BAB 3

METODOLOGI PENELITIAN

3.1 Metodologi Penelitian

Berikut adalah metodologi penelitian yang dilakukan.

1. Studi Literatur

Pada tahap ini, studi literatur terkait Hyperledger Fabric dilakukan untuk memahami cara kerja, konfigurasi *blockchain*, dan penggunaan SDK. Literatur yang dipelajari berasal dari jurnal, dokumentasi Hyperledger Fabric, dan *website* pendukung lainnya.

2. Perancangan

Pada tahap ini, dilakukan perancangan fitur-fitur yang ingin diimplementasikan pada Oricon berdasarkan observasi terhadap solusi aplikasi serupa yang telah ada. Perancangan pada sistem, *flowchart* aplikasi, dan *user interface* kemudian dilakukan sebelum melakukan implementasi. Aplikasi yang dirancang merupakan aplikasi *mobile cross-platform*.

3. Implementasi

Pada tahap ini, implementasi Hyperledger Fabric, REST API, dan *user interface* dilakukan berdasarkan hasil perancangan yang dibuat.

4. Pengujian dan Evaluasi

Pada tahap ini, pengujian terhadap perfoma Hyperledger Fabric berdasarkan *throughput* dan *latency* dilakukan terhadap seluruh fungsi transaksi pada *smart contract*. Hasil pengujian kemudian dievaluasi untuk menilai perfoma yang didapatkan.

3.2 Perancangan Fitur

Terdapat berbagai solusi aplikasi autentikasi produk dengan *blockchain* yang telah ada, seperti Seal, Orygene, dan Uatag. Berdasarkan observasi yang dilakukan, fitur-fitur yang ingin diimplementasikan pada Oricon dapat dilihat pada Tabel 3.1 terkait fitur-fitur manajemen produk, Tabel 3.2 terkait fitur-fitur manajemen pengguna, dan Tabel 3.3 terkait fitur-fitur token ORC. Token ORC dirancang pada *smart contract* menggunakan standar token ERC-20, dengan model *account-based* di mana setiap pengguna memiliki saldo token masing-masing. Aplikasi ini juga memiliki akses kontrol terhadap fitur yang digunakan berdasarkan hak pengguna tersebut. Peran pengguna dibagi menjadi 3, yaitu *admin* (A) sebagai administrator perusahaan, *client* sebagai klien perusahaan (C), dan *token admin* (T) sebagai administrator Oricon.

Tabel 3.1 Fitur-fitur manajemen produk

Fitur	**Deskripsi Fitur**	**Hak Akses**		
		A	**C**	**T**
Create Product	Fitur ini berfungsi untuk menyimpan produk baru pada *blockchain*.	v	-	-
Read Product	Fitur ini berfungsi untuk membaca data produk yang disimpan pada *blockchain*.	v	-	-
Update Product	Fitur ini berfungsi untuk memperbaharui data produk yang disimpan pada *blockchain*.	v	-	-
Delete Product	Fitur ini berfungsi untuk menandai data produk yang dihapus pada *blockchain*.	v	-	-
Get Product History	Fitur ini berfungsi untuk mengambil *history* data produk berdasarkan perubahan yang telah dilakukan.	v	-	-
Search Product	Fitur ini berfungsi untuk mengautentikasi produk menggunakan kode QR yang dipindai pada aplikasi. Kode QR tersebut digunakan untuk melakukan pencarian data produk yang disimpan pada *blockchain*.	v	v	-

Tabel 3.2 Fitur-fitur manajemen pengguna

Fitur	Deskripsi Fitur	Hak Akses		
		A	C	T
Register User	Fitur ini berfungsi untuk mendaftarkan pengguna baru.	v	-	-
Enroll User	Fitur ini berfungsi untuk mengimpor identitas (*wallet*) pengguna yang telah terdaftar pada sistem.	v	v	-
Login	Fitur ini berfungsi untuk mengautentikasi pengguna yang ingin menggunakan aplikasi.	v	v	-
Get Users	Fitur ini berfungsi untuk mengambil seluruh pengguna yang terdaftar pada sistem.	v	-	-

