BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

2.1.1 Blockchain Based Smart Door Lock System [11]

Penelitian "Blockchain Based Smart Door Lock System" yang diteliti oleh Donhee Han, Hongjin Kim dan Juwook Jang ini merupakan salah satu implementasi awal dari penggunaan *blockchain* untuk sistem akses kontrol fisik. Tujuan dari penelitian ini adalah untuk membuat suatu sistem pengunci pintu pintar yang mengaransikan adanya integritas data dan autentikasi dan serta dapat mengambil keputusan terhadap sebuah *event* yang didapatkan dari sensor yang berada didalam sistem. Sistem ini menggunakan sistem *blockchain* yang dibuat oleh para peneliti. Sistem fisik dari penelitian ini terdiri atas CPU, sebuah pintu dengan operator *OPEN/LOCK*, modul komunikasi menggunakan Bluetooth dan Zigbee, GPS untuk mengukur jarak, sensor *ultrasonic*, sensor PIR dan sensor gerak. Sistem bekerja dengan cara sebagai berikut: ketika sistem menerima sebuah transaksi yang berupa pesan *OPEN/LOCK*, sistem akan memulai verifikasi terhadap transaksi tersebut dengan cara mengecek jarak GPS antara pengguna yang disebut dengan *node* dengan pintu memiliki jarak dibawah *threshold*, jika iya, maka pintu akan menjalankan pesan kontrol tersebut dan mengirimkan transaksi ke jaringan *blockchain*. Jika pintu berada dalam status *LOCK* dan sensor Gerakan mendapatkan data yang menandakan bahwa ada pengguna yang tidak terautentikasi masuk, maka sistem akan mengambil keputusan apakah terdapat penyelundup yang ingin untuk memasuki ruangan. Sistem juga memiliki kemampuan untuk mengetahui apakah ada penyelundup yang ingin masuk ruangan, dengan menggunakan sensor PIR dan *ultrasonic*, dimana jika jarak antara sensor *ultrasonic* dibawah *threshold* dan sensor PIR juga mendapatkan data setelah sekian waktu, maka sistem akan mengambil keputusan bahwa penyelundup ingin memasuki

ruangan, sistem kemudian akan mengirimkan transaksi tersebut ke jaringan *blockchain*. Pada penelitian ini sistem berhasil untuk diimplementasikan.

Beberapa poin penting yang dapat didapatkan dari penelitian ini adalah:

- Penelitian ini merupakan salah satu dari penelitian pertama yang berhasil untuk mengimplementasikan sistem akses kontrol fisik dengan *blockchain*.
- Penggunaan banyak sensor dapat digunakan untuk menentukan apakah penyelundup ingin melakukan akses, namun penggunaan banyak sensor ini membuat sistem menjadi sedikit lebih rumit
- Penggunaan sistem pesan kontrol *OPEN/LOCK* dapat menjadi pilihan agar sistem lebih aman dari penggunaan sistem seperti *smart card*, karena adanya otentikasi dari pengguna

2.1.2 Physical Access Control Management System Based on Permissioned Blockchain [12]

Studi ini meneliti tentang penggunaan *blockchain* sebagai sistem akses kontrol fisik, serta penggunaan *blockchain* sebagai sistem yang *tamper-proof* sehingga dapat diaudit dengan baik. Penelitian ini dilakukan oleh Sara Rouhani, Vahid Pourheidari, Ralph deters. Pada penelitian ini, sistem dibuat dengan menggunakan *blockchain* berupa Hypledger Fabric. Untuk memberikan kontrol akses kepada pengguna, sistem dalam penelitian ini menggunakan *Role Based Access Control* dan *Rule Based Access Control. Role* pada sistem diatur dengan menggunakan ACL pada Hypledger Fabric sedangkan *Rule* untuk user diatur dengan menggunakan *Smart Contract*. Sistem kemudian dibagi menjadi beberapa modul dengan beberapa fungsi yang berbeda-beda. Modul Participant berisi *database* pengguna, modul Asset berisi mesin fisik yang digunakan pada sistem, modul Transaction berisi tentang transaksi yang terjadi pada *blockchain*, modul Events yang berisi tentang logging dari semua *event* yang dikeluarkan oleh *blockchain*. *Smart Contract* dibuat dengan menggunakan Javascript dan

kemudian modul Query digunakan untuk melakukan pencarian terhadap transaksi yang terdapat pada *blockchain*. Alur kerja sistem adalah sebagai berikut, pertama pengguna akan meminta akses untuk melakukan transaksi pada *blockchain*. *blockchain* akan menerima dan mengecek apakah pengguna memenuhi *Rule* dan *Role* yang telah terdefinisi pada ACL, jika tidak, maka sistem akan mengirimkan pesan error, jika terdefinisi maka sistem akan melanjutkan ke tahapan selanjutnya. Sistem kemudian akan memanggil fungsi yang berada pada *smart contract* yang digunakan untuk melakukan transaksi dan menyimpan log pada *blockchain*. Jika gagal, maka sistem akan mengirimkan pesan error. Jika transaksi pada *blockchain* berhasil maka sistem akan memanggil API eksternal untuk memberikan otorisasi pada pengguna untuk masuk.

Pada penelitian ini, sistem berhasil untuk diimplementasikan, dan memiliki latency yang sedikit. Sistem diuji dengan menggunakan program docker dan program Hyperledger Caliper yang merupakan bagian dari program dari Hyperledger Fabric. Beberapa poin yang dapat diambil dari penelitian ini adalah:

- Penelitian ini menggunakan *blockchain* sebagai salah satu alat *logging* agar *log* tidak dapat diubah-ubah;
- Penelitian ini menggunakan Hypeledger Fabric yang banyak digunakan pada jaringan privat. Namun, Hyperledger Fabric merupakan *permissioned blockchain*, berbeda dengan Ethereum yang merupakan *blockchain* yang memberikan setiap *node* kuasa untuk melakukan *mining* terhadap transaksi.
- Sistem menggunakan *Smart Card* yang merupakan salah satu medium akses kontrol fisik yang banyak digunakan, namun juga berarti bahwa *smart card* juga mungkin paling mudah untuk di-*copy* dan digunakan oleh orang lain
- Sistem hanya menentukan *log* berdasarkan ID dari *Smart Card*, namun karena *Smart Card* dapat berpindah tangan, maka masih ada celah untuk pengguna yang menyalahgunakan sistem.

- Sistem menggunakan API eksternal sebagai cara untuk memberikan akses pada pengguna, jika konfigurasi pada API eksternal tidak aman, maka seluruh sistem dapat terkompromi, selain itu dengan adanya sistem API eksternal, maka sistem ini menjadi semi-terdistribusi.
- Pembunyian alarm dan peringatan penyelundup pada sistem merupakan salah satu cara yang baik dalam menanggulangi akses masuk secara paksa.

