung jawab dari penyedia aplikasi web sehingga pengguna tidak memerlukannya lagi.
4. Dapat diakses melalui media apapun seperti: smartphone, tablet dan komputer atau laptop.

2.7 *Blockchain*

Blockchain adalah sistem buku besar *(master-ledger)* dengan catatan setiap transaksi yang pernah ada dalam bentuk jaringan database terdesentralisasi [5]. Seperti sistem perusahaan yang lingkungannya memiliki banyak layanan, teknologi blockchain dapat diterapkan [27]. Blockchain dapat berfungsi sebagai blockchain privat *(permissioned blockchain)* atau blockchain publik *(public blockchain)* [28].



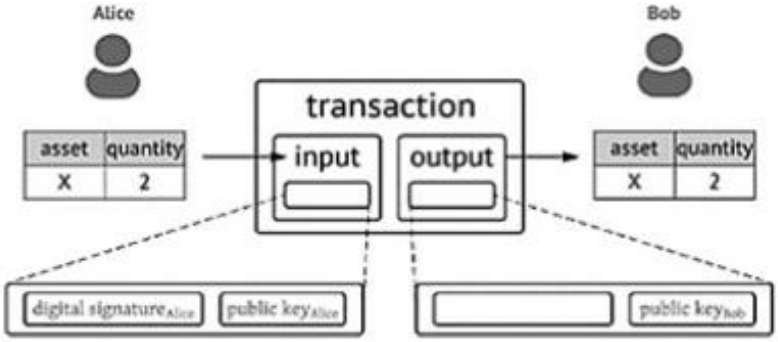
Gambar 2.4 Ilustrasi Blockchain [29]

Gambar 2.4 adalah ilustrasi dari blockchain, yang mana blockchain merupakan struktur data yang tidak dapat dihapus ataupun diubah yang dibentuk

oleh serangkaian blok data yang terhubung secara linear dalam urutan waktu. Informasi disimpan di setiap blok dan dienkripsi dengan algoritma kriptografi asimetris untuk memastikan keamanan akses dan transmisi data. Ciri-ciri teknologi blockchain yang dapat diunggulkan adalah desentralisasi *(decentralized)*: blockchain terdiri dari blok peer-to-peer (P2P), yang akan merekam dan menyimpan semua transaksi, kepercayaan terdistribusi *(distrusting)*: karena teknologi blockchain diterapkan secara desentralisasi sistem, transfer data antar node dalam jaringan tidak memerlukan rasa saling percaya di antara para peserta, transparasi *(transparency)*: melalui blockchain, semua peserta berbagi catatan dan permintaan data dalam node pada struktur terdesentralisasi, dapat dilacak dan unforgeable: blockchain menggunakan stempel waktu untuk mengidentifikasi dan merekam setiap transaksi, sehingga meningkatkan dimensi waktu data, anonimitas: blockchain mengenkripsi data menggunakan teknik enkripsi asimetris. Enkripsi asimetris ini memiliki dua kegunaan dalam blockchain: enkripsi data dan tanda tangal digital *(digital signature)*. Enkripsi data dalam blockchain memastikan keamanan data transaksi dan mengurangi risiko kehilangan atau pemalsuan data transaksi. Kredibilitas: pertukaran data blockchain sepenuhnya tergantung pada setiap node untuk membentuk perhitungan yang kuat untuk bertahan dari serangan eksternal tanpa campur tangan manusia [29].

2.8 *Smart Contract*

Perkembangan blockchain pada tahap selanjutnya disebut dengan Blockchain 2.0. Penggunaan teknologi blockchain yang semakin luas dengan memanfaatkan revolusi mekanisme baru yang disebut kontrak cerdas *(smart contract)* konsep *smart contract* dapat dilihat pada Gambar 2.5.



Gambar 2.5 Ilustrasi *Smart Contract* [29]

Smart Contract adalah kontrak yang dibangun dengan tujuan khusus untuk mengeksekusi serangkaian instruksi lengkap pada blockchain [30], [31]. *Smart Contract* adalah program komputer yang berisi perjanjian kontrak antar entitas, yang dihasilkan oleh pengguna dan diekstraksi oleh lingkungan (blockchain). Tujuan *smart contract* adalah untuk efisiensi, keamanan, dan kemandirian dalam perjanjian, mengurangi biaya implementasi kontrak dan meningkatkan kepercayaan antar entitas [32]. Kontrak adalah mekanisme yang melibatkan aset digital dan dua pihak atau lebih. Beberapa atau semua pihak memasukkan aset, yang secara otomatis didistribusikan kembali di antara pengguna sesuai dengan formula, pada data tertentu yang tidak diketahui kapan kontrak dimulai [33]. Berdasarkan mekanismenya, kontrak cerdas memiliki lima tahap perkembangan, yaitu: 1) negosiasi; 2) pengembangan; 3) penyebaran; 4) pemeliharaan; dan 5) pembelajaran dan penghancuran diri [34]. *Smart Contract* dapat mengurangi campur tangan manusia dalam alur proses bisnis di lingkungan sistem [35] dan memiliki kemampuan audit otomatis sehingga kontak dan pekerjaan yang harus dilakukan pengguna dapat diselesaikan lebih cepat dan efisien [36].