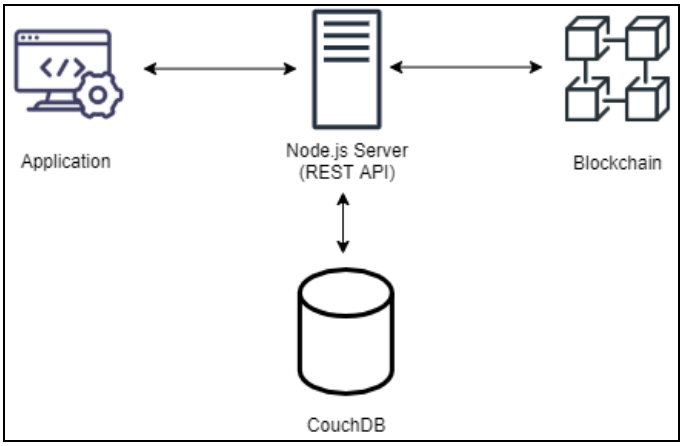
Tabel 3.3 Fitur-fitur token ORC

Fitur	Deskripsi Fitur	Hak Akses		
		A	C	T
Get Token Details	Fitur ini berfungsi untuk mengambil detail token dan saldo dari akun tersebut.	v	v	v
Mint Token	Fitur ini berfungsi untuk membaca data produk yang disimpan pada *blockchain*.	-	-	v
Transfer Token	Fitur ini berfungsi untuk memindahkan sejumlah token dari akun pemilik ke akun tujuan.	v	v	v
Approve Token	Fitur ini berfungsi untuk mengizinkan sejumlah token dari akun pemilik digunakan oleh akun pemakai.	v	v	v
Get Allowance Token	Fitur ini berfungsi untuk mengambil jumlah token yang diizinkan oleh akun pemilik kepada akun pemakai.	v	v	v
Tranfer From Token	Fitur ini berfungsi untuk memindahkan sejumlah token yang diizinkan oleh akun pemilik untuk akun pemakai kepada akun tujuan.	v	v	v

3.3 Perancangan Sistem

Oricon berinteraksi dengan Node.js *server* yang berfungsi sebagai REST API dan berkomunikasi dengan Hyperledger Fabric, yang dapat dilihat pada Gambar 3.1. REST API berjalan menggunakan protokol HTTP untuk menerima *request* dan mengembalikan *response*. REST API diimplementasikan menggunakan *framework* Express.js, dengan format *request* dan *response* yang

digunakan berbentuk JSON. Node.js *server* berkomunikasi dengan Hyperledger Fabric menggunakan SDK yang disediakan untuk Node.js. CouchDB digunakan sebagai *database* pada Node.js *server* untuk menyimpan *wallet* (berisi *public key*, *private key*, dan *digital certificate* pengguna) dan kredensial akun (berisi *username* dan *password* pengguna).



Gambar 3.1 Interaksi aplikasi dengan sistem

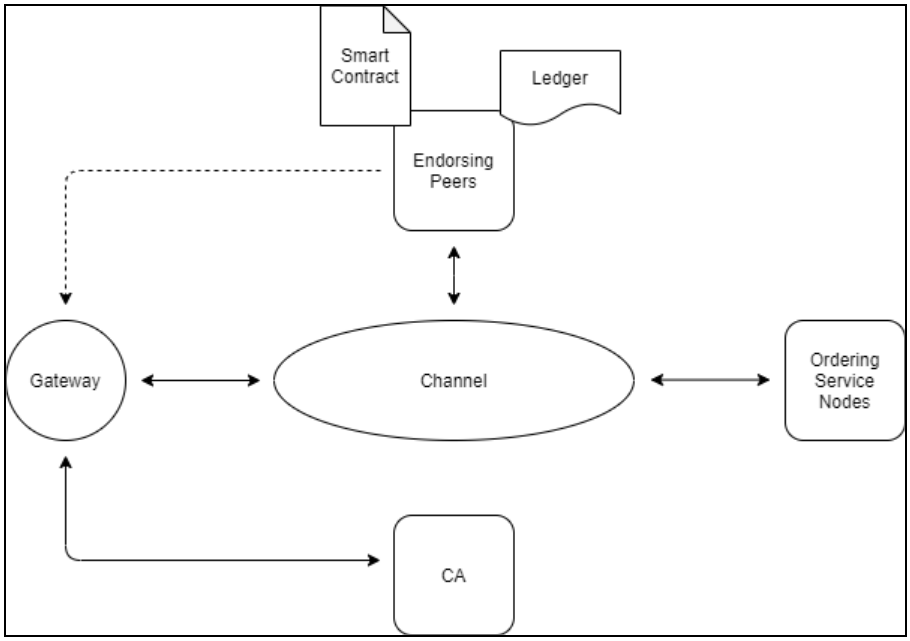
Jaringan *blockchain* Hyperledger Fabric yang dirancang terdiri dari satu *channel* yang menghubungkan aplikasi (Node.js *server*), *endorsing peers*, dan *ordererering service nodes*, yang dapat dilihat Gambar 3.2. *Blockchain* ini bersifat *private*, di mana hanya ditujukan untuk partisipan yang terotorisasi dari sebuah perusahaan tertentu. Identitas dari setiap partisipan didaftarkan pada *certificate authorities* (CA) oleh administrator perusahaan. Agar dapat terhubung ke *channel*, aplikasi (Node.js server) menggunakan identitas tersebut pada Fabric Gateway yang berjalan dengan protokol gRPC untuk berkomunikasi dengan jaringan.