2.1.3 Blockchain-based Secure Data Storage for Door Lock System [13]

Penelitian "Blockchain-based Secure Data Storage for Door Lock System" yang diteliti oleh Ulfah Nadiya, Muhammad Ilham Rizqyawan, dan Oka Mahnedra meneliti mengenai implementasi *blockchain* pada *smart home*, terutama sistem *door lock.* Pada penelitian ini, *blockchain* yang digunakan adalah Ethereum. Sistem *node* terdiri atas sebuah Raspberry Pi dan USB *webcam*. Cara kerja dari sistem ini adalah sistem pertama mengambil gambar dengan *webcam* apabila mendeteksi ada wajah di depan *webcam* tersebut. *Webcam* kemudian melakukan pengenalan terhadap wajah tersebut dan melakukan pengecekan apakah wajah tersebut terdaftar atau tidak. Jika wajah tersebut tidak terdaftar, maka sistem akan menolak akses. Jika wajah terdaftar, kemudian akan dilakukan pengecekan apakah *webcam* tersebut merupakan *webcam* yang terdaftar di dalam *blockchain* atau tidak, jika tidak, maka sistem akan menolak akses, jika terdaftar, maka sistem akan melakukan transaksi pada blockchain mengenai data dari pengguna yang masuk. Sistem kemudian akan membuka pintu dan kembali mengunci pintu. Data transaksi dari sistem akan berisi mengenai identitas dari orang yang masuk, status akses dan waktu masuk atau keluar serta waktu akses. Sistem juga menggunakan *smart contract* untuk membuat beberapa aturan, yaitu *store transaction* dan *monitor transaction*. *Store transaction* digunakan untuk menyimpan data transaksi ke *blockchain*, sedangkan *monitor transaction* digunakan untuk memonitor data yang terdapat pada jaringan *blockchain*. Pada penelitian ini, dilakukan pengujian

menggunakan sistem *avalanche effect* yang mendapatkan skor berupa 96% dan 100%.

Beberapa poin yang dapat diambil dari penelitian ini adalah:

- Penelitian ini menggunakan sistem biometrik yaitu pengenalan wajah sebagai salah satu cara identifikasi pengguna yang merupakan salah satu sistem yang cukup aman
- Pada penelitian ini, tidak dibahas mengenai bagaimana data wajah disimpan, sehingga tidak dapat diketahui apakah data wajah disimpan secara terpusat atau terdistribusi

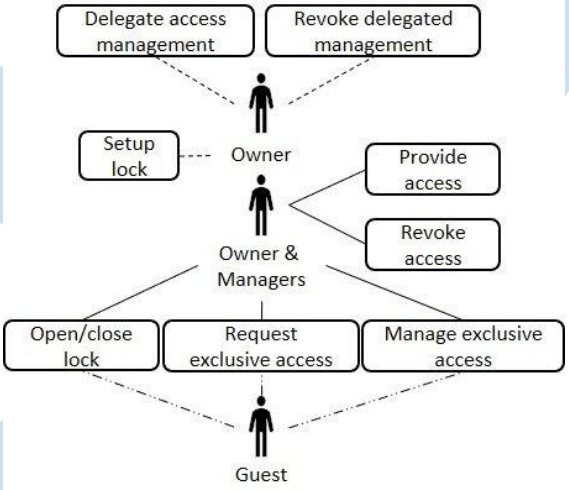
2.1.4 IoT and Blockchain for Smart Locks [14]

Penelitian yang dilakukan oleh Lucas de Camargo, Mayra Samniego dan Ralph Deters, membahas mengenai sistem kunci pintar yang mengintegrasikan antara perangkat IoT, *smart contract* pada *blockchain* dan teknologi web, sehingga menghilangkan keperluan dari pemilik dengan penyewa untuk bertemu, serta memberikan sekuritas dan privasi pada penyewa dengan cara memberikan fitur hanya penyewa yang dapat mengontrol pintu. Sistem pada penelitian ini, ditujukan untuk penggunaan pada AirBnB, dimana pada awalnya pemilik AirBnB harus memberikan kunci terlebih dahulu ke penyewa. Sistem pada penelitian ini menggunakan jaringan *blockchain* Ethereum pada jaringan Testnet dengan menggunakan Infura API. Pada sistem yang dirancang pada penelitian ini, pengguna yang merupakan pemilik, memiliki kemampuan untuk mengelola manajemen akses serta pengguna yang berupa penyewa untuk melakukan akses, selain itu pemilik juga memiliki kewenangan untuk melakukan *setup* kunci pintar tersebut. Sedangkan pengguna, memiliki kemampuan untuk membuka atau mengunci pintu, meminta akses eksklusif dan mengelola akses eksklusif. Akses eksklusif adalah akses dimana hanya orang yang meminta tersebut yang dapat membuka dan menutup pintu. Untuk mengelola manajemen akses tersebut, *smart contract* digunakan, sebuah manajemen website juga dibuat sebagai UI untuk mengelola *smart contract*.

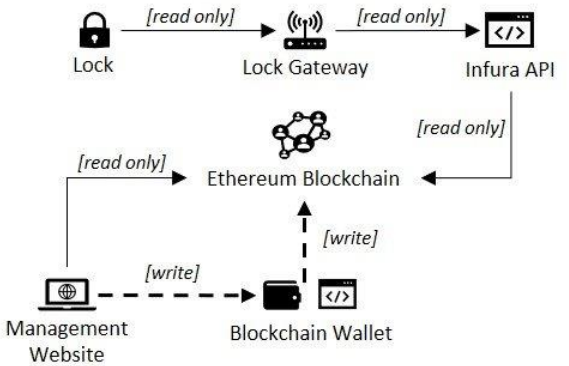
Untuk berinteraksi dengan jaringan *blockchain*, maka digunakan API berbasis HTTP yang terhubung dengan infura. Hasil dari penelitian ini adalah sistem berhasil untuk diimplementasikan, serta memiliki biaya yang cukup murah, dengan biaya yang paling mahal terdapat pada *setup lock* yang memakan biaya sebanyak USD 3.83.

Beberapa poin yang dapat diambil dari penelitian ini adalah:

- Penggunaan web sebagai *interface* dengan *smart contract* merupakan salah satu cara yang baik daripada menggunakan *command line.*
- Biaya yang dikeluarkan cukup murah, sehingga dapat diimplementasi.