2.9 Konsensus

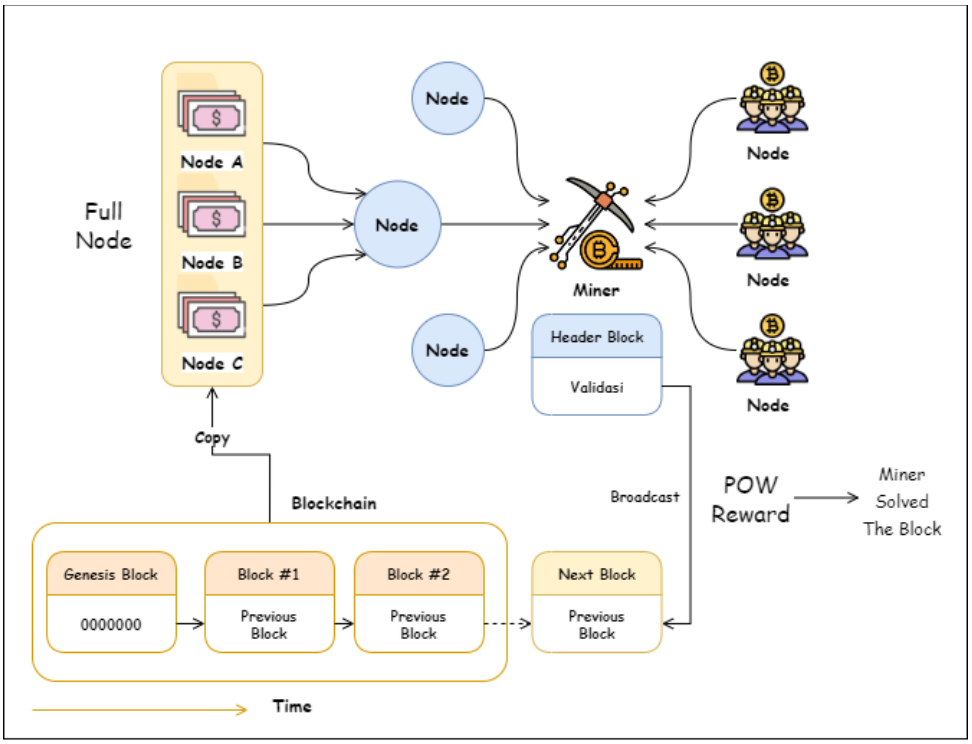
Konsensus merupakan kemampuan untuk mencapai kesepakatan bersama. Konsensus dalam blockchain adalah suatu proses perhitungan rumit untuk menghasilkan kesepakatan bersama tentang validasi suatu transaksi. Dengan kata lain konsensus dimaksudkan untuk menghasilkan sistem yang ketat tanpa aturan penguasa. Tidak ada satu orang, organisasi atau kelompok yang bertanggungjawab atau lebih tepatnya kekuatan dan kontrol tersebar di seluruh jaringan peserta.

Kemampuan untuk mencapai konsensus diseluruh jaringan terdistribusi dibawah kondisi persaingan dan tanpa kontrol yang bersifat sentral merupakan prinsip inti dari *public blockchain*. Berikut merupakan beberapa algoritma konsensus yang digunakan untuk memvalidasi transaksi pada jaringan blockchain.

2.9.1 *Proof of Work* (POW)

Proof of Work (POW) merupakan sebuah protokol yang mempunyai fungsi untuk mencegah aktifitas serangan DDoS yang dapat melumpuhkan atau melemahkan suatu sumber daya sistem komputer. Konsep POW pertama kali dikenalkan oleh Cynthia Dwork & Moni Naor pada tahun 1993 dan baru diimplementasikan oleh Markus Jakobsson (mata uang Shell) pada tahun 2009. Dalam teknologi *blockchain*, POW digunakan oleh Satoshi Nakamoto sebagai algoritma konsensus dan Bitcoin sendiri sebagai mata uang dari konsensus *Proof of Work.* Persyaratan utama dalam konsensus POW adalah proses kegiatan mining (proses komputasi dari CPU, GPU, ASIC, FPGA) yang berfungsi sebagai penemu, pencari solusi dan memvalidasi setiap masalah *(hash)* kedalam sebuah *block* dan akan didistribusikan ke dalam sebuah buku besar *(ledger)* yang disebut dengan *blockchain.*

Untuk mencapai sebuah konsensus, sebuah transaksi harus melewati beberapa proses yang juga melibatkan adanya proses komputasi yang dilakukan oleh beberapa *miners*, sehingga bisa terciptanya sebuah block yang valid. Sistem distribusi konsensus *proof of work* dapat dilihat pada Gambar 2.6.



Gambar 2.6 Ilustrasi Sistem Distribusi *Proof of Work* (POW)

Berikut merupakan penjelasan dari alur transaksi pada jaringan konsensus *proof of work*:

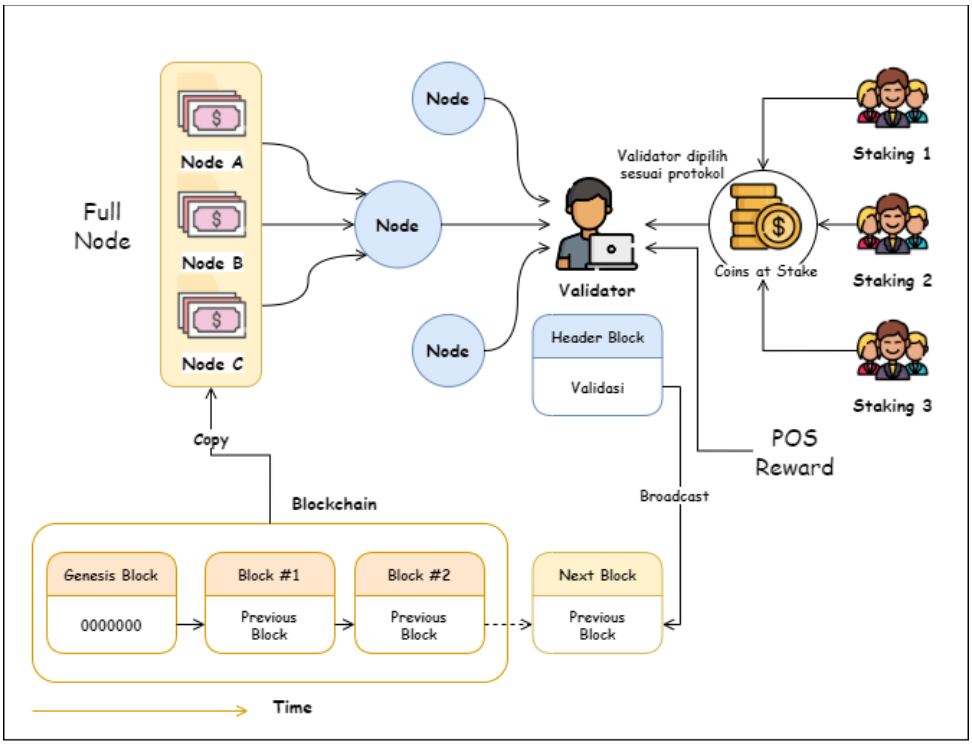
1) Suatu transaksi yang muncul dari sebuah wallet yang bertindak sebagai full node (salinan blockchain) akan di publikasi pada jaringan peer-to-peer (P2P).
2) Transaksi-transaksi ini akan saling terhubung ke sebuah jaringan blockchain yang juga terhubung dengan node miners.
3) Miner akan melakukan proses komputasi *(hash function)* untuk menyelesaikan persoalan matematika rumit ini ke dalam sebuah block.
4) Jumlah maksimal transaksi dalam setiap *block* tergantung dari protokol yang berlaku.
5) Setelah masalah *(hash)* terpecahkan, maka miner yang pertama kali memecahkan masalah ini akan melakukan *broadcast block* ke jaringan (P2P).
6) Node *(miner)* lainnya yang menerima *block* ini akan melakukan proses verifikasi.