Setiap *endorsing peer* memiliki salinan *smart contract* dan *ledger* masing-masing, yang dirancang untuk menjalankan *endorsement* dan *validation*. World

state yang digunakan adalah CouchDB, yang memungkinkan *smart contract* untuk melakukan *query* yang kompleks. Aplikasi (Node.js *server*) akan terhubung dengan sebuah *endorsing peer* dan menjalankan *smart contract* untuk membuat *transaction proposal*. Jika transaksi berupa *query*, maka *transaction response* (hasil eksekusi dari *smart contract*) akan dikembalikan pada aplikasi. Jika transaksi berupa pembaharuan *state* pada *ledger*, maka aplikasi akan mengumpulkan *transaction response* yang dibutuhkan dari *endorsing peer* tertentu menjadi *transaction message* untuk dikirimkan ke *ordering service nodes*.

Seluruh transaksi yang diterima *ordering service nodes* akan diurutkan melalui konsensus dan dibungkus menjadi blok untuk dikirimkan ke sebuah *leader peer*. *Leader peer* kemudian akan menyebarkan blok tersebut ke *peer* lainnya menggunakan protokol *gossip peer-to peer*. Seluruh transaksi pada blok akan divalidasi satu per satu dan di-*commit* pada *ledger* setiap *peer*. *Peer* akan mengirimkan *event* kepada aplikasi untuk memberitahu transaksi yang berhasil di-*commit*.

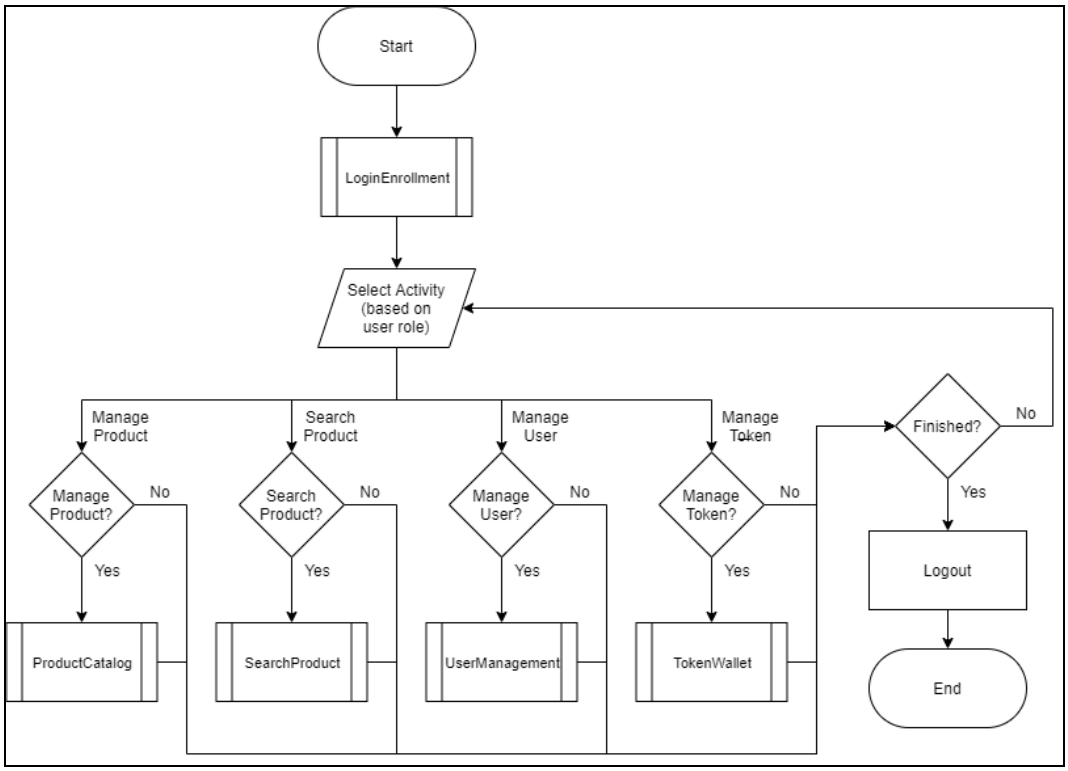
Terdapat beberapa algoritma yang digunakan untuk mengimplementasikan Hyperledger Fabric. Algoritma konsensus yang digunakan adalah Raft, yang bersifat *crash fault tolerant*. Pengurutan transaksi dilakukan oleh *leader orderers*, di mana *orderers* lainnya yang berperan sebagai *followers* akan mengikuti *state* yang dimiliki leader. Algoritma *digital signature* yang digunakan adalah ECDSA dengan SHA-256 untuk memberikan setiap *signature* pada setiap transaksi. Algoritma *hashing* yang digunakan untuk melakukan *block hashing* adalah SHA-256.