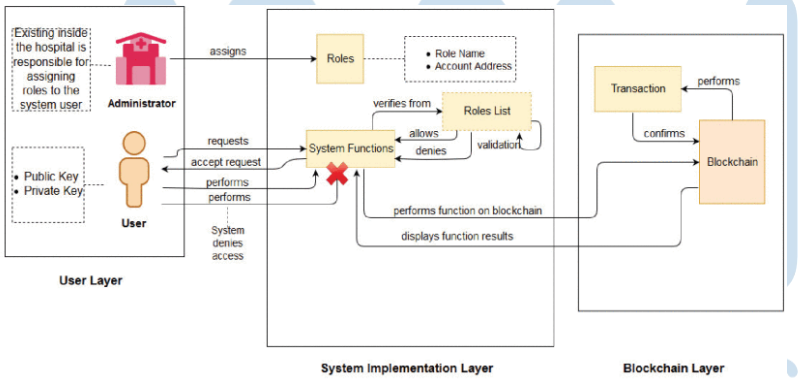
Gambar 2.1 *Diagram use case*



Gambar 2.2 Arsitektur sistem

2.1.5 Using Blockchain for Electronic Health Records [15]

Penelitian berjudul "Using Blockchain for Eletronic Health Records" yang dilakukan oleh Ayesha Shahnaz, Usman Qamar, Ayesha Khalid ini merancang sebuah sistem *framework* yang terdistribusi, yang dimana *framework* ini berisi catatan mengenai pasien dan akses kontrol terhadap orang yang dapat mengakses catatan tersebut. Sistem ini diharapkan dapat menyelesaikan isu mengenai informasi yang asimetrikal dan kebocoran data yang terdapat pada sistem EHR. Sistem *framework* pada penelitian ini terdiri atas 3 layer, yaitu *User Layer*, yang menjadi titik interaksi antara pengguna dengan *blockchain*; *Blockchain Layer* yang berisi tentang transaksi dari jaringan *blockchain*, dan *System Implementation* yang berisi mengenai *smart contract* yang diimplementasikan kedalam *blockchain*. Sistem ini menggunakan jaringan *blockchain* berupa Ethereum. *Smart contract* yang digunakan pada sistem ini adalah *Patient Records*, yang digunakan untuk melakukan CRUD pada catatan pasien serta *role* pada pengguna dan *Roles* yang digunakan sebagai akses kontrol terhadap data catatan dari pasien. Setiap catatan pasien akan disimpan pada IPFS. Cara kerja dari sistem ini akan dijelaskan pada gambar 2.3.



Gambar 2.3 Cara kerja sistem

Sistem ini berhasil untuk diimplementasikan dengan waktu eksekusi per fungsi dari *smart contract* adalah dari 18 detik hingga 1 menit 48 detik.

Poin-poin yang dapat diambil dari penelitian ini adalah:

- Penelitian ini menggunakan *blockchain* sebagai salah satu medium untuk mencatat pasien, pada sistem yang akan dirancang pada penelitian yang dilakukan penulis, sistem catatan pasien dapat diganti menjadi sistem logging
- Catatan pasien disimpan pada IPFS yang merupakan salah satu solusi penyimpanan yang terdistribusi, pada penelitian ini hash dari IPFS dimasukan ke dalam transaksi sehingga setiap catatan menjadi *immutable*
- Sebagai titik interaksi antara pengguna dengan *blockchain*, penelitian ini menggunakan DApp, yang merupakan aplikasi yang menggunakan *blockchain* sebagai tempat penyimpanan data.

2.1.6 Residential access control system using QR code and the IoT [16]

Penelitian yang dilakukan oleh Pak Satanasaowapak, Witawat Kawseewai, Suchada Promlee, dan Anuwat Vilamat, membahas mengenai sebuah sistem yang dapat menambahkan keamanan dan membantu pemilik rumah untuk membawa kunci dengan cara menggunakan *QR Code* dan IoT. Sistem terdiri atas aplikasi Resident Access Control (RAC) yang dibuat untuk sistem operasi Android, yang digunakan untuk membuat *QR Code*, serta perangkat RAC, yang terdiri atas ESPIno32CAM, kunci solenoid, relay, tombol dan LED. Cara kerja dari sistem ini adalah, pertama pengguna akan meminta pembuatan *QR Code* pada aplikasi Resident Access Control (RAC), pengguna meminta dengan memilih nama pintu yang terdapat pada aplikasi. Kemudian, aplikasi akan membuat *QR Code* yang berisi hasil SHA-2 dari IMEI perangkat, nama pintu yang dipilih dan kode random. Kemudian, perangkat RAC akan melakukan pembacaan dari *QR Code* yang telah dibuat pada aplikasi dan melakukan perbandingan terhadap hasil *hash* dari SHA-2 *QR Code*, jika hasil sama, maka sistem akan mengirimkan notifikasi ke aplikasi LINE serta NETPIE mengenai status dari pintu tersebut. Sistem berhasil diimplementasikan dan didapatkan hasil pengujian performa dari sistem yaitu sebesar 5.13 detik. Sistem

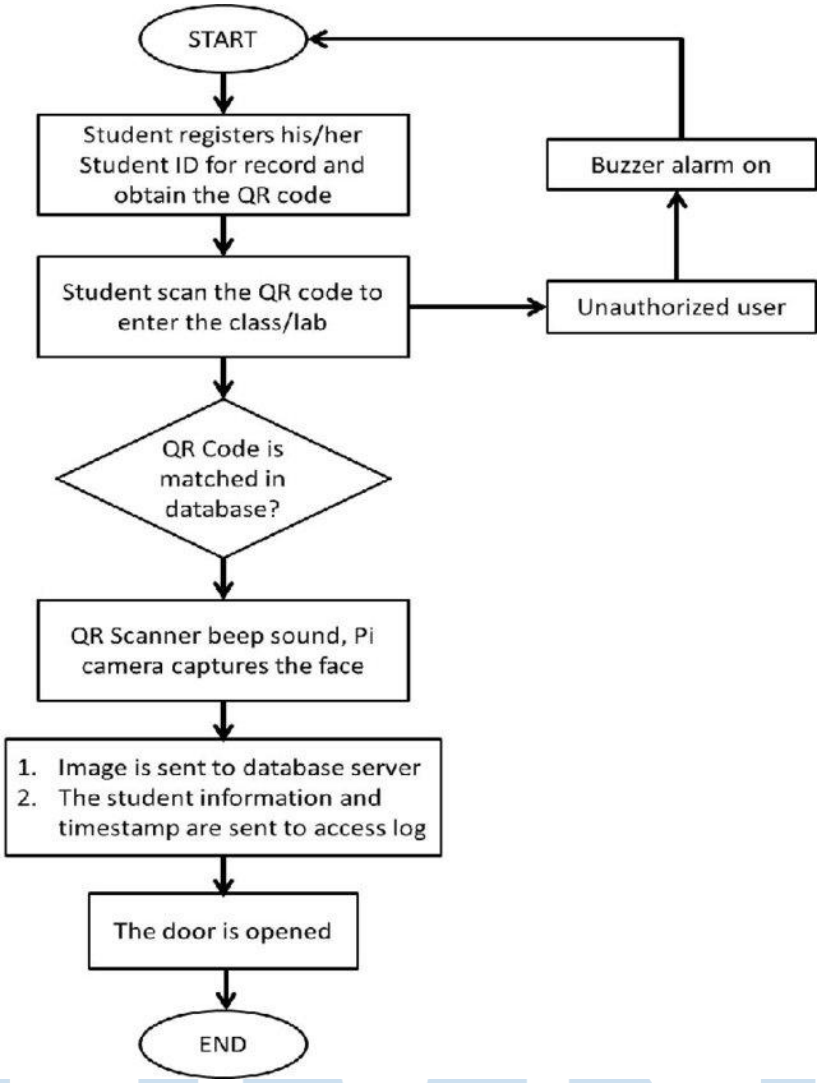
ini juga tidak dapat diserang karena sistem *hash* yang digunakan menggunakan IMEI dan kode random.

Beberapa Poin yang dapat diambil dari penelitian ini adalah:

- Sistem menggunakan *QR Code*, yang merupakan salah satu sistem keamanan yang paling banyak digunakan namun juga yang paling aman
- Sistem menggunakan sistem *hash* pada sistem, yang menambah keamanan dari sistem tersebut.

