

Pemanfaatan Blockchain untuk Pencatatan Transaksi UMKM yang Menjamin Integritas Data

Ensof Barhami
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
Bandung, Indonesia
barhami12345@gmail.com

Dr. Ir. Rinaldi Munir, M.T
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
Bandung, Indonesia
rinaldi@informatika.org

Anggrahita Bayu Sasmita M.T
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
Bandung, Indonesia
anggrahita.bayu@informatika.org

Abstrak—UMKM merupakan sector usaha terbesar di Indonesia. Namun, banyak UMKM yang tidak dapat berkembang karena tidak memiliki modal yang cukup. Untuk melakukan pinjaman modal ke bank, UMKM perlu menyediakan laporan keuangan dari data transaksi UMKM tersebut. Terdapat banyak aplikasi pencatatan transaksi yang dapat digunakan, namun biasanya aplikasi ini memiliki penyimpanan yang *centralized*. Hal ini dapat menimbulkan masalah yaitu data yang tersimpan tidak dapat dijamin integritasnya. Terdapat beberapa cara untuk menangani hal ini, salah satunya menggunakan sistem terdistribusi seperti blockchain yang dapat memastikan data yang sudah tersimpan tidak dapat diubah lagi. Untuk itu, sistem yang akan dibangun pada makalah ini akan menggunakan blockchain. Teknologi blockchain yang akan digunakan adalah Hyperledger Fabric. Sistem ini memiliki fungsionalitas dasar pencatatan transaksi seperti memasukkan data barang, membeli barang, melihat daftar barang terjual, dan melihat data barang pribadi yang menjamin data tidak dapat diubah dengan cara yang tidak diinginkan.

Kata kunci—UMKM; Pencatatan Transaksi; Sistem Terdistribusi; Blockchain

I. PENDAHULUAN

UMKM merupakan salah satu kontributor terbesar di dalam bidang ekonomi Indonesia dan juga Dunia. Pada tahun 2017, menurut Kementerian Koperasi dan UKM RI, UMKM di Indonesia memiliki pangsa pasar sebesar 99,99% atau sebesar 62,9 juta unit dari total pelaku usaha di Indonesia

Namun, UMKM sulit mendapatkan pinjaman modal dari bank karena berbagai faktor syarat-syarat pinjaman. Menurut Kementerian Koperasi dan UKM pada tahun 2014 terdapat lebih dari 50 juta pelaku UMKM yang tidak bisa melakukan pinjaman bank karena dianggap belum produktif atau sudah produktif namun belum layak. Salah satu faktor terbesar sulitnya mendapat pinjaman menurut Jamkrindo adalah pembukuan keuangan yang tidak dilakukan dengan baik oleh UMKM sehingga bank menjadi sulit mengukur kesanggupan UMKM tersebut.

Pembukuan keuangan yang dilakukan UMKM menurut Jamkrindo masih belum tertib. Sehingga bank atau pihak lain akan sulit mengetahui kredibilitas dari UMKM yang melakukan peminjaman modal.

Pembukuan keuangan dapat dilakukan dengan mulai melakukan pencatatan transaksi yang terjadi pada suatu usaha. Terdapat banyak teknologi yang dapat digunakan untuk melakukan pencatatan transaksi ini. Misalnya menggunakan buku catatan tradisional dan aplikasi *point of sale*.

Penggunaan buku catatan memiliki kekurangan yaitu data yang tercatat bisa saja salah tulis. Kemudian buku tersebut dapat mudah hilang atau rusak sehingga data yang disajikan bisa tidak akurat. Kemudian bisa juga digunakan aplikasi *point of sale* untuk menghindari data hilang. Ada beberapa jenis aplikasi semacam ini, yang paling sering ditemukan adalah *point of sale* dengan penyimpanan data di server tersentralisasi seperti Mokapos. Solusi ini juga memiliki beberapa kekurangan, salah satunya adalah data yang dimasukkan ke dalam sistem bisa saja diubah-ubah untuk keperluan peminjaman atau keperluan lainnya. Hal ini tentu menimbulkan masalah kepercayaan (*trust issue*) terhadap data yang dimiliki UMKM.