Gambar 3.2 Interaksi *server* dengan Hyperledger Fabric

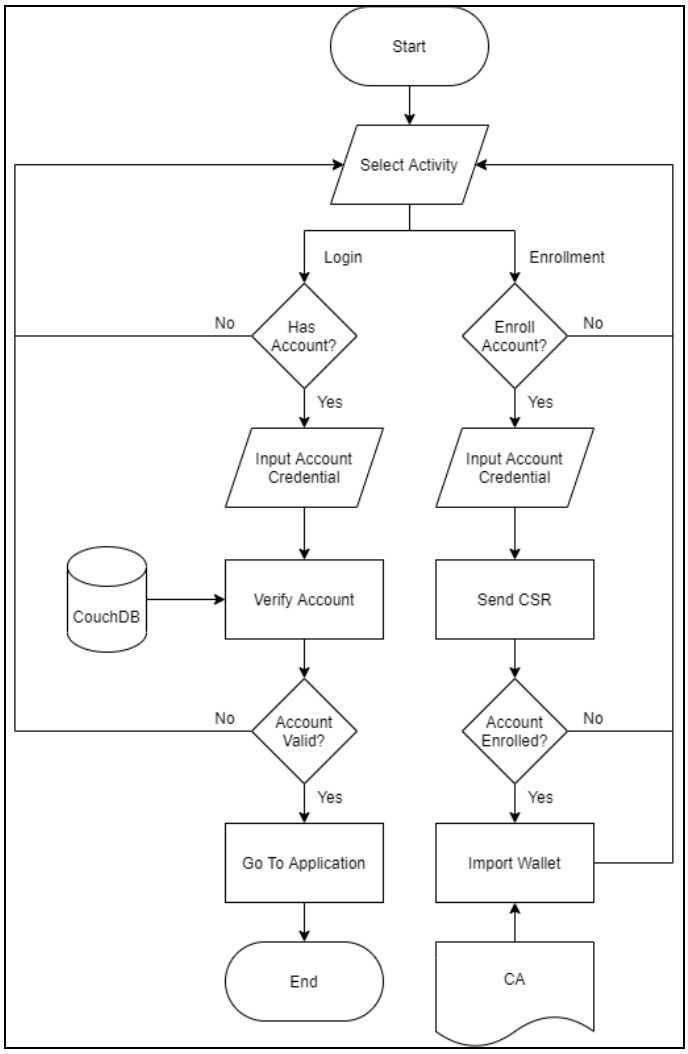
3.4 Perancangan Flowchart Aplikasi

Flowchart aplikasi dirancang dari sisi pengguna. Flowchart utama dapat dilihat pada Gambar 3.3. Terdapat beberapa modularisasi dari flowchart ini, yaitu LoginEnrollment, ProductCatalog, SearchProduct, UserManagement, dan TokenWallet. Setelah proses melewati LoginEnrollment, halaman yang dapat diakses pengguna akan disesuaikan dengan perannya masing-masing. Pengguna yang berperan sebagai *admin* dapat mengakses halaman *product catalog*, *search product*, *user management* dan *token wallet*, sedangkan pengguna yang berperan sebagai *client* hanya dapat mengakses halaman *search product* dan *token wallet*. Jika sudah selesai, pengguna dapat keluar dari akunnya dan kembali ke halaman awal aplikasi.