2.1.7 Development of Web-Based Smart Security Door Using QR Code System [17]

Penelitian yang dilakukan oleh Ahmad Fahmi, Nur Nabila, Habibah Hasim, dan Mohammad A. Saleh, membahas mengenai rancangan sistem sebuah sistem akses menggunakan *QR Code* dan website, untuk memasuki sebuah laboratorium ruangan dengan penambahan sistem logging. Sistem pada penelitian ini, menggunakan Raspberry Pi, Pi *Camera* dan *QR Scanner*. Cara kerja dari sistem adalah sebagai berikut:



Gambar 2.4 Alur sistem kerja penelitian [17]

Pada sistem ini, jika data yang diterima dari *QR* tidak sesuai dengan data yang terdapat pada *database* dan percobaan untuk akses masuk dengan kode *QR* mencapai 3 kali, maka sistem akan membunyikan *buzzer*. Kamera akan digunakan oleh sistem untuk menyimpan gambar dari pengguna yang ingin memasuki ruangan dan kemudian akan disimpan kedalam *database*.

Beberapa poin yang dapat diambil dari penelitian ini adalah:

- Penggunaan Pi Camera dengan Raspberry Pi merupakan salah satu sistem yang dapat diimplementasikan

- Dengan memasukan gambar *user* kedalam logging, maka dapat menambah keamanan dari sistem tersebut
- Percobaan sebanyak 3 kali sebelum *buzzer* dibunyikan dapat menjadi alternatif sebagai pendeteksi dari pengguna yang tidak terdaftar

2.1.8 Securing Face Recognition System Using Blockchain Technology [18]

Penelitian yang dilakukan oleh Saumya Shankar, Jitendra Madarkar, and Poonam Sharma, merancang sistem pengenalan wajah dengan menggunakan *blockchain* sebagai *database* dari *dataset* wajah. Model pengenalan wajah yang digunakan adalah VCG Face, yang merupakan model *machine learning* yang telah di-*training* dengan menggunakan dataset wajah. Algoritma yang dipakai sistem untuk menyimpan wajah pada *blockchain* adalah sebagai berikut: pertama. gambar wajah akan diubah menjadi *grayscale* dan dipecah menjadi vektor 1 dimensi; Kemudian, setiap pixel 1 dimensi ini akan dimasukan kedalam *blockchain*. Ketika akan dilakukan training, maka sistem akan mengambil data dari *block*, kemudian akan merekonstruksi pixel kembali menjadi gambar. Gambar yang telah direkonstruksi kemudian akan digunakan sebagai masukan dari VCG Face. Hasil dari pengujian sistem ini mendapatkan akurasi sebanyak 95% persen pada *dataset* ORL, 87% pada dataset FEI dan 30.86% pada dataset LFW.

Beberapa poin yang dapat diambil dari penelitian ini adalah:

- Penelitian ini merupakan salah satu penelitian yang mencoba untuk menggunakan *blockchain* sebagai salah satu media penyimpanan untuk dataset dari *machine learning*
- Penggunaan *blockchain* dapat digunakan untuk mencegah perubahan terhadap dataset, namun biaya yang diperlukan untuk menyimpan data kedalam *blockchain* akan semakin besar sehingga diperlukan media lain untuk menyimpan data seperti gambar pada *blockchain*

- VCGFace dapat digunakan sebagai salah satu *model* pengenalan wajah, karena memiliki akurasi yang cukup baik untuk digunakan.

2.1.9 FaceHub: Facial Recognition Data Management in Blockchain [19]

Penelitian ini membahas mengenai rancangan sistem yang menggunakan sistem pengenalan wajah untuk partisipasi mahasiswa pada perkuliahan, dengan menggunakan *blockchain* sebagai tempat penyimpanan data. Sistem ini bekerja dengan cara sebagai berikut: Sistem pertama akan mengambil gambar wajah dari pengguna, dimana pada *demo*, sistem mengambil sebanyak 250 wajah pengguna. Wajah tersebut kemudian disimpan ke *database* lokal yang kemudian akan diproses dengan menggunakan algoritma *Haar Cascade*. Gambar dari wajah pengguna sebelumnya akan diubah menjadi file .yml. File .yml ini kemudian akan disimpan pada *blockchain*, yang nantinya akan digunakan sebagai *dataset*. Ketika ada pengguna yang ingin masuk melalui aplikasi, maka sistem akan meminta *user* dan *password* serta pengenalan wajah sebagai sistem verifikasi. Hasil pengujian dari sistem, mendapatkan sistem dapat menghasilkan akurasi sebesar 85%. Berbeda dengan penelitian sebelumnya, pada penelitian ini, sistem mengubah gambar menjadi file .yml, sehingga biaya yang dikeluarkan untuk menyimpan data kedalam *blockchain* akan lebih murah dari sebelumnya, namun masih cukup mahal untuk digunakan sebagai tempat penyimpanan.

2.2 Tinjauan Teori

2.2.1 Blockchain

Blockchain adalah sebuah *database* yang terdesentralisasi, yang berisi blok-blok yang ditandatangani secara digital secara berurutan dan terhubung secara kriptografis dimana blok-blok ini diatur oleh sebuah model konsensus

[20]. *blockchain* pertama kali dibuat oleh Nakamoto Satoshi dengan nama Bitcoin, yang merupakan sebuah versi *peer-to-peer* dari uang digital yang memungkinkan pembayaran untuk dilakukan dari suatu pihak ke pihak lainnya tanpa adanya lembaga finansial [21]. Sistem *blockchain* bekerja berdasarkan *Transaction* atau transaksi. Pada *blockchain* Ethereum, struktur sebuah transaksi terdiri atas penerima dari transaksi, sebuah *signature* dari pengirim transaksi, jumlah Ether yang akan dikirimkan, data yang akan dikirimkan, STARTGAS dan GASPRICE [22]. Transaksi ini kemudian akan dimasukan kedalam suatu blok dari *blockchain*. Sebuah blok *blockchain* berisi atas waktu, nonce, *hash* dari blok sebelumnya dan semua transaksi yang terjadi setelah blok sebelumnya dibentuk. Blok ini kemudian akan disambungkan dari blok yang baru dengan blok yang lama, penyambungan antar blok inilah yang disebut dengan *blockchain.* Blok ini kemudian akan disebarkan ke jaringan dan membentuk sebuah *blockchain*. Untuk memverifikasi sebuah blok, maka setiap node akan melakukan *mining,* dimana jika node berhasil melakukan *mining* pada sebuah blok, maka akan dibayar dengan menggunakan *cryptocurrency* dari *blockchain* tersebut [23]. *Node* pada *blockchain* berkomunikasi satu dengan yang lainnya dengan menggunakan arsitektur P2P yang merupakan arsitektur jaringan *Full Mesh*.