Untuk mengatasi hal tersebut, salah satu solusi yang bisa digunakan adalah dengan menggunakan salah satu sistem terdistribusi yaitu blockchain. Blockchain memungkinkan data yang sudah masuk ke dalam sistem tidak dapat diubah lagi, bahkan oleh pemilik atau *creator* dari data tersebut. Saat ini, terdapat beberapa aplikasi pencatatan transaksi yang memanfaatkan teknologi blockchain seperti Kendi dan Tokoin. Namun sayangnya, desain sistem yang mereka buat tidak benar-benar terdistribusi, karena data yang masuk diolah oleh mereka sendiri. Begitu juga token yang tersedia di dalam sistem yang *lifecycle* nya juga ditentukan oleh mereka. Selain itu, masih ada permasalahan yang muncul, yaitu pengguna aplikasi tetap dapat memasukkan data transaksi secara manual ke dalam sistem. Hal ini lagi-lagi menimbulkan *trust issue*, yaitu data penjualan yang dimasukkan ke dalam sistem bisa saja bukan dari transaksi yang benar-benar terjadi antar penjual dan pembeli.

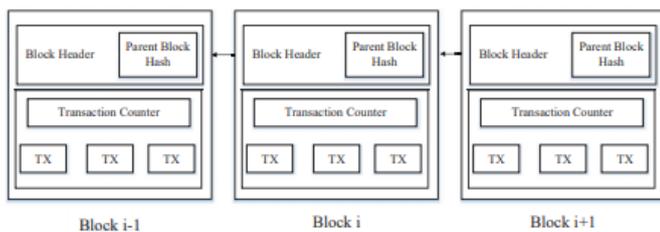
Untuk mengatasi masalah-masalah tersebut, dapat dirancang sistem pencatatan transaksi UMKM sedemikian hingga menjamin integritas data—data transaksi tidak dapat diubah oleh pihak yang tidak memiliki izin atau oleh cara-cara yang tidak diinginkan. Sistem ini harus memiliki fungsionalitas-fungsionalitas dasar untuk melakukan pencatatan transaksi UMKM seperti memasukkan data barang yang ingin dijual, membeli barang, melihat daftar barang terjual, dan melihat daftar barang milik pribadi di dalam sistem.

II. BLOCKCHAIN

A. Blockchain

Blockchain adalah struktur data yang terdiri dari banyak blok dan saling terhubung dengan kriptografi dalam jaringan peer-to-peer. Setiap blok dalam jaringan ini menyimpan data transaksi, timestamp, hash-key blok sebelumnya dan juga memiliki hash-key untuk dirinya sendiri, sehingga membentuk rantai hash secara kriptografi [1]. Rangkaian blok ini semuanya secara keseluruhan kemudian didistribusikan pada tiap *peer* yang ada pada jaringan blockchain, sehingga membuat data transaksi di dalam blockchain resisten terhadap modifikasi. Jika terjadi modifikasi data pada salah satu blok, maka hash-key blok tersebut berubah dan membuat blok lain yang terhubung secara matematis menggunakan hash-key akan mendeteksi perubahan ini dan menolak modifikasi yang dilakukan.

Rangkaian blok pada blockchain memiliki struktur seperti pada Gambar 1. Setiap blok menyimpan data transaksi seperti pada catatan transaksi konvensional biasanya. Rangkaian blok ini terhubung dengan cara menyimpan kunci *hash* blok sebelumnya (*parent block hash*). Sedangkan blok awal yang tidak memiliki *parent* disebut *genesis block* tidak menyimpan *hash* blok sebelumnya.