7) Setelah *block* mendapat validasi, maka *block* ini akan didistribusikan ke dalam *blockchain* sebagai *block* baru yang valid.
8) *Miners* bertindak sebagai pembuat block valid akan menerima bayaran *(reward)*.

2.9.2 *Proof of Stake* (POS)

Proof of Stake (POS) adalah algoritma konsensus alternatif yang lebih hemat energi dibandingkan *Proof of Work* (POW). Pada POS, penambang harus membuktikan kepemilikan sejumlah mata uang agar bisa berpartisipasi dalam proses konsensus. Hal ini dipercayai akan membuat orang dengan kepemilikan mata uang yang lebih banyak memiliki kecil kemungkinan untuk menyerang jaringan. Namun, pemilihan berdasarkan saldo akun tidak adil karena orang terkaya akan mendominasi jaringan blockchain. Untuk mengatasinya, ada banyak solusi yang mengkombinasikan ukuran stake dengan faktor-faktor lain untuk memutuskan blok mana yang akan ditambang selanjutnya. Sebagai contoh, Blockchain menggunakan pengacakan untuk memprediksi generator berikutnya dengan menggunakan formula yang mencari nilai hash terendah dalam kombinasi dengan ukuran stake. Sementara itu, Peercoin menggunakan pemilihan berdasarkan usia koin, di mana set koin yang lebih tua dan lebih besar memiliki kemungkinan lebih besar untuk menambang blok berikutnya.

Dibandingkan dengan POW, POS lebih hemat energi dan lebih efisien. Namun, ada risiko serangan karena biaya penambangan hampir nol. Oleh karena itu, banyak blockchain mengadopsi POW di awal dan kemudian beralih secara bertahap ke POS. Sebagai contoh, Ethereum sedang merencanakan untuk beralih dari Ethash (sejenis POW) ke Casper (semacam POS). Sistem distribusi konsensus *proof of stake* dapat dilihat pada Gambar 2.7.



Gambar 2.7 Ilustrasi Sistem Distribusi *Proof of Stake* (POS)

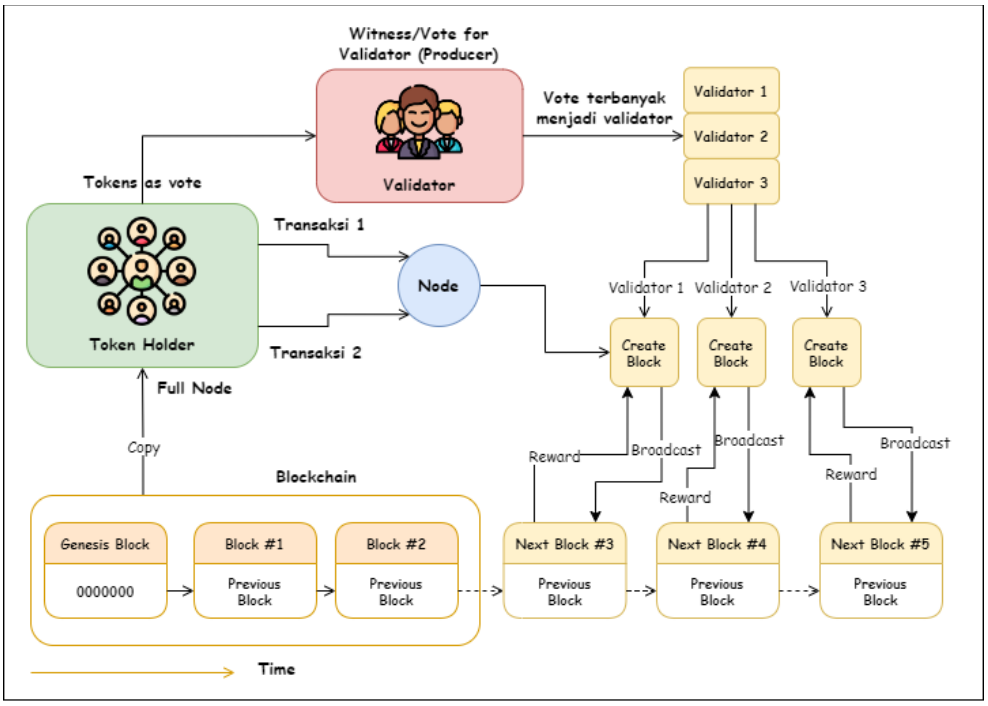
Berikut merupakan penjelasan dari alur transaksi pada jaringan konsensus *proof of stake*:

1) Untuk bisa melakukan forging *(minting)*, maka pemegang token harus mengunci sejumlah koin didalam walletnya (full node) yang terhubung dengan jaringan blockchain.
2) Pada saat sebuah transaksi masuk ke dalam jaringan (P2P). Pembuatan sebuah *block* (validasi transaksi) dipilih secara *pseudorandom* berdasarkan jumlah koin yang di *stake* dan berapa lama koin tersebut sudah di *stake*.
3) *Token Holder* dengan jumlah staking coin yang besar dan waktu (umur) staking yang lama, mempunyai peluang (kesempatan) lebih tinggi untuk melakukan proses forging pada block berikutnya.
4) Setelah proses pembuatan block dan proses validasi selesai, maka block ini akan didistribusikan ke dalam jaringan blockchain sebagai block baru yang valid.
5) Forger (staker) akan menerima bayaran dari hasil kerjanya dan untuk umur *staking coinnya* akan direset ulang.