Untuk memvalidasi sebuah blok, terdapat beberapa konsensus yang umum digunakan, antara lain adalah: *Proof of Works (*POW*), Proof of Stake (*POS*) dan Proof of Authority (*POA*).* POW merupakan salah satu konsensus yang paling banyak digunakan dan paling awal digunakan. POW digunakan oleh Ethereum 1.0 dan Bitcoin. POW bekerja dengan cara setiap *miner* akan melakukan *brute force* untuk menghitung *nonce* dari sebuah blok. Untuk melakukan perhitungan *nonce,* maka setiap *miner* akan melakukan perhitungan matematika pada *dataset* yang didapatkan dari *chain* sebelumnya [21], [22]. Jika *nonce* sesuai, maka *miner* kemudian akan mendapatkan *reward* berupa ether. Sedangkan POS merupakan sistem konsensus dimana pengguna diminta untuk melakukan *stacking* terhadap *cryptocurrency* sehingga menjadi *validator* sebuah jaringan [24]. Konsensus POS digunakan pada Ethereum 2.0. *POA*

merupakan algoritma konsensus dimana sebuah blok hanya dapat divalidasi oleh *signer* yang dipercaya [25] . Konsensus POA dipakai pada *private* dan *testnet* Ethereum.

Konsensus PoW merupakan konsensus utama yang dipakai pada Bitcoin dan Ethereum. Konsensus PoW memiliki kelebihan yaitu keamanan yang tinggi dan *availability* yang tinggi [26]. Konsensus PoS merupakan konsensus yang dibuat untuk menggantikan konsensus PoW yang mempunyai beberapa limitasi seperti efisiensi energi. Konsensus PoS memiliki beberapa kelebihan yaitu pembuatan *block* yang tinggi, *throughput* yang besar serta efisiensi yang tinggi [26]. Sedangkan PoA merupakan konsensus yang banyak digunakan pada *private* dan *permissioned blockchain.* PoA memiliki kelebihan seperti lebih efisien terhadap energi dibandingkan PoW dan PoS dikarenakan tidak diperlukannya perhitungan kompleks seperti yang dilakukan pada PoW, dan memiliki *throughput* yang lebih besar dibandingkan PoW dan PoS karena waktu pembuatan *block* yang konstan, selain itu PoA juga memiliki kelebihan yaitu dapat menanggulangi masalah *market-fluctuating processing fees* yang terjadi pada 2 konsensus lainnya dikarenakan PoA membuat biaya tersebut menjadi konstan pada jaringan.

Blockchain dapat dibagi menjadi beberapa jenis yaitu *Public Blockchain, Private Blockchain* dan *Permissioned Blockchain* [27] . *Public Blockchain* merupakan *blockchain* yang dimana pada jaringan *blockchain* ini, setiap orang dapat untuk menjadi *node* dari *blockchain* dan terdapat kode yang *open-source* sehingga dapat dipakai oleh semua orang. *Private Blockchain*, merupakan jaringan *blockchain* yang dimana jaringan tersebut merupakan jaringan yang tertutup dan hanya *node* yang terdapat pada organisasi yang membuat *blockchain* tersebut yang dapat berpartisipasi. *Permissioned Blockchain* merupakan jaringan *blockchain* yang memberikan hanya beberapa *node* yang dapat berpartisipasi pada jaringan tersebut. *Public Blockchain* yang biasa digunakan adalah Ethereum 1.0 dan Bitcoin. *Permissioned Blockchain* yang biasa digunakan adalah Hyperledger Fabric.

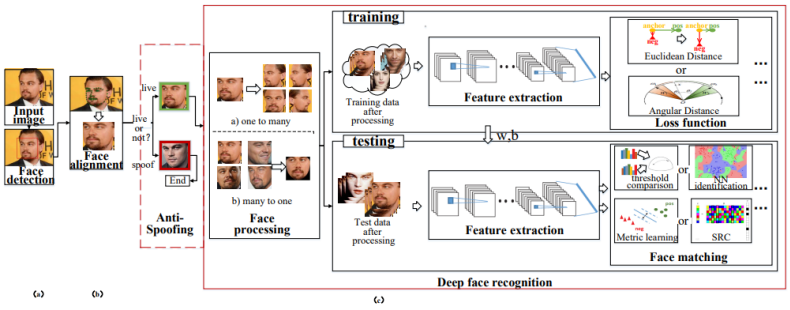
Keamanan dari *blockchain* terhadap *tampering* disebabkan oleh *ledger*, yang merupakan koleksi dari beberapa transaksi, yang terdistribusi. Dengan adanya *ledger* yang terdistribusi ini, maka data dari setiap transaksi yang dimasukan kedalam blok akan disimpan oleh semua *node*, dan jika ada *node* baru yang masuk kedalam jaringan, maka *node* tersebut akan meminta *copy* dari *ledger* tersebut, sehingga untuk mengubah satu transaksi, maka dibutuhkan pengubahan pada semua *node*. Dengan adanya sistem terdistribusi juga, maka *blockchain* berjalan pada *software, hardware* dan infrastruktur jaringan yang berbeda-beda, sehingga *vulnerability* pada satu *node* akan berbeda dengan *node* lainnya. Desain dari sistem *blockchain* yang akan mendeteksi apabila adanya blok yang tidak *valid,* serta jika blok yang dibuat tidak mempunyai referensi ke blok sebelumnya menjadi tidak valid juga membuat sistem akan menjadi *anti-tampering*. Penggunaan *digital signature* dan fungsi kriptografi juga membuat *ledger* menjadi *anti-tamper*. Penggunaan *blockchain* yang hanya *Create* dan *Write* (CR) juga menambahkan *anti-tamper*, karena tanpa adanya *Delete* dan *Edit*, maka tidak akan ada penghapusan ataupun pengubahan terhadap transaksi yang ada di *blockchain*. [28]

2.2.2 Face Recognition

Face recognition atau pengenalan wajah adalah teknologi sains yang mempelajari tentang bagaimana wajah dapat dikenali oleh sistem biologis dan cara pengenalan wajah tersebut dapat disimulasikan pada sistem komputer [29]. Metode-metode yang digunakan untuk pengenalan wajah antara lain adalah *Principal Component Analysis (PCA), Linear Discriminate Analysis (LDA), Support Vector Machine (SVM), AdaBoost, Small Sample, Neural Networks* dan *Deep Learning* [30]. *Deep Learning* merupakan metode yang paling banyak digunakan pada pengenalan wajah. Penggunaan *deep learning* lebih disukai karena *deep learning* dapat beradaptasi dengan menggunakan data yang besar ataupun *dataset* yang kecil, namun salah satu kekurangan dari metode *deep learning* adalah waktu yang banyak diperlukan untuk melakukan *training.*

Beberapa model dari *deep learning* untuk pengenalan wajah adalah DeepFace, VCGFace, FaceNet. FaceNet merupakan salah satu model yang mendapatkan akurasi pengenalan wajah tertinggi, yaitu sebesar 99,63 pada dataset *Labeled Faces in The Wild.* Aplikasi dari sistem pengenalan wajah antara lain adalah sebagai berikut: identifikasi wajah, akses kontrol, sistem keamanan, pengawasan, manajemen multimedia, dsb [31].