Gambar 1 Rangkaian Blok pada Blockchain [2]

Blockchain dapat dibedakan menjadi dua jenis berdasarkan *visibility* datanya yaitu Blockchain Publik dan Blockchain Privat. Blockchain publik menyimpan data transaksi yang dapat dilihat oleh semua peserta di dalam jaringan blockchain. Sedangkan Blockchain privat datanya hanya dapat dilihat pihak-pihak yang memenuhi aturan-aturan tertentu.

Blockchain juga dapat dibedakan menjadi dua jenis berdasarkan hak akses ke dalam jaringannya yaitu *Permissioned Blockchain* dan *Permissionless Blockchain*. *Permissionless Blockchain* memungkinkan siapapun untuk dapat bergabung ke dalam jaringan, sedangkan pada *Permissioned Blockchain*, untuk dapat bergabung memerlukan

izin terlebih dahulu dari system untuk bisa menjadi peserta di dalam jaringan.

B. Smart Contract

Smart contract adalah program yang dapat melakukan eksekusi atau kontrol terhadap suatu dokumen atau objek lain sesuai dengan aturan-aturan tertentu seperti kontrak pada umumnya. *Smart contract* digunakan untuk mengontrol *lifecycle* dari objek di dalam sistem—penciptaan objek, transfer objek, hapus objek, dan sebagainya .

Salah satu bentuk primitif dari smart contract adalah vending machine [3]. Vending machine memiliki aturan-aturan yang harus dipenuhi untuk dapat digunakan. Misalnya jika ada pengguna yang ingin membeli minuman seharga Rp.5000, pengguna tersebut harus memiliki uang lebih besar atau sama dengan Rp.5000. Jika uang yang dimasukkan ke vending machine tidak memenuhi aturan itu, maka minuman tidak akan keluar. Kemudian jika uang yang dimasukkan melebihi harga yang ditentukan, maka vending machine akan mengeluarkan minuman tersebut beserta uang kembalian yang sesuai. Proses-proses ini terjadi karena vending machine mendefinisikan beberapa aturan-aturan dan fungsionalitas di dalam sistemnya seperti fungsi untuk mengeluarkan minuman dan kembalian, fungsi mengembalikan uang jika uang rusak dan lain-lain. Begitu juga yang terjadi pada *smart contract*, terdapat aturan-aturan yang telah disepakati oleh pengguna yang mengatur transaksi yang terjadi antar pengguna di dalam sistem.

Kini *smart contract* banyak digunakan pada teknologi *blockchain*. Hal ini memungkinkan aturan-aturan tertentu dapat diterapkan dalam menangani transaksi yang terjadi pada blockchain. Terdapat dua teknologi blockchain yang memanfaatkan *smart contract* ini yang juga sangat sering digunakan pada level industry yaitu Hyperledger Fabric dan Ethereum.

C. Hyperledger Fabric

Hyperledger Fabric adalah *platform distributed ledger technology open-source* yang merupakan *permissioned blockchain* [4]. Hyperledger Fabric adalah teknologi yang dikembangkan oleh Linux Foundation yang memang dibuat modular untuk keperluan industri.

Hyperledger Fabric memiliki smart contract yang disebut *chaincode*. Smart contract pada Hyperledger Fabric dapat ditulis menggunakan *general purpose programming language* seperti Java, Go dan Node.js. Hyperledger Fabric juga menggunakan pendekatan yang berbeda dengan smart contract pada umumnya yaitu *execute-order-validate*. Transaksi dilakukan terlebih dahulu pada smart contract, kemudian transaksi di-order melalui *consensus protocol* tertentu, dan terakhir divalidasi menggunakan *endorsement policy* sebelum ditulis pada ledger.

Hyperledger Fabric memiliki beberapa komponen utama sebagai berikut:

1. Asset

Aset adalah representasi objek yang memiliki nilai. Hyperledger Fabric menyimpan aset dalam bentuk *key-value*.

2. Shared Ledger

Terdapat komponen bernama ledger yang berfungsi mencatat state dari aset-aset pada Hyperledger Fabric.

3. Chaincode

Chaincode adalah smart contract dari Hyperledger Fabric. Chaincode berisi potongan kode yang mengatur business logic dari system.