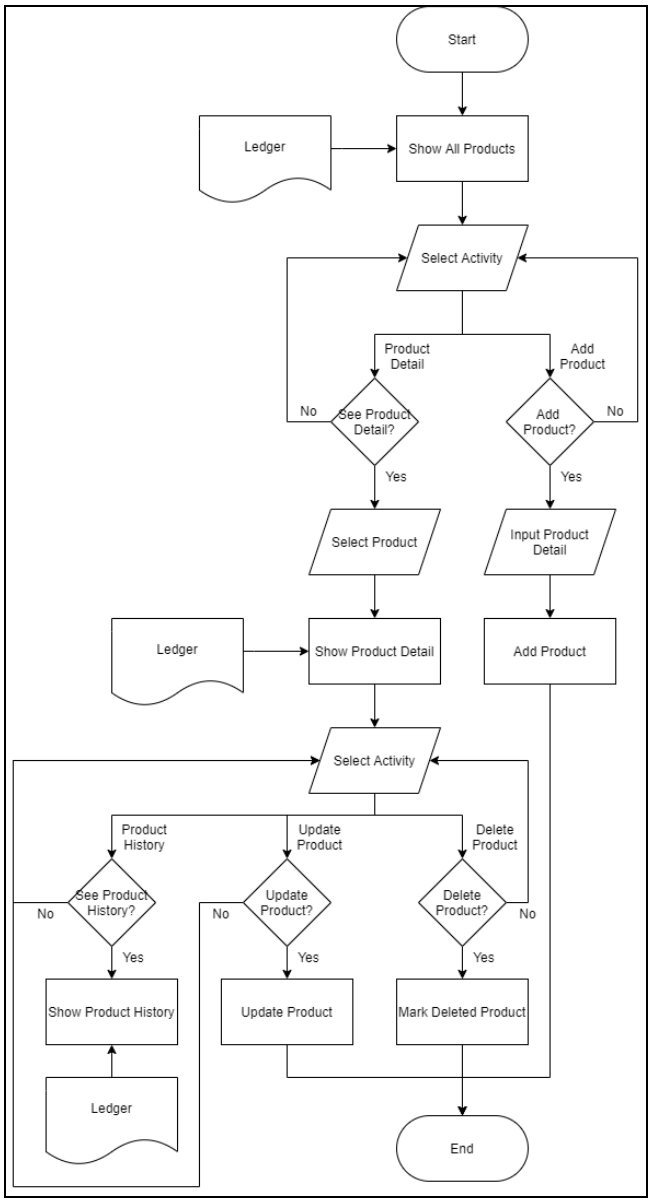
Gambar 3.3 *Flowchart* utama aplikasi

Flowchart login dan *enrollment* dapat dilihat pada Gambar 3.4. Halaman awal aplikasi dimulai dengan *login.* Pengguna yang telah memiliki akun terdaftar dapat langsung *login* ke aplikasi, sedangkan pengguna yang belum memiliki akun terdaftar dapat menuju ke halaman *enrollment*. Kredensial akun yang dimasukkan saat *login* akan diverifikasi dengan kredensial yang disimpan pada CouchDB, yang telah melalui proses *enrollment* ke CA. Aplikasi akan mengirimkan *certificate signing request* (CSR) ke CA untuk mendapatkan *wallet* pengguna dan mengimpornya pada CouchDB.