Cara kerja dari sistem pengenalan wajah pada metode *deep learning* antara lain adalah sebagai berikut [32]: pertama, dari *input* gambar, akan dilakukan deteksi wajah dengan algoritma. Salah satu algoritma yang sering digunakan untuk deteksi wajah adalah AdaBoost. Setelah itu, hasil deteksi wajah tersebut akan disesuaikan untuk menjadi *input* dari model *deep learning*. Kemudian, gambar wajah tersebut akan diproses dan diagumentasi. Terdapat dua metode augmentasi yaitu "*one to many*" dan *"many to one"*. Metode *one to many* merupakan metode dimana gambar wajah diagumentasi dari satu gambar *input.* Augmentasi tersebut dapat berupa perubahan pose dari satu gambar masukan tersebut. Sedangkan metode *many to one* merupakan metode augmentasi, dimana augmentasi dilakukan dengan cara mengambil beberapa tampilan kanonikal dari satu atau banyak gambar. Setelah gambar wajah diproses, maka gambar wajah akan dimasukan kedalam *model* dan akan dilakukan *training* dan *testing*. *Training* dilakukan agar model dapat mengenal wajah dari hasil *input.* Sedangkan *testing* dilakukan untuk mengetahui akurasi dari sebuah model.



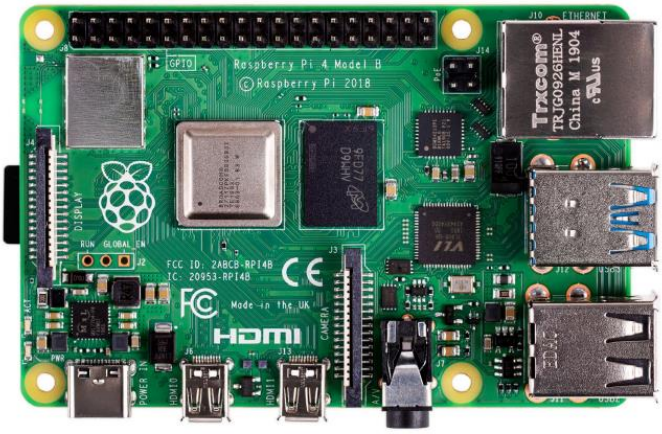
Gambar 2.5 Cara kerja pengenal wajah metode deep learning

2.2.3 Electronic Access Control

Akses kontrol merupakan sebuah sistem elektronik yang memfasilitasi sistem persetujuan perizinan pada portal keamanan tanpa harus adanya petugas untuk memvalidasi pengguna yang ingin memasuki portal [33]. Sebuah akses kontrol elektronik biasanya terdiri atas sensor atau perangkat yang dikontrol, modul pengambil keputusan, jaringan komunikasi, satu atau lebih database, dan satu atau lebih komputer [34]. Beberapa contoh dari sistem akses kontrol elektronik adalah *smart card, keypad,* dan *biometric system.* Sistem akses kontrol elektronik bekerja dengan menggunakan komputer, kredensial, pembaca kredensial dan kunci pintu untuk mengontrol akses secara elektronik[33]. Kredensial merupakan sesuatu yang seseorang punya yang unik, seperti kartu akses, sidik jari ataupun kode password. Pembaca kredensial merupakan sebuah sistem yang membaca kredensial seseorang dan mengirimkan kredensial tersebut untuk dibandingkan dengan data yang terdapat pada database.

2.2.4 Raspberry Pi

Raspberry Pi merupakan *single board computer* yang mendukung berbagai fitur dan banyak digunakan dalam bidang IoT dan *robotic system*. Raspberry Pi merupakan sistem yang dibentuk oleh Raspberry Pi Foundation yang berbasis di Inggris [35]. Raspberry Pi seringkali digunakan sebagai Perangkat IoT karena harganya yang murah dan cukup bertenaga sehingga lebih murah daripada menggunakan komputer. Raspberry Pi berbeda daripada perangkat IoT lainnya karena raspberry Pi dapat menjalankan OS seperti komputer biasanya, dimana perangkat IoT lain seperti Arduino dan Espressif tidak dapat menjalankan OS melainkan RTOS. Sebuah Raspberry Pi terdiri atas sebuah prosessor, RAM, port USB, port Ethernet, modul WiFi dan Bluetooth, dan port GPIO. Raspberry Pi mendukung menggunakan sistem operasi seperti Raspbian atau Raspberry Pi OS, Ubuntu dan Debian.



Gambar 2.6 Raspberry Pi 4 Model B [36]

2.2.5 Smart Contract

Smart Contract merupakan program yang disimpan didalam *blockchain* yang akan dijalankan ketika ada kondisi yang terpenuhi [37]. *Smart contract* sendiri bekerja seperti kontrak fisik, namun berada pada digital. *Smart contract* biasanya ditulis menggunakan bahasa pemrograman Solidity dan Viper. Beberapa keuntungan *smart contract* adalah *smart contract* cepat, efisien dan memiliki akurasi yang baik, karena ketika kondisi terpenuhi, *contract* langsung dapat dijalankan; Terpercaya dan transparan, karena dijalankan pada *blockchain*, maka dapat dipastikan bahwa *smart contract* lebih terpercaya dan lebih transparan; aman, karena setiap transaksi di *blockchain* dan *smart contract* terenkripsi, maka dipastikan bahwa *smart contract* aman. *Smart contract* merupakan basis dari transaksi pada *blockchain* Ethereum. Pada Ethereum, sebuah *smart contract* juga merupakan sebuah akun Ethereum, sehingga memiliki saldo dan dapat mengirimkan transaksi pada jaringan, namun mereka tidak dikontrol oleh pengguna dan hanya disebarkan pada jaringan [38].

2.2.6 QR Code [39]

QR Code adalah sebuah simbol matrix 2 dimensi. *QR Code* pertama kali diciptakan oleh perusahaan Denso pada tahun 1994 dan menjadi standar ISO pada tahun 2000. *QR Code* awalnya digunakan pada kontrol produksi pada bagian otomotif namun akhirnya digunakan pada banyak bidang. *QR Code* banyak digunakan pada berbagai bidang karena beberapa alasan yaitu mempunyai densitas data yang lebih besar dari barcode, dapat digunakan dengan bahasa Jepang dan Mandarin, dapat digunakan secara gratis karena telah dirilis pada *public domain*, dan *QR code* dapat dibaca dengan mudah dengan kamera dari *smartphone*. *QR Code* memiliki karakteristik sebagai berikut: pembacaan kode cepat pada segala arah, tahan terhadap simbol yang terdistorsi, memiliki restorasi data, dan dapat melakukan encoding terhadap Kanji dan Kana. Simbol *QR Code* bervariasi dari versi 1 hingga 40 [40]. Sebuah *QR Code* juga memiliki 4 level *error correction* yaitu level L dengan *correction* sebesar 7%, level M sebesar 15%, level Q sebesar 25% dan level H sebesar 30%.