4. Peer nodes

Pada Hyperledger Fabric terdapat elemen fundamental yaitu peer. Peer merupakan komponen yang menjalankan chaincode dan memegang ledger.

5. Channel

Channel adalah struktur logis berupa koleksi dari beberapa peer di dalam jaringan. Semua peer di dalam satu channel yang sama memiliki ledger yang sama juga.

6. Organization

Hyperledger Fabric memiliki anggota-anggota berupa organisasi. Organisasi-organisasi ini memiliki satu atau lebih peer di dalamnya. Sebuah jaringan Hyperledger Fabric bisa berjalan karena terdapat organisasi-organisasi yang berkontribusi menyumbang sumber daya nya untuk menjalankan jaringan secara bersama-sama.

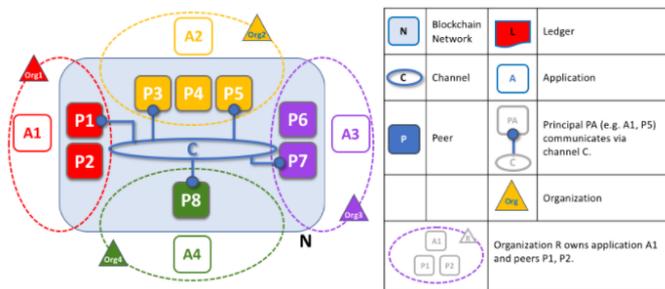
7. Membership Service Provider (MSP)

MSP diimplementasikan sebagai Certificate Authority yang berfungsi sebagai pemberi identitas kepada anggota di dalam jaringan agar bisa bertransaksi. Sebelum dapat mengakses jaringan, pengguna harus terlebih dulu mendapatkan identitas yang dibangkitkan oleh Certificate Authority.

8. Ordering Service

Ordering service membungkus transaksi-transaksi di dalam jaringan untuk kemudian dikirimkan ke peers yang ada di dalam channel yang sama.

Komponen-komponen tersebut membentuk struktur seperti yang dapat dilihat pada Gambar 2. Pada gambar tersebut terdapat aplikasi yang menjadi perantara pengguna untuk mengakses jaringan melalui suatu channel tertentu.



Gambar 2 Arsitektur Hyperledger Fabric dengan 4 Organisasi [4]

D. Ethereum

Ethereum merupakan salah satu teknologi blockchain yang populer. Ethereum merupakan cryptocurrency platform terbesar kedua setelah Bitcoin yang menggunakan mata uang digital yang disebut Ether. Ethereum didesain sebagai public permissionless blockchain. Sama seperti Hyperledger Fabric, Ethereum merupakan teknologi blockchain yang memiliki smart contract, sehingga dapat digunakan untuk berbagai macam keperluan.

Ethereum didesain memiliki lima prinsip sebagai berikut [5] :

1. Simplicity

Ethereum dibuat sesederhana mungkin sedemikian hingga mudah untuk diimplementasi. Optimisasi yang membuat Ethereum menjadi kompleks, sebisa mungkin dihindari.

2. Universality

Filosofi dasar dari Ethereum adalah teknologi ini tidak memiliki fitur spesifik. Namun Ethereum menyediakan scripting language yang bersifat turing-complete yang bisa digunakan untuk membangun smart contract. Dengan kata lain, dengan Ethereum programmer dapat membuat berbagai macam aplikasi yang memanfaatkan blockchain.

3. Modularity

Ethereum didesain semodular mungkin. Jika terdapat bagian yang dimodifikasi, keseluruhan system harus tetap bisa berjalan tanpa mengubah juga semua bagian yang lainnya.

4. Agility

Ethereum akan terus dikembangkan dengan modifikasi-modifikasi sesuai kebutuhan.

5. Non-discrimination dan non-censorship

Protokol pada Ethereum dibuat dengan tidak membatasi atau melarang beberapa spesifik kegiatan. Misalnya jika programmer mengimplementasikan infinite loop di dalam sistem, Ethereum tetap akan berjalan tanpa melakukan blocking terhadap kegiatan atau instruksi tersebut.