2.9.3 *Delegated Proof of Stake* (DPOS)

Delegated Proof of Stake (DPOS) diciptakan oleh Daniel Larimer pada saat membuat sebuah *cryptocurrency* yang disebut dengan BitShares pada tahun 2014. Algroritma konsensus ini adalah sebua metode baru sebagai tindakan pengamanan terhadap sebuah jaringan cryptocurrency untuk mencapai sebuah konsensus tanpa memerlukan adanya otoritas terpusat. Prinsip kerjanya hampir mirip dengan konsensus Proof of Stake (POS), dimana untuk mencapai konsensus yang valid maka dibutuhkan proses verifikasi dari pelaku-pelaku otoritas terdistribusi dan berjalan sesuai dengan protokol yang ada didalam metode konsensus tersebut.

Jika didalam POS pemilihan pembuatan block bersifat *Pseudorandom* (berdasar jumlah dan umur coin/token). Maka pada DPOS setiap witness (pembuat block) akan ditentukan dengan voting. Setiap pemegang koin/token mempunyai suara (vote) yang bisa digunakan untuk memilih satu atau beberapa witness atau validator dan dapat memindahkan vote-nya kepada witness lainnya setiap saat. Validator atau witness terpilih (diurutkan berdasarkan jumlah vote) akan mempunyai otoritas untuk membuat sebuah block dan mempublikasinya ke dalam jaringan blockchain. Sistem distribusi delegated proof of stake (DPOS) dapat dilihat pada Gambar 2.8.



Gambar 2.8 Ilustrasi Sistem Distribusi *Delegated Proof of Stake* (DPOS)

Berikut merupakan penjelasan dari alur transaksi pada jaringan konsensus *delegated proof of stake*:

1) Setiap transaksi yang terjadi akan masuk ke jaringan (P2P).
2) Token/Coin Holder yang menjalankan full node (wallet) akan melakukan vote untuk menentukan beberapa witness/delegasi. Semakin besar jumlah token maka akan semakin besar juga nilai votenya.
3) Token/Coin Holder bisa melakukan vote terhadap dirinya sendiri setelah melakukan registrasi menjadi witness (validator) sesuai dengan protokol yang berlaku.
4) Dalam setiap putaran, masing-masing validator/witness terpilih (jumlah maksimal (N) witness berdasarkan protokol yang berlaku) akan memproses beberapa transaksi ke dalam sebuah block.
5) Sebelum block valid dibroadcast ke dalam blockchain, urutan ranking dari validator/witness akan berubah-ubah sesuai dengan jumlah vote yang didapat.
6) Setelah block berhasil divalidasi oleh witness/validator, maka masin-masing block akan didistribusikan ke dalam jaringan blockchain berdasarkan urutan rank dari validator.

7) Masing-masing validator/witness akan menerima reward sesuai protokol yang berlaku.

2.9.4 *Practical Byzantine Fault Tolerance* (PBFT)

Dalam sistem terdistribusi, toleransi kesalahan Bizantium dapat digunakan untuk memperbaiki kesalahan transmisi. Namun, sistem Bizantium awal memerlukan operasi yang kompleks. Pada tahun 1999, PBFT *(Practical Byzantine Fault Tolerance)* dikembangkan dengan mengurangi kompleksitas algoritmanya menjadi tingkat polinomial, sehingga meningkatkan efisiensi.

Proses PBFT melibatkan lima tahap, yaitu:

1) Permintaan: Klien mengirimkan permintaan ke node master server, dan node master memberikan cap waktu.
2) Pre-Prepare: Node server master merekam permintaan pesan dan memberikan nomor urutan. Kemudian node master menyiarkan pesan pra-persiapan ke node server berikutnya. Node server lain menentukan apakah akan menerima permintaan atau tidak.
3) Persiapan: Jika node server memilih untuk menerima permintaan, ia akan menyiarkan pesan persiapan ke semua node server lain dan menerima pesan persiapan dari node lain. Jika mayoritas node memilih untuk menerima permintaan setelah mengumpulkan pesan persiapan, maka itu memasuki kondisi komit.
4) Komit: Setiap node dalam keadaan komit mengirimkan pesan komit ke semua node lain di server. Jika node server menerima pesan komit, diyakini bahwa sebagian besar node telah mencapai konsensus untuk menerima permintaan tersebut, dan node akan mengeksekusi instruksi dalam permintaan.
5) Balasan: Node server membalas klien, dan jika klien tidak menerima balasan karena penundaan jaringan, permintaan dikirim ulang ke node server. Jika permintaan telah dieksekusi, node server hanya perlu mengirim pesan balasan sekali saja.

2.10 JavaScript

JavaScript adalah sebuah Bahasa pemrograman yang digunakan untuk membuat kumpulan skrip yang dapat berjalan pada sebuah dokumen HTML. Bahasa ini telah menjadi Bahasa skrip pertama dalam sejarah internet yang digunakan untuk memperluas kemampuan Bahasa HTML dengan memungkinkan penggunaan perintah-perintah pada sisi pengguna, yaitu di sisi browser *(client-side)*, bukan di sisi server web *(server-side)*.

JavaScript diperkenalkan oleh Netscape pada tahun 1995 dengan nama awal "LiveScript". Awalnya, bahasa ini digunakan sebagai bahasa sederhana untuk browser *Netscape* Navigator 2. Meskipun demikian, JavaScript banyak dikritik pada saat itu karena kurang aman dan pengembangannya terkesan terburu-buru serta tidak menampilkan pesan kesalahan saat membuat kesalahan dalam menulis program. Namun, seiring dengan kerjasama yang semakin erat antara *Netscape* dan *Sun Microsystems* (pengembang bahasa pemrograman "Java"), *Netscape* kemudian memberikan nama "JavaScript" kepada bahasa ini pada tanggal 4 Desember 1995. Pada saat yang sama, Microsoft juga mencoba mengadopsi teknologi ini dengan nama "JScript" pada browser Internet Explorer 3.