Gambar 3.4 *Flowchart login* dan *enrollment*

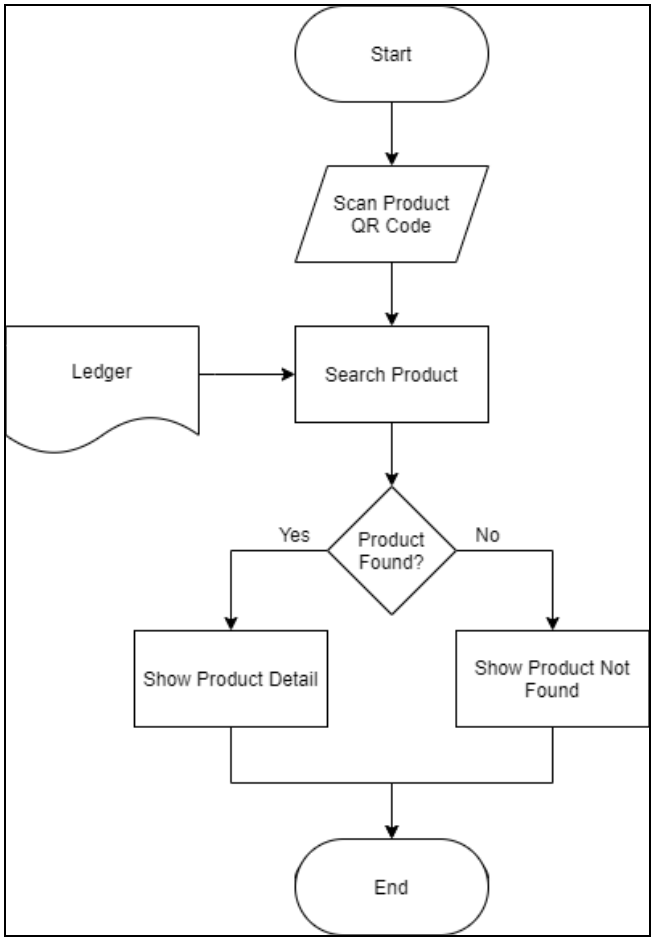
Flowchart product catalog dapat dilihat pada Gambar 3.5. Pada halaman ini, pengguna dapat mengelola produk perusahaannya yang disimpan pada *blockchain*. Fitur-fitur yang dapat dilakukan aplikasi terkait produk adalah menampilkan seluruh produk, membaca detail produk, menambahkan produk, memperbaharui produk, melihat *history* produk, dan menghapus produk. Seluruh proses *query* ke *ledger* dilakukan pada *world state* untuk mengambil *current state* dari objek produk yang disimpan.



Gambar 3.5 *Flowchart product catalog*

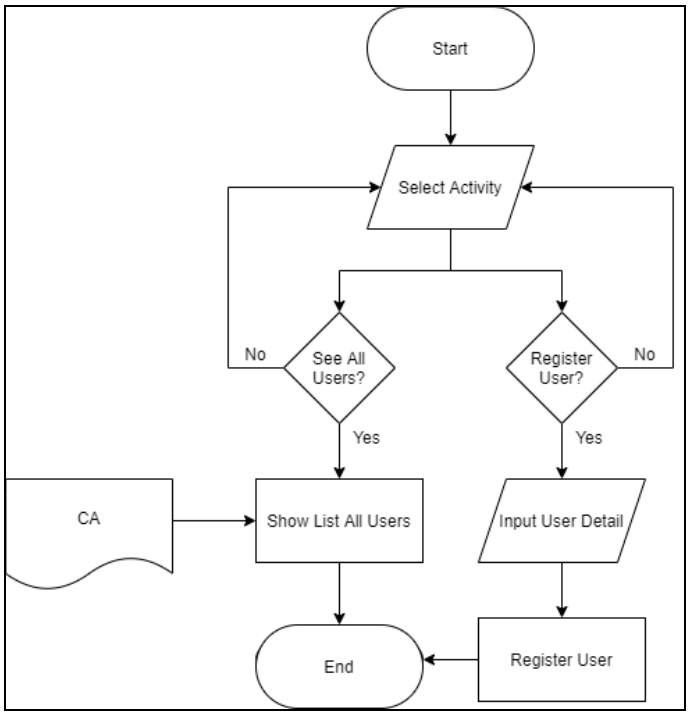
Flowchart search product dapat dilihat pada Gambar 3.6. Pada halaman ini, pengguna dapat memindai kode QR produk untuk mencari produk yang diinginkan. Kode QR produk merepresentasikan ID transaksi yang dikirimkan oleh *smart contract* ketika membuat *transaction proposal*, sehingga setiap produk memiliki kode QR yang unik. Proses *query* ke *ledger* dilakukan pada *blockchain* berdasarkan ID transaksi untuk mengambil *state objek* yang diinginkan pada saat

tertentu. Jika produk ditemukan, maka aplikasi akan menampilkan detail produk. Jika produk tidak ditemukan, maka aplikasi akan menampilkan pesan bahwa produk tidak ditemukan.