III. RANCANGAN SOLUSI

A. Pemilihan Teknologi

Salah satu masalah dalam pencatatan transaksi adalah menjamin integritas data. Integritas data yang dimaksud adalah memastikan data yang tercatat merupakan data valid yang tidak dapat diubah oleh pihak atau cara yang tidak diinginkan. Terdapat beberapa cara untuk memastikan integritas data ini, misalnya menggunakan enkripsi kunci public dan kunci privat kriptografi.

Cara nya adalah dengan melakukan *signing* pada transaksi yang terjadi menggunakan kunci privat yang dimiliki seorang pengguna di dalam sistem. Kemudian sistem akan membaca transaksi ini menggunakan kunci publik yang dimiliki pengguna yang menginisiasi transaksi tersebut. Dengan cara ini, dapat dipastikan suatu transaksi adalah benar dari seorang pengguna tertentu, dan pengguna lain tidak dapat mengubahnya selama tidak memiliki kunci privat pengguna tersebut.

Namun, cara ini memiliki kekurangan yaitu data transaksi yang sudah tersimpan di dalam sistem dapat diubah oleh pengguna yang memiliki data tersebut. Hal dapat diatasi dengan teknologi blockchain. Data yang sudah tersimpan di dalam blockchain tidak dapat diubah lagi karena setiap blok di dalam jaringan saling terhubung dengan kriptografi. Memodifikasi data satu blok memerlukan modifikasi terhadap seluruh blok di dalam jaringan dan membutuhkan perhitungan

yang rumit sehingga hamper tidak mungkin data yang sudah tersimpan dapat diubah lagi.

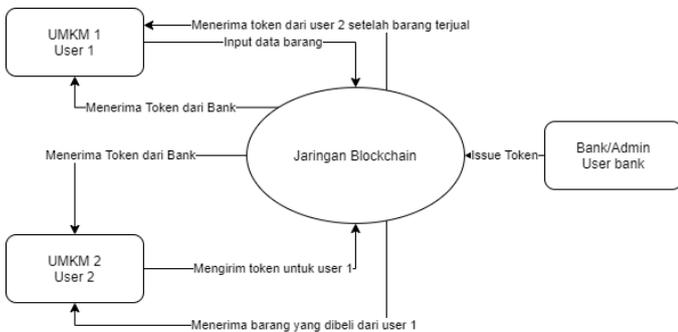
Blockchain yang akan digunakan adalah Hyperledger Fabric. Hyperledger Fabric digunakan karena secara desain awal sudah merupakan *permissioned blockchain*. Fitur ini berguna agar sistem juga dapat memastikan identitas setiap pengguna di dalam jaringan. Tidak semua orang dapat masuk ke dalam jaringan tanpa izin sistem. Selain itu Hyperledger Fabric memiliki *smart contract* yang akan berguna untuk mengatur *lifecycle* dari barang yang diperjualbelikan di dalam sistem pencatatan transaksi UMKM ini.

B. Gambaran Umum

Sistem yang akan dibangun selain harus memastikan integritas data, juga harus memiliki fitur-fitur pencatatan transaksi dasar seperti input data barang yang ingin dijual, beli barang, lihat daftar barang yang sudah dibeli dan lain-lain. Selain itu, untuk dapat memastikan transaksi valid, akan digunakan token di dalam sistem sebagai alat pembayaran. Token ini hanya dapat *diissue* oleh pihak bank dan dikirimkan kepada pengguna di dalam jaringan.

Terdapat dua *role* pada sistem, yaitu pengguna yang merupakan UMKM atau hanya pembeli biasa, dan admin yang merupakan bank-bank yang sudah disepakati di awal untuk menjalankan jaringan blockchain ini.

Pada Gambar 3 dapat dilihat gambaran umum sistem yang akan dibangun. Terdapat *role* pengguna yaitu User 1 dan User 2, dan juga *role* admin yaitu bank.