Javascript bergantung pada browser *(navigator)* yang memanggil halaman web yang berisi skrip-skrip dari javascript dan tentu saja disisipkan di dalam dokumen HTML. Javascript juga tidak memerlukan kompilator atau penerjemah khusus untuk menjalankannya (pada kenyataannya kompilator javascript sendiri sudah termasuk dalam browser tersebut) [37].

2.11 Node JS

Node.js merupakan sebuah platform perangkat lunak open source yang digunakan untuk membangun aplikasi berbasis *server-side* menggunakan Bahasa pemrograman JavaScript. Node.js berjalan pada mesin JavaScript bernama V8 yang juga digunakan oleh browser Google Chrome. Node.js menyediakan *runtime environtment* untuk JavaScript yang *asynchronous* sehingga memungkinkan aplikasi untuk mengoperasikan I/O *(input/output)* dan jaringan dengan efisien dan cepat.

Dengan menggunakan Node.js, pengembang dapat dengan mudah membuat aplikasi web karena *backend* dan *frontend* aplikasi menggunakan Bahasa pemrograman yang sama yaitu JavaScript. Node.js dilengkapi dengan fitur-fitur seperti modul yang terpisah, manajemen paket dengan NPM *(Node Package Manager)*, serta dukungan protokol HTTP dan *Web Socket* [38].

2.12 Docker

Docker adalah suatu platform terbuka bagi pengembang perangkat lunak dan pengelola sistem jaringan untuk membangun, menjalankan dan mengirimkan aplikasi-aplikasi yang terdistribusi. Definisi tersebut membawa pengertian praktis bahwa Docker merupakan suatu cara memasukkan layanan ke dalam lingkungan terisolasi bernama *container*, sehingga layanan tersebut dapat dipaketkan menjadi satu bersama dengan semua pustaka dan *software* lain yang dibutuhkan [39].

Terdapat dua masalah umum yang bisa diselesaikan menggunakan Docker, yaitu:

1. Mengurangi beban *hardware harddisk* dibandingkan menggunakan salinan Sistem Operasi atau mesin virtual yang membebani *hardware* fisik aslinya.
2. Lebih mudah dalam implementasi sistem yang mungkin berbeda platform atau berbeda versi tertentu.

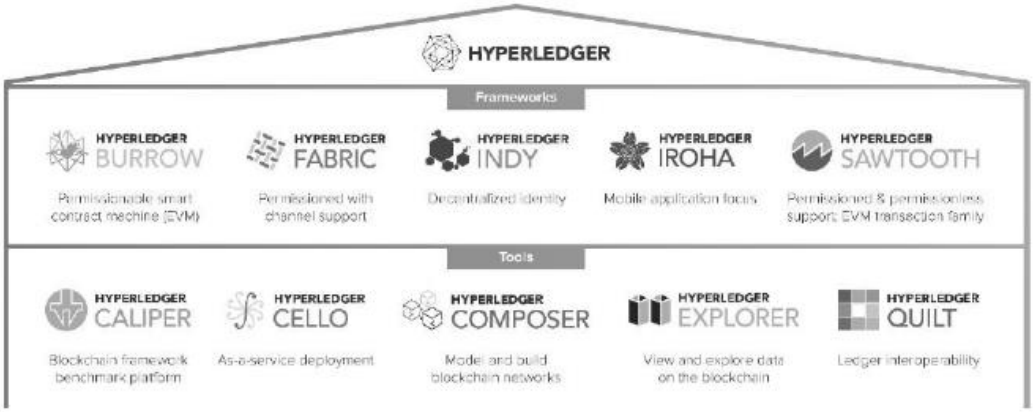
Secara teknis, Docker memanfaatkan fitur-fitur dari kernel Linux dimana daemonnya berjalan demi terwujudnya *container* yang ringan dan stabil.

2.13 Hyperledger

Hyperledger merupakan sebuah proyek kolaborasi yang bersifat *open-source* tentang blockchain. Proyek ini dikembangkan dalam rangka memajukan teknologi *blockchain* dalam implementasi di bidang industri. *Hyperledger* berisi kolaborasi atas bidang perbankan, manufaktur, keuangan, rantai pasok, *Internet of Things*, dan teknologi. Proyek ini mulai diperkenalkan sejak tahun 2015 dimana banyak perusahaan tertarik dengan eksistensinya dan kesadaran akan manfaat dari bekerja dengan kolaborasi. Perusahaan-perusahaan ini memutuskan mengumpulkan sumber daya untuk menciptakan teknologi blockchain *open-source* yang dapat digunakan oleh siapapun. *Hyperledger* berada dibawah *Linux*

Foundation yang telah berkembang pesat dalam beberapa tahun terakhir. Pada tanggal publikasi, *Hyperledger* memiliki lebih dari 230 organisasi anggota serta 10 proyek dengan 3,6 juta baris kode, 10 kelompok kerja aktif, dan hampir 28.000 peserta yang datang ke 110 pertemuan di seluruh dunia [40].

Hyperledger berfungsi sebagai *"greenhouse"* yang menyatukan pengguna, pengembang, dan vendor dari berbagai sektor dan ruang pasar. Semua partisipan memiliki satu kesamaan, yaitu tertarik dalam mempelajari, mengembangkan dan menggunakan blockchain. Setiap perusahaan membuka fitur dan modifikasi khusus untuk membuat blockchain mencapai tujuan. Karena setiap organisasi yang berbeda memiliki kebutuhan yang berbeda, maka tidak ada standar tunggal blockchain, tetapi dengan berbagai fitur yang disediakan dapat menjadi solusi di banyak industry untuk dapat melihat berbagai jenis blockchain. Struktur *Hyperledger Greenhouse* dapat dilihat pada Gambar 2.9 Struktur *Hyperledger Greenhouse* [40].