Gambar 2.7 *QR Code*

2.2.7 IPFS [41]

IPFS merupakan sebuah *peer to peer file system* yang terdistribusi. IPFS sendiri dibuat untuk menggabungkan beberapa sistem *peer-to-peer* seperti BitTorrent, Git, SFS dan DHT menjadi satu platform, sehingga IPFS menjadi suatu sistem baru untuk pendistribusian dan *versioning* data yang terdistribusi. Protokol dari IPFS terdiri atas beberapa sub-protokol yaitu antara lain: *identity, network, routing, exchange, objects, file* dan *naming. Identity* merupakan sub-protokol yang mengelola generasi dari identitas serta verifikasi node, sedangkan *network* mengelola koneksi dengan peer lainnya, dan menggunakan protokol jaringan yang lainnya. Sub-protokol *routing* digunakan IPFS untuk mencari *peer* lainnya pada jaringan dan *peer* yang dapat memberikan sebuah objek. *Exchange* merupakan sub-protokol yang mengelola dari distribusi setiap blok, dimana setiap blok ini berisi data. *Object* merupakan sub-protokol yang berisi *Merkel DAG* yang berguna untuk mengetahui *link* antar objek dan juga *versioning,* yang juga diatur pada sub-protokol dari *files*. *Naming* merupakan sub-protokol yang mengelola sebuah sistem penamaan yang *mutable.* Pada IPFS, jaringan menggunakan protokol *transport* berupa WebRTC atau uTP, sedangkan untuk mengecek autentikasi dari *packet* yang dikirim, IPFS menggunakan HMAC dari *public key* pengirim. Untuk menukar blok antar sesama *peer,* IPFS menggunakan protokol BitSwap. BitSwap merupakan protokol pertukaran blok yang mirip dengan BitTorrent namun pertukaran tidak terbatas pada satu blok saja. Setiap *node* yang menjalankan BitSwap memiliki *ledger* yang menyimpan data transfer antar *peer* yang membuat protokol BitSwap menjadi *anti-tampering.* Penggunaan Merkle DAG juga membuat membuat IPFS menjadi *tamper resistance* dan mentiadakan duplikasi.

Sebuah *item* pada IPFS akan dikenali dengan CID miliknya, CID atau *Content Identifier,* adalah sebuah *hash* yang digunakan untuk menunjuk ke suatu *item* pada IPFS. CID dibuat berdasarkan sebuah *cryptographic hash* yang berarti perbedaan pada suatu konten akan menghasilkan CID yang berbeda dan suatu konten yang sama akan menghasilkan CID yang sama. untuk sebuah item pada IPFS akan dilakukan *multihash,* yang menggunakan beberapa *hash* untuk

melakukan *hashing. Hash* yang biasa digunakan pada IPFS adalah SHA-256 dan Base58. Untuk menandakan *multihash* versi 0, yang merupakan versi yang banyak digunakan, sebuah CID akan ditandakan dengan awalan "Qm" [42]. Sistem CID ini juga membuat setiap file pada IPFS menjadi *immutable* karena dengan adanya metode *hash* yang digunakan pada setiap konten file, maka sebuah file hanya bisa memiliki satu CID dan jika isi file tersebut diubah maka CID dari file tersebut juga akan berubah juga.

Cara kerja dari IPFS adalah sebagai berikut: ketika pengguna melakukan *upload* fail, IPFS akan memecahkan fail tersebut menjadi beberapa bagian yang disebut dengan blok. Setiap bagian, kemudian akan dilakukan *hash*, dimana *hash* tersebut digunakan sebagai *content addressing* dari setiap *block*. Setiap blok kemudian akan disebarkan ke jaringan IPFS, dimana sebuah file akan ditandai dengan sebuah CID dari sebuah Merkle DAG. Ketika pengguna ingin mengambil konten dari IPFS, maka IPFS akan mencari Merkle DAG dari konten tersebut dan membentuk file dari blok-blok tersebut menjadi sebuah fail.

2.2.8 Multi Factor Authentication

Multifactor authentication (MFA) merupakan sebuah proses authentikasi yang aman, yang membutuhkan lebih dari satu teknik autentikasi yang dipilih dari beberapa kategori kredensial yang independen [43]. Pada MFA terdapat 3 grup faktor atau komponen yang digunakan sebagai kredensial antara lain yaitu [44]:

- *Knowledge factor*, yang merupakan sesuatu yang pengguna ketahui;
- *Ownership factor*, yang merupakan sesuatu yang pengguna punya;
- *Biometric factor*, yang merupakan sesuatu yang terdapat pada pengguna.

Pada *single factor authentication*, hanya satu dari beberapa faktor ini yang digunakan. MFA memiliki tujian untuk melakukan validasi kepada pengguna yang sah untuk mengamankan informasi sensitif yang mereka punya dengan memberikan lapisan pertahanan yang berlapis dan membuat individual yang

tidak diberikan otoritas lebih susah untukmelakukan akses [43]. Sebuah sistem MFA yang baik memiliki beberapa karakteristik, antara lain adalah: [43]

- Mudah digunakan,
- Dapat digunakan dimana saja,
- Resistan terhadap serangan,
- Dapat diandalkan,
- Memiliki Single Sign-On.

2.2.9 Apache Cassandra [45]

Apache Cassandra merupakan *database no-SQL* yang terdistribusi. Apache Cassandra pertama kali muncul pada tahun 2008. Apache Cassandra memiliki arsitektur berupa *masterless* yang berarti setiap node memiliki *role* yang sama, sehingga mengurangi single point of failure. Apache Cassandra juga merupakan *database* yang *fault-tolerant,* hal tersebut dapat dicapai dengan adanya *replication* dari data yang disebarkan pada setiap *node* Cassandra yang dapat diatur dengan menggunakan *replication factor*. Untuk membuat sebuah *database* Cassandra menjadi terdistribusi, pengguna hanya perlu menambahkan *node* Cassandra yang baru, tanpa harus adanya *downtime.* Untuk membagi data, maka Apache Cassandra menggunakan sistem *partitioning*, dimana setiap data yang masuk akan didistribusikan ke node secara adil. Untuk membagi data, Apache Cassandra menggunakan *partitioning key* dan *clustering key.* Apache Cassandra dapat terdiri dari beberapa *cluster* dimana sebuah *cluster* merupakan gabungan dari satu atau beberapa *datacenter*. Sebuah *datacenter* merupakan gabungan dari beberapa *node* yang mirip satu dengan yang lainnya.

2.2.10 BigChainDB [46]

BigChainDB merupakan *database* terdistribusi yang memiliki karakteristik dari *blokchain* dan *database* umumnya. BigChainDB pertama kali dirilis pada tahun 2016, dan versi terbaru dari BigChainDB yaitu BigChainDB

2.0 dirilis pada tahun 2018. Arsitektur BigChainDB adalah dengan membangun sebuah *database* terdistribusi dan menambahkan *blockchain* sebagai cara untuk menambahakan karakteristik dari *blockchain* pada *database* tersebut. Untuk membuat BigChainDB menjadi *fault-tolerant* maka BigChainDB menggunakan Byzantine Fault Tolerance yang terdapat pada Tendermint. Untuk membuat sebuah data pada BigChainDB menjadi *immutable*, BigchainDB menggunakan beberapa strategi, yaitu dengan menutup API yang digunakan untuk mengedit atau menghapus data, melakukan replikasi data pada Node dan menggunakan *hash cryptografi* pada setiap transaksi. Salah satu perbedaan mendasar dari versi 1.0 dan versi 2.0 dari BigChainDB adalah pada BigChainDB 1.0 masih terdapat *master node* yang menerima semua data untuk menulis, baru data disebarkan ke *blockchain.* Dengan adanya *master node* tersebut membuat sistem BigChainDB menjadi semi-desentralisasi. Hal tersebut diubah pada BigChainDB 2.0 dengan menggunakan Tendermint sebagai protokol utama. Pada protokol Tendermint, masih terdapat *node* sebagai *primary* yang melakukan penulisan data, namun *primary* node tersebut berubah-ubah setiap waktu.