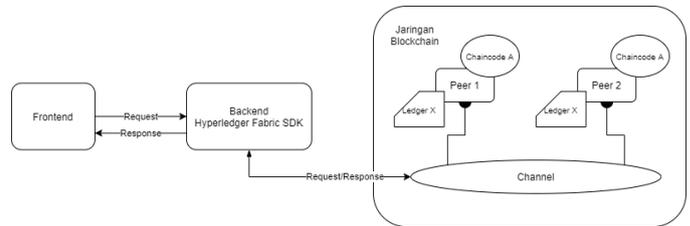


Gambar 3 Gambaran Umum Sistem

Pada gambar tersebut bank dapat mengirimkan token sejumlah tertentu kepada User 1 dan User 2. User 1 melakukan input barang ke dalam sistem, kemudian User 2 membeli barang tersebut dan mengirimkan token seharga barang ke sistem. Token tersebut akan diteruskan ke User 1 dan barang yang diperjualbelikan tersebut akan beralih pemilik dari User 1 menjadi User 2.

C. Arsitektur Sistem

Sistem yang dibangun terdiri atas tiga komponen, yaitu *frontend*, *backend*, dan *jaringan blockchain*. *Backend* dan *blockchain* akan dijalankan oleh pihak bank-bank yang sudah disetujui di awal. Sedangkan pengguna UMKM hanya akan mengakses *frontend*. Arsitektur sistem yang digunakan dapat dilihat pada Gambar 4.



Gambar 4 Arsitektur Sistem Secara Umum

Pada gambar tersebut, pengguna mengakses *frontend* dan mengirimkan *request* kepada *backend* melalui REST API. *Backend* menggunakan SDK yang sudah disediakan oleh Hyperledger untuk mengirim *request* ke jaringan blockchain. SDK yang disediakan dapat mengakses *channel* tertentu di dalam jaringan, dan memerintahkan *chaincode* yang ada pada *channel* tersebut untuk menjalankan fungsi-fungsi tertentu sesuai *request* dari SDK.

Setelah *request* sampai ke jaringan Hyperledger Fabric, Hyperledger Fabric akan menjalankan perintah itu dan mengembalikan hasilnya berupa *response* ke SDK dan akan diteruskan kepada *frontend*.

IV. IMPLEMENTASI DAN PENGUJIAN

A. Implementasi

1. Blockchain

Blockchain menggunakan Hyperledger Fabric. Hyperledger Fabric memiliki komponen utama yang dapat dibuat sesuai kebutuhan yaitu *chaincode*. *Chaincode* merupakan *smart contract* pada Hyperledger Fabric. *Chaincode* ini kemudian akan mengatur transaksi yang terjadi pada jaringan blockchain. Untuk memenuhi kebutuhan sistem, dilakukan implementasi beberapa fungsionalitas pada *chaincode* sebagai berikut :

- Register
 - Mendaftarkan pengguna dengan inisiasi atribut *username*, nama depan, nama belakang dan jumlah token yang dibuat nol di awal.
- Create Asset
 - Buat asset yang akan dijual beserta dengan kategori dan harga nya.
- Buy Asset
 - Membeli asset orang lain. *Chaincode* akan menolak jika asset yang coba dibeli adalah milik sendiri.
- GetMyAsset
 - Melihat semua asset yang dimiliki pengguna.
- GetIssuedAsset
 - Melihat semua asset orang lain yang belum dibeli.
- GetReportAdmin
 - Melihat history transaksi dari pengguna.
- GetMyReport
 - Melihat history transaksi diri sendiri.
- IssueToken

Mencetak token sejumlah tertentu. Hanya bisa dilakukan oleh admin (bank)

- SendToken

Mengirim token sejumlah tertentu. Hanya bisa dilakukan oleh admin (bank)