Gambar 2.9 Struktur *Greenhouse Hyperledger*

2.14 Hyperledger Fabric

Hyperledger Fabric adalah teknologi blockchain yang diatur oleh Linux Foundation, dibuat untuk mempromosikan pengguna teknologi blockchain ke perusahaan, terutama untuk aplikasi perusahaan. *Hyperledger Fabric* adalah platform *Distributed Ledger Technology* (DLT) *open-source* yang diizinkan perusahaan, dirancang untuk digunakan dalam konteks perusahaan, yang memberikan kemampuan membedakan kunci dibandingkan platform ledger atau

blockchain popular lainnya. *Fabric* adalah platform ledger terdistribusi pertama yang mendukung smart contract yang ditulis dalam Bahasa pemrograman Java, go dan Node.js. Platform fabric para peserta dalam jaringannya dapat diketahui satu samalain karena bersifat privat *(permissioned)*. Salah satu yang terpenting pembeda platform adalah dukungannya untuk protokol consensus pluggable yang memungkinkan platform untuk lebih efektif disesuaikan dengan kasus penggunaan tertentu dan model kepercayaan. Fabric dapat memanfaatkan protokol consensus yang tidak memerlukan *cryptocurrency asli* atau monetisasi data untuk intensif penambangan yang mahal atau untuk memicu eksekusi *smart* contract [9]. Berikut adalah beberapa tujuan dari Hyperledger [10].

1. *Rich Query* yang berarti pengguna dapat melakukan transaksi yang dilakukan pada platform blockchain.
2. Arsitektur modular, artinya berbagai modul dapat digunakan baik secara bersamaan maupun sesuai kebutuhan.
3. Perlindungan kunci digital dan data sensitif berarti mengerjakan konsep distribusi buku besar dan melindungi kunci digital dan data dari gangguan.
4. Izin data yang artinya hanya pihak tertentu saja yang diperbolehkan untuk dapat melihat dan menggunakan data tertentu.

Adapun komponen-komponen yang terdapat dalam *Hyperledger Fabric* yang perlu untuk diketahui juga, diantaranya adalah [40]:

1. *Asset:* asset merupakan segala sesuatu yang memiliki nilai. Aset memiliki kepemilikan. Aset diwakili dalam *Hyperledger Fabric* sebagai kumpulan pasangan nilai-kunci.
2. *Shared Ledger:* buku besar yang mencatat keadaan dan kepemilikan suatu asset, terdiri atas dua komponen yaitu *the world state* yang menggambarkan keadaan buku besar pada suatu titik tertentu, dan blockchain yaitu riwayat log transaksi yang mencatat seluruh transaksi.
3. *Smart Contract: chaincode* yaitu perangkat lunak yang mendefinisikan asset dan transaksi terkait. Dengan kata lain, *smart contract* berisi logika bisnis sistem. *Chaincode* dipanggil saat aplikasi perlu berinteraksi dengan

buku besar. *Chaincode* dapat ditulis dalam Bahasa pemrograman Golang ataupun Node.Js.

4. *Peer nodes:* elemen dasar dari jaringan karena *peer* merupakan tuan rumah buku besar dan *chaincode*. Sebuah *peer* mengeksekusi *chaincode*, mengakses data buku besar, mengatur transaksi, dan berinteraksi dengan aplikasi. Beberapa *peer* lainnya dapat menjadi *endorser* yang mendukung. Setiap *chaincode* dapat menentukan kebijakan *endorsement*, yang mendefinisikan kondisi yang diperlukan dan cukup untuk *endrose* transaksi yang valid.
5. *Channel:* struktur logis yang dibentuk oleh sekumpulan *peer*. Kemampuan ini membuat sekelompok *peer* memungkinkan untuk membuat transaksi *ledger* yang terpisah.
6. *Organizations: Hyperledger Fabric* dibangun dari *peers* yang dimiliki dan diberikan oleh berbagai organisasi yang menjadi anggota jaringan. Jaringan ada karena organisasi memberikan sumber daya individual mereka ke jaringan kolektif. *Peer* memiliki identifitas sertifikat digital yang ditetapkan oleh *membership service provider* (MSP) dari organisasi pemiliknya. *Peer* dari organisasi yang berbeda dapat berada di saluran yang sama.
7. *Membership Service Provider* (MSP): *Membership Service Provider* (MSP) diimplementasikan sebagai *Certificate Authority* (CA) untuk mengelola sertifikat yang digunakan untuk mengotentikasi identitas dan peran anggota. Tidak ada identitas yang tidak dikenal yang dapat berinteraksi di jaringan *Hyperledger Fabric*. MSP mengelola ID pengguna dan mengotentikasi semua peserta di jaringan yang memungkinkan *Hyperledger Fabric* sebagai jaringan pribadi.
8. *Ordering Service:* transaksi paker *ordering service* dijadikan blok untuk dikirim ke seluruh *peer* di saluran. *Ordering service* menjamin pengiriman transaksi dalam jaringan. *Ordering service* berkomunikasi dengan *peers* dan me-endorse peer lainnya. Mekanisme konfigurasi yang didukung untuk *ordering service* adalah Solo dan Kafka.

Dalam blockchain *Hyperledger Fabric* berfungsi sebagai *back-end* dengan aplikasi *front-end* untuk berkomunikasi dengan jaringan. SDK membantu mengatur

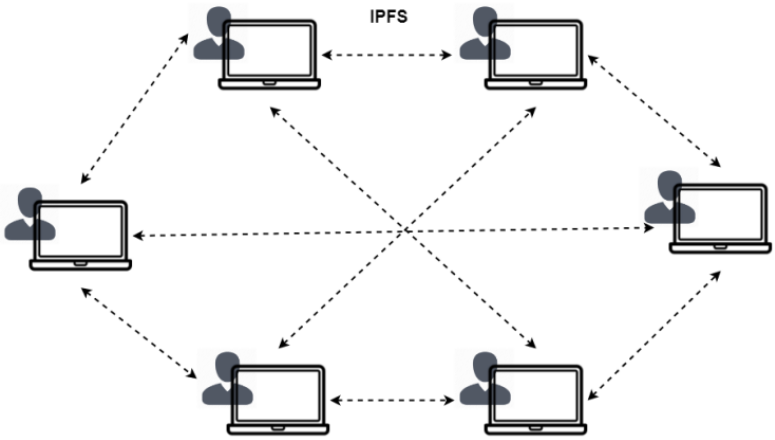
komunikasi antara *front-end* dan *back-end*, seperti Node.Js SDK dan Java SDK. SDK menyediakan cara untuk mengeksekusi kode berantai pengguna, melakukan transaksi di jaringan, memantau event dan lain-lain [40].