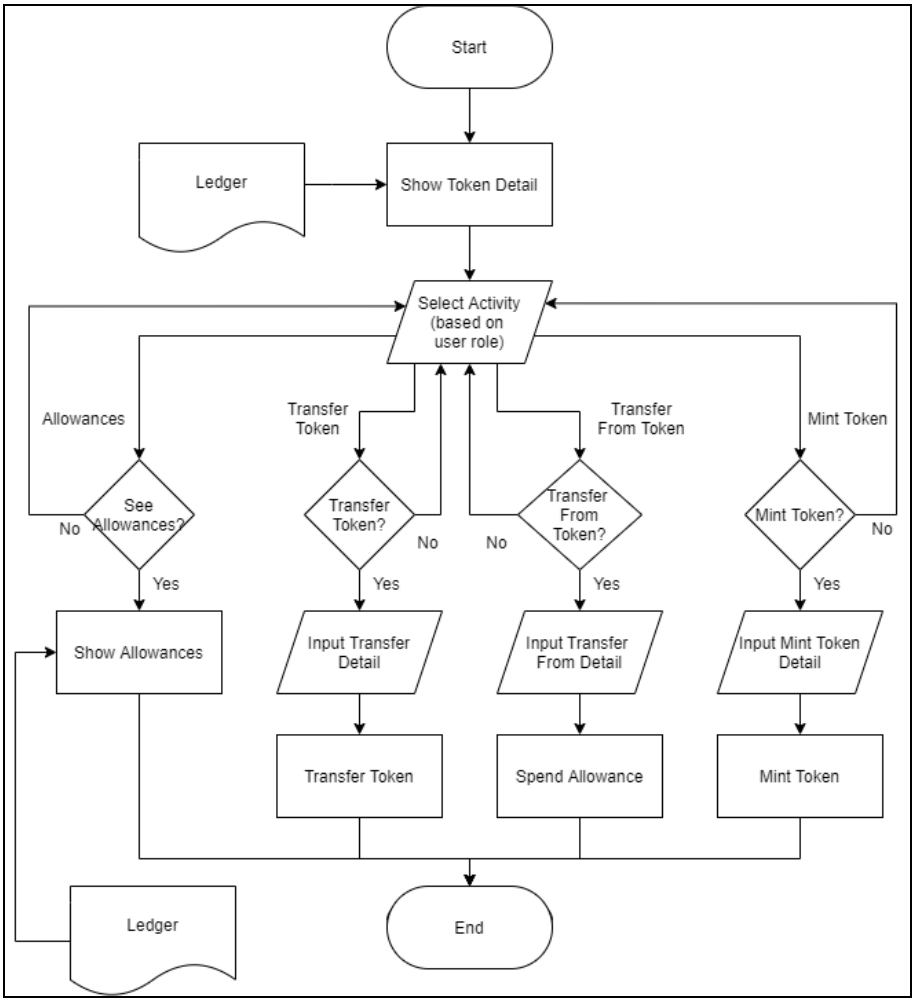
Gambar 3.6 *Flowchart search product*

Flowchart user management dapat dilihat pada Gambar 3.7. Pada halaman ini, pengguna dapat melihat daftar seluruh pengguna dan mendaftarkan pengguna baru. Daftar seluruh pengguna yang dimaksud adalah pengguna yang telah didaftarkan oleh administrator perusahaan ke CA. Administrator perusahaan akan memasukkan kredensial pengguna tersebut beserta peran yang diberikan kepadanya.



Gambar 3.7 *Flowchart user management*

Flowchart user management dapat dilihat pada Gambar 3.8. Pada halaman ini, pengguna dapat mengelola token ORC yang dimilikinya. Fitur-fitur yang dapat dilakukan aplikasi terkait token adalah menampilkan detail token (berupa saldo akun, dan total persediaan token untuk pengguna *token admin*), melihat seluruh *allowances* yang diberikan kepada pengguna lain, *transfer token*, *transfer from token*, dan *mint token* (jika pengguna adalah *token admin*). Token ORC bersifat *fungible token*, di mana setiap token yang ditukarkan bernilai sama. Standar token ERC-20 digunakan untuk mengimplementasikan token ORC pada *smart contract*. Seluruh proses *query* ke *ledger* dilakukan pada *world state* untuk mengambil *current state* dari objek token yang disimpan.