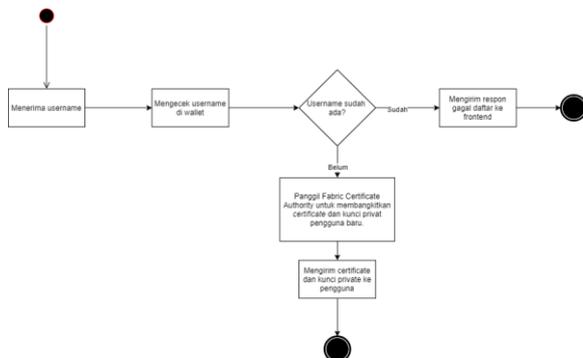
2. Backend

Backend pada sistem ini diimplementasikan menggunakan Node.js karena SDK yang digunakan untuk berkomunikasi dengan chaincode pada blockchain juga menggunakan Fabric SDK Node.

Backend yang diimplementasikan akan dijalankan oleh pihak bank-bank yang telah disetujui di awal. Sama seperti jaringan blockchain yang juga dijalankan oleh mereka.

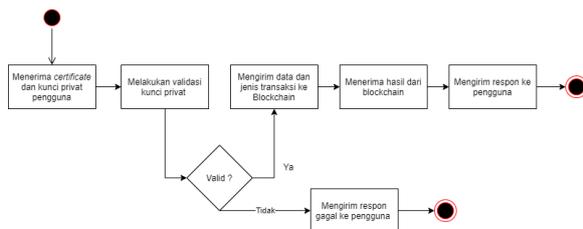
Backend berfungsi sebagai penghubung antara frontend dengan jaringan blockchain dengan cara menyediakan endpoint REST API yang dapat diakses oleh pengguna melalui frontend.

Terdapat dua alur yang disediakan backend pada sistem ini seperti yang dapat dilihat pada Gambar 5 dan Gambar 6 berikut.



Gambar 5 Alur Pendaftaran

Pada Gambar 5, Backend menerima username dari pengguna dan melakukan pengecekan di wallet yang disediakan oleh Node SDK. Jika username sudah ada, maka pengguna tidak dapat melakukan registrasi. Jika username belum ada, maka akan dipanggil certificate authority fabric untuk membangkitkan certificate dan kunci private untuk pengguna agar dapat melakukan login.



Gambar 6 Alur Request Transaksi

Gambar 6 menunjukkan alur transaksi mulai dari diterimanya request dari frontend sampai mengirimnya ke jaringan blockchain. Transaksi apapun yang dilakukan pengguna melalui frontend akan divalidasi oleh backend. Jika kunci privat dan certificate yang dikirim oleh pengguna valid, maka backend akan meneruskan permintaan transaksi dari pengguna ke blockchain menggunakan fungsi yang sudah disediakan Fabric Node SDK yaitu contract.submitTransaction(). Kemudian backend akan menunggu respon dari blockchain dan meneruskan respon tersebut ke pengguna.

3. Frontend

Frontend diimplementasikan menggunakan Vue.js berbasis single page app. Frontend adalah satu-satunya komponen di dalam sistem yang dapat diakses oleh pengguna UMKM. Frontend hanya dapat memanggil fungsionalitas yang disediakan oleh backend.

B. Pengujian

Pengujian dilakukan dengan menguji kebutuhan fungsional dan non fungsional sistem.

1. Pengujian Kebutuhan Fungsional

Pengujian dilakukan dengan metode blackbox testing yaitu pengujian dengan mencoba fitur tanpa melihat atau mempertimbangkan kode program.

Kasus-kasus yang diujikan adalah sebagai berikut:

- Registrasi pengguna A dan pengguna B
- Login Pengguna A
- Issue Token oleh bank
- Kirim token ke pengguna B oleh bank
- Pengguna A menginput barang yang ingin dijual
- Pengguna A melihat daftar barang nya di dalam sistem.
- Pengguna B membeli barang.
- Pengguna B melihat daftar barang yang sudah dibeli
- Pengguna A melihat daftar barang yang sudah terjual
- Cek token yang dimiliki Pengguna A dan Pengguna B. Pastikan total token benar.