2.15 Fablo

Fablo merupakan sebuah alat yang digunakan untuk menghasilkan jaringan blockchain *Hyperledger Fabric* dan menjalankan menggunakan Docker. Alat ini mendukung protokol konsensus RAFT dan Solo, organisasi-organisasi dan saluran yang berbeda, serta instalasi dan peningkatan *chaincode (smart contract)* [41]. Dengan Fablo, pengembang dapat dengan mudah membuat dan mengkonfigurasi jaringan *Hyperledger Fabric* yang terdiri dari beberapa organisasi mengatur saluran-saluran (*channels*) untuk membatasi akses data, dan mengelola proses instalasi dan pembaruan *chaincode*. Fablo menyediakan CLI yang sederhana sehingga pengguna dapat memulai dengan cepat dalam pengembangan aplikasi *permissioned* blockchain berbasis *Hyperledger Fabric*.

2.16 *Interplanetary File System* (IPFS)

IPFS *(InterPlanetary File System)* adalah sebuah teknologi yang dapat mendesentralisasikan data agar konten yang terdapat pada internet bisa didistribusikan secara *peer-to-peer*, dengan teknologi IPFS file yang tersimpan akan lebih aman karena file akan dipecah menjadi potongan-potongan kecil lalu file tersebut akan di hash secara kriptografi kemudian akan diberi sidik jari unik yang disebut *content identifier* (CID) [11], [12]. Teknologi lain seperti Git menggunakan struktur kompleks yaitu *Merkle-linked*. IPFS mengintegrasikan penggunaan struktur terkait Merkle yang kompleks dengan kemampuan pengalamatan data dari sistem berbagi file P2P. Konten didistribusikan melalui jaringan peer-to-peer.



Gambar 2.10 Konsep *Peer-To-Peer* IPFS

2.16.1 *Content Based Address*

Dalam sistem sentralisasi, untuk mengakses file, umumnya digunakan *location-based addressing*, di mana interaksi antara pengguna dan host komputer tempat file tersimpan dapat dilakukan dengan mudah. Namun, metode ini tidak dapat digunakan dalam skenario *offline* atau distribusi file secara terdistribusi. Oleh karena itu, IPFS menggunakan metode *content-based addressing*. Dalam metode ini, pengguna mencari file yang diinginkan dengan menggunakan *hash* dari objek file tersebut. Setelah itu, node yang tergabung dalam jaringan yang memiliki file dengan *hash* yang dicari akan memberikannya kepada pengguna yang meminta.

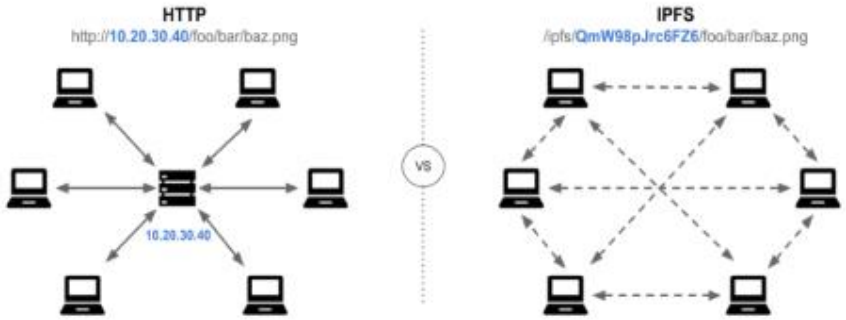
2.16.2 *Tamper Proof*

Data akan diverifikasi menggunakan checksum-nya pada IPFS, sehingga jika terjadi perubahan hash, IPFS dapat mengetahui bahwa data tersebut telah rusak.

2.16.3 *No Duplication*

File yang memiliki konten yang sama tidak akan diizinkan untuk diduplikasi dan hanya akan disimpan satu kali pada IPFS.

Perbedaan mengenai teknologi IPFS dengan HTTP dapat dilihat pada gambar 2.10.



Gambar 2.11 Perbandingan HTTP dengan IPFS

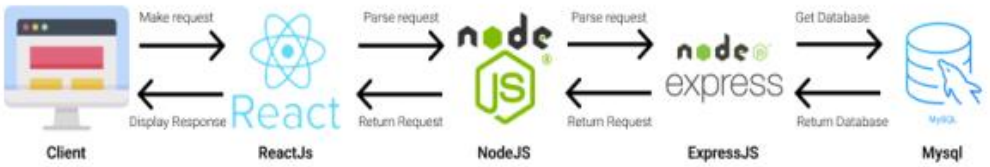
Protokol HTTP klasik yang digunakan pada web seperti yang terlihat pada gambar 2.10 menggunakan alamat lokasi, mengandalkan arsitektur terpusat dimana pengguna terhubung ke server pusat (lokasi) yang menyediakan file. Sebaliknya, IPFS menggunakan alamat konten, dimana pengguna dapat mengambil file yang diidentifikasi secara unik dari setiap node dalam jaringan terdistribusi yang menyimpan file tersebut [42].