2.3 Summary

Berdasarkan studi pustaka yang dilakukan penulis, maka rancangan sistem dari penulis adalah sebagai berikut:

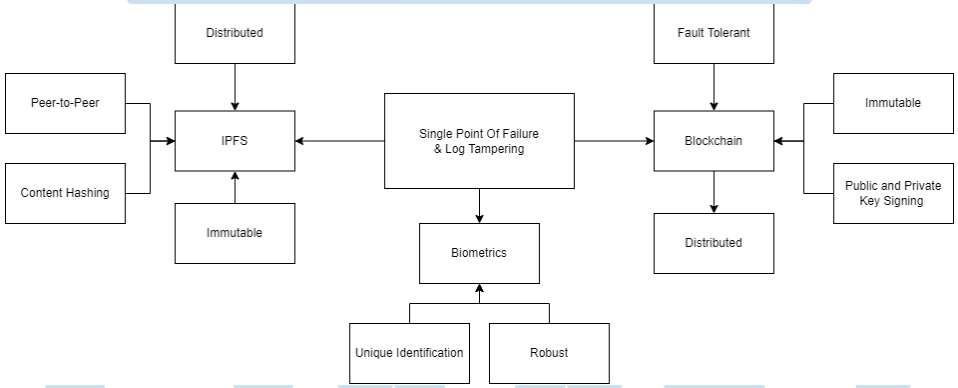
- Sistem akan menggunakan *blockchain* berupa Ethereum, karena Ethereum merupakan salah satu *Public blockchain* yang banyak dipakai dan juga dapat dikonfigurasi menjadi *Private blockchain*. *blockchain* Ethereum juga digunakan pada penelitian [3], [4], [5].
- Sistem akan menggunakan konsensus PoW dan PoA pada pengujian, untuk menguji performa, keamanan terhadap *log-tampering* serta *availability* pada kedua konsensus. PoW dipilih karena PoW merupakan konsensus paling umum digunakan pada *blockchain* sedangkan PoA dipilih karena PoA merupakan konsensus paling umum digunakan pada *permissioned* dan *private blockchain*.

Konsensus PoS tidak digunakan, karena pada geth, tidak terdapat pemilihan untuk konsensus PoS serta PoS juga masih memerlukan PoW untuk melakukan *mining* dan *stacking* pada konsensus tersebut.

- Sistem tidak menggunakan Apache Cassandra karena pada Apache Cassandra, data yang terdapat pada Apache Cassandra merupakan data yang *mutable* yang artinya data tersebut dapat dihapus atau diubah, jika salah satu *node* diserang, dan data tersebut diubah, maka data yang telah ada dapat dihapus, sehingga sistem dapat dilakukan *log-tampering*, selain itu, pada Apache Cassandra, replikasi data dapat menjadi pilihan opsional, sehingga jika terjadi *attack* terhadap sistem, dan *attacker* mengganti *replication factor* sehingga data tidak di-replikasi, maka sistem akan memberikan data yang salah ke pengguna.
- Sistem tidak menggunakan BigChainDB dikarenakan pada BigChainDB, data yang digunakan biasanya adalah data yang kompleks, sedangkan data yang digunakan pada sistem yang dirancang penulis, tidak memerlukan data yang kompleks, selain itu pada BigChainDB, data disimpan pada *database* MongoDB dan transaksi baru disimpan pada *blockchain*, sehingga jika beberapa *node* terkompromi, data masih dapat diubah oleh *attacker.* Selain itu, juga masih terdapat *primary node* pada protokol Tendermint yang membuat sistem semi-desentralisasi, selain itu, pemilihan primary *node* tendermint juga masih dapat diprediksi.
- Sistem akan menggunakan Raspberry Pi sebagai *hardware node.* Raspberry Pi dipakai karena Raspberry Pi merupakan SBC yang mudah untuk digunakan dan didapatkan, selain itu juga mendukung sistem operasi Linux yang banyak digunakan untuk keperluan Embbeded. Raspberry Pi juga memiliki spesifikasi yang cocok untuk melakukan kriptografi yang dibutuhkan *blockchain,* melakukan *face recognition* serta menampilkan UI melalui modul LCD. Raspberry Pi juga digunakan pada penelitian [7]
- Untuk menyimpan data berukuran besar, akan digunakan IPFS, karena penggunaan *blockchain* untuk menyimpan data berukuran besar, akan memakan biaya yang besar, dimana *blockchain* juga dibuat untuk menyimpan data yang ukurannya cukup kecil. IPFS digunakan karena IPFS merupakan jaringan *peer-to-peer* yang memiliki sistem yang mirip dengan sistem *Blockchain.*

- Sistem menggunakan autentikasi MFA untuk mengautentikasi pengguna, *password* digunakan sebagai *knowledge factor* yang digunakan ketika pengguna membuat *QR Code. QR Code* pada *smartphone* digunakan sebagai *ownership factor*, dan wajah pengguna yang digunakan untuk *face recognition* merupakan *biometric factor.*
- *QR Code* dan *face recognition* digunakan karena *QR Code* merupakan salah satu media yang aman dan cepat, *QR Code* juga sering digunakan pada *Wallet* yang menyimpan blockchain sebagai salah satu cara untuk melakukan *sharing* atas *private key* dan *address* dari sebuah akun *Blockchain*, sedangkan sistem *face recognition* merupakan salah satu sistem biometric yang aman dan banyak digunakan.

Berdasarkan *summary* yang dijelaskan diatas, maka dapat dibuat diagram *state-of-art* yang dapat dilihat pada gambar 2.8



Gambar 2.8 Diagram *state of the art*

Permasalahan *single point of failure* dan *log tampering* yang terdapat pada penelitian ini, dapat diselesaikan dengan menggunakan solusi berupa *blockchain*, IPFS dan *biometrics. Blockchain* dipilih menjadi solusi dikarenakan sistem *blockchain* merupakan sistem yang *fault tolerant, immutable*, dibuat dengan menggunakan sistem *public* dan *private key*, serta merupakan sistem yang terdistribusi, sedangkan IPFS menjadi solusi dikarenakan sistem IPFS merupakan sistem yang terdistribusi, berkomunikasi secara *peer-to-peer*, memiliki penyimpanan dengan *content hashing* dan juga setiap data yang disimpan adalah *immutable*. Terakhir, *biometrics* dipilih karena sistem *biometrics* merupakan sistem

yang memiliki identifikasi yang unik untuk setiap pengguna dan merupakan sebuah sistem yang *robust* sehingga akan susah untuk diubah ataupun diserang oleh *attacker*.