2. Pengujian Kebutuhan Non-Fungsional

Pengujian ini bertujuan menguji kebutuhan non fungsional yaitu Data Integrity. Dilakukan beberapa percobaan untuk merubah data yang masuk ke dalam jaringan yang seharusnya tidak boleh terjadi. Pengujian ini dilakukan dengan percobaan sebagai berikut:

- Mengubah data transaction history milik sendiri
- Mencoba memasukkan transaksi baru ke transaction history milik sendiri.
- Mencoba membeli aset milik sendiri.

Ketiga percobaan di atas dilakukan dengan cara *invoke* chaincode secara langsung. Ketiga nya tidak dapat dilakukan karena sudah *dihandle* di sisi chaincode.

Seluruh pengujian fungsional dan non-fungsional dari sistem sudah terpenuhi. Namun, terdapat kekurangan yaitu untuk memenuhi pengujian tersebut, terdapat asumsi yang harus terpenuhi. Asumsi tersebut adalah sistem berjalan pada satu mesin lokal saja. Dengan kata lain di dunia nyata hanya ada satu bank saja yang memiliki banyak peer. Pengujian dengan lingkungan produksi yang nyata mungkin membutuhkan skenario-skenario tambahan dalam pengujian sistemnya untuk memastikan kebutuhan-kebutuhan di atas terpenuhi.

V. KESIMPULAN

Untuk membangun sistem pencatatan transaksi UMKM yang menjamin integritas data, dapat dibangun sebuah sistem seperti desain sistem pada makalah ini. Sistem ini sudah berhasil diujikan. Berikut beberapa hal yang dapat dilakukan sistem yang dibangun ini:

1. Sistem ini menyediakan fungsionalitas untuk mengelola lifecycle dari data transaksi UMKM. Dimulai dari pembuatan data barang, pertukaran pemilik barang dengan transaksi jual beli, dan pencatatan data transaksi jual beli yang valid.
2. Sistem dapat menjamin data yang sudah tersimpan tidak dapat diubah lagi bahkan oleh pemilik data tersebut. Hal ini merupakan kelebihan dari blockchain dibanding dengan penggunaan kunci publik dan kunci privat kriptografi saja. Penggunaan kunci privat dapat memastikan transaksi hanya dapat dilakukan oleh

pemilik dan tidak dapat diubah orang lain, namun tidak bisa memastikan apakah data yang sudah disimpan dapat diubah oleh pemiliknya atau tidak.

3. Sistem ini dapat memastikan tidak semua orang dapat melakukan transaksi. Pengguna harus terlebih dahulu mendaftar dan mendapat certificate yang valid. Selain itu, untuk dapat bertransaksi di dalam sistem, pengguna harus menggunakan token yang hanya bisa dicetak oleh pihak bank dan hanya valid untuk transaksi pada sistem ini. Hal ini dapat membantu bank memastikan terlebih dahulu siapa saja pengguna yang dapat mengakses sistem.

ACKNOWLEDGMENT

Terimakasih kepada Bapak Dr.Rinaldi Munir, MT.dan Bapak Anggrahita Bayu Sasmita, MT. atas bimbingan dan semua masukan selama proses penelitian ini berlangsung. Terimakasih juga kepada seluruh teman yang membantu berdiskusi dan menginspirasi saya untuk menyelesaikan desain solusi pada makalah ini.

REFERENCES

- [1] Chang, Y. Blockchain Technology for e-Marketplace. 2019
- [2] Zheng et al, "An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends". 2017
- [3] Szabo, N. The Idea of Smart Contracts. 1997.
- [4] HyperledgerFabric. "Readthedocs". [Online] <https://hyperledger-fabric.readthedocs.io/en/release-2.2/whatis.html>. 2020
- [5] Ethereum. "Ethereum Whitepaper". [Online] <https://ethereum.org/en/whitepaper/>. 2020