Menggunakan IPFS tidak hanya memungkinkan untuk mengakses file pada halaman web, namun juga berbagai file yang disimpan pada komputer dalam bentuk dokumen atau bahkan rekaman database. Sehingga dalam konsepnya terdapat tiga prinsip fundamental yang digunakan oleh IPFS:

1. Identifikasi yang unik melalui konten *addressing*
2. Menemukan konten melalui *distributed hash tables* (DHTs)
3. Menghubungkan konten melalui *directed acyclic graphs* (DAGs)

2.17 MERN Stack

MERN (MySQL, Express, React dan Node.Js) Stack merupakan sekumpulan teknologi atau gaya koding untuk membangun sebuah aplikasi web. MERN Stack memiliki 2 versi, yaitu versi yang menggunakan MySQL atau versi yang menggunakan Mongo DB sebagai database utama yang digunakan dalam membangun sebuah aplikasi web [38].



Gambar 2.12 Struktur MERN Stack

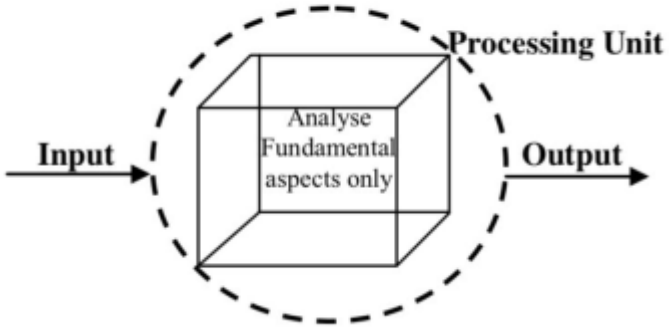
MERN stack sangat popular karena Bahasa yang digunakan untuk membangun aplikasi menggunakan satu Bahasa yaitu JavaScript. Selain itu, MERN Stack juga menerapkan konsep SPA (*Single Page Application*). Menurut Mesbah Ali, antarmuka web yang menerapkan konsep SPA terdiri dari komponen individual yang dapat diperbarui atau diganti secara independen, sehingga seluruh halaman tidak perlu dimuat pada setiap tindakan pengguna [43].

2.18 Metode Pengujian

Metode pengujian merupakan teknik atau pendekatan sistematis yang digunakan untuk mengevaluasi dan memverifikasi kualitas sistem yang dibangun. Tujuannya adalah untuk mengidentifikasi kesalahan atau cacat dalam sistem serta memastikan bahwa sistem berfungsi sesuai dengan ketentuan yang telah ditetapkan.

2.18.1 Pengujian *Black Box*

Pengujian *Black Box* yaitu pengujian perangkat lunak dari segi spesifikasi fungsionalitas tanpa menguji desain dan kode program. Metode ini dimaksudkan untuk memastikan semua fungsionalitas berjalan dengan baik dan sesuai dengan apa yang telah direncanakan [44], [45].



Gambar 2.13 Ilustrasi Metode Black Box

Metode Pengujian *Black Box* memiliki keuntungan dan kekurangan diantaranya adalah:

A. Keuntungan

1. Efisien diterapkan pada segmentasi kode yang besar.
2. Persepsi yang harus dimiliki tester sederhana.
3. Perspektif pengguna dipisahkan dari perspektif pengembang *(programmer* dan *tester* independen satu samalain).
4. Pengembangan kasus uji relatif cepat.

B. Kekurangan

1. Hanya sejumlah skenario yang dilakukan dan dipilih dalam pengujian. Akibatnya, cakupan pada pengujian terbatas.
2. Tanpa spesifikasi yang jelas sehingga kasus uji sulit untuk dirancang.
3. Pengujian tidak efisien.

2.19 Profil Sekolah

SMK Ma’arif Terpadu Cicalengka adalah sekolah menengah kejuruan yang dibawahi Yayasan Ma’arif Terpadu Cicalengka. Sekolah Menengah Kejuruan Ma’arif Terpadu Cicalengka adalah salah satu Sekolah yang berbasis holistik yang mempunyai 3 Program Studi unggulan terakreditasi. Sekolah yang mengedepankan Pendidikan dan Akhlak Mulia, menjadikan peserta didik SMK Ma’arif Terpadu Cicalengka Bermatabat, Berkualitas, dan Terpercaya.

Tabel 2.1 Biodata Sekolah

Nama Sekolah	: SMK Ma'arif Terpadu Cicalengka
Alamat Sekolah	: Jl. Rd. Dewi Sartika No.119, Cicalengka Kulon, Kec. Cicalengka, Kabupaten Bandung, Jawa Barat 40395
Alamat Email	: smkmaarifterpaduclk@gmail.com

SMK Ma'arif Terpadu Cicalengka merupakan sekolah yang mempunyai letak strategis pada pusat kota Cicalengka, yang menjadikan mobilitas SMK Ma'arif Terpadu Cicalengka menjadi mudah untuk diakses dan juga dilengkapi dengan pendidik yang berkompeten dan ahli pada setiap program studinya.

Program studi pada SMK Ma'arif Terpadu Cicalengka antara lain:

1) Perhotelan (PH)
2) Desain Komunikasi dan Visual (DKV)
3) Teknik Komputer dan Jaringan (TKJ)

2.19.1 Logo Sekolah

Logo merupakan ciri khas atau karakter yang mencerminkan suatu instansi, perusahaan, atau sekolah. SMK Ma'arif Terpadu Cicalengka memiliki logo yang dapat dilihat pada Gambar 2.14.

Gambar 2.14 Logo SMK Ma'arif Terpadu Cicalengka

2.19.2 Visi dan Misi

Sekolah SMK Ma'arif Terpadu Cicalengka memiliki visi dan misi untuk mewujudkan sekolah yang Bermatabat, Berkualitas dan Terpercaya.

1) Visi

Unggul dalam prestasi, teladan dan bertindak, takdim kepada islam dan khidmat kepada budaya.

2) Misi

a. Mengedepankan kompetensi dengan pangsa pasar dan industri
b. Menawarkan alternatif praktik wirausaha dan anak didik sebagai antisipasi daya saing dunia kerja
c. Menumbuhkan sikap profesional dalam segala tindakan akademik maupun non-akademik
d. Ber-Islam yang mantap dengan mengedepankan nilai-nilai toleransi sebagai wujud manusia berbudaya.
e. Membiasakan sikap yang dilandasi kesolehan sosial