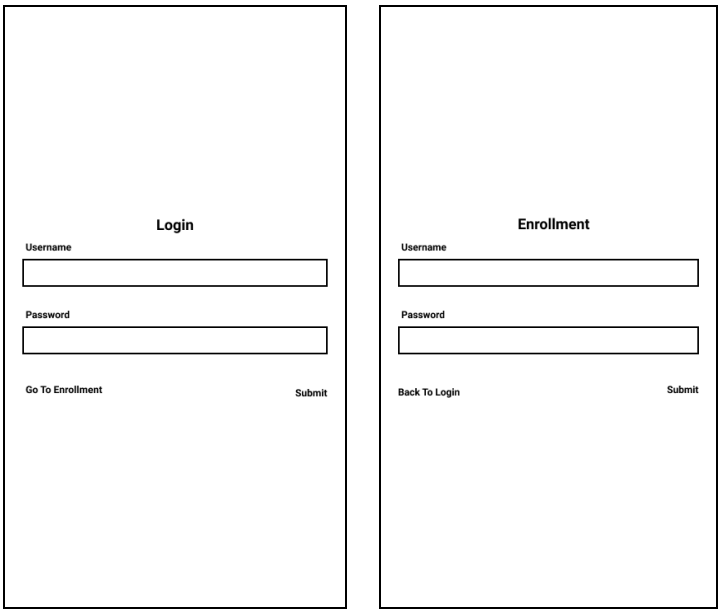
Gambar 3.8 *Flowchart token wallet*

3.5 Perancangan User Interface

User interface yang dirancang ditujukan untuk aplikasi *mobile cross-platform*. *Framework* yang digunakan untuk mengimplementasikan *user interface* adalah Ionic dengan Angular. Perancangan *user interface* dilakukan untuk keseluruhan halaman, di mana halaman atau fitur yang tidak dapat diakses oleh pengguna tertentu tidak akan ditampilkan. Berikut adalah perancangan *user interface* yang dibuat.

1. Halaman Login dan Enrollment

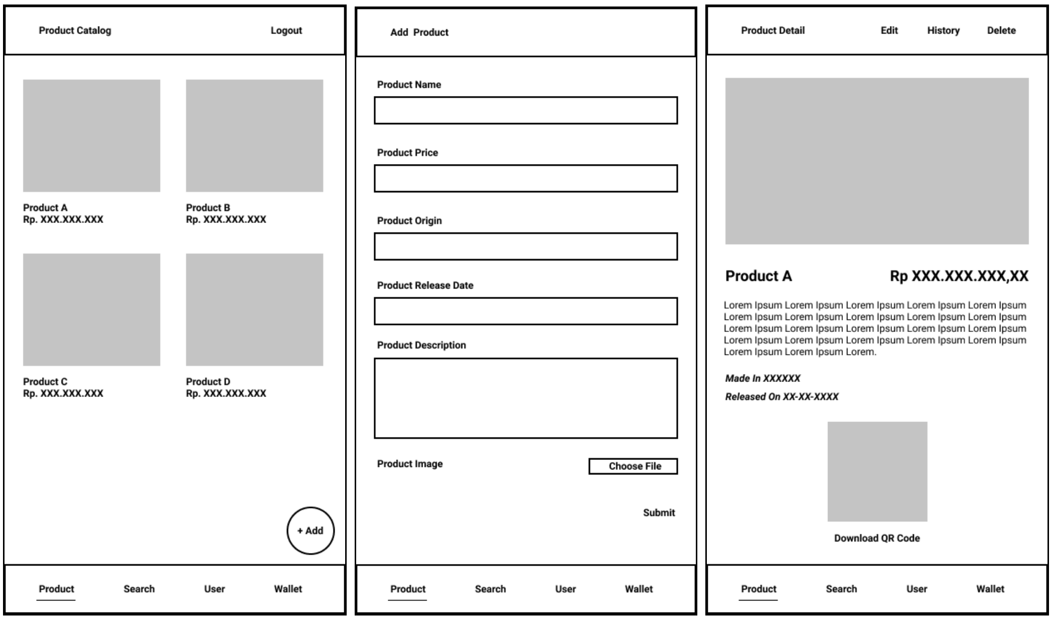
Pada halaman ini, pengguna dapat melakukan *login* untuk masuk ke aplikasi atau *enrollment* untuk mendaftarkan akun. Rancangan halaman Login dan Enrollment dapat dilihat pada Gambar 3.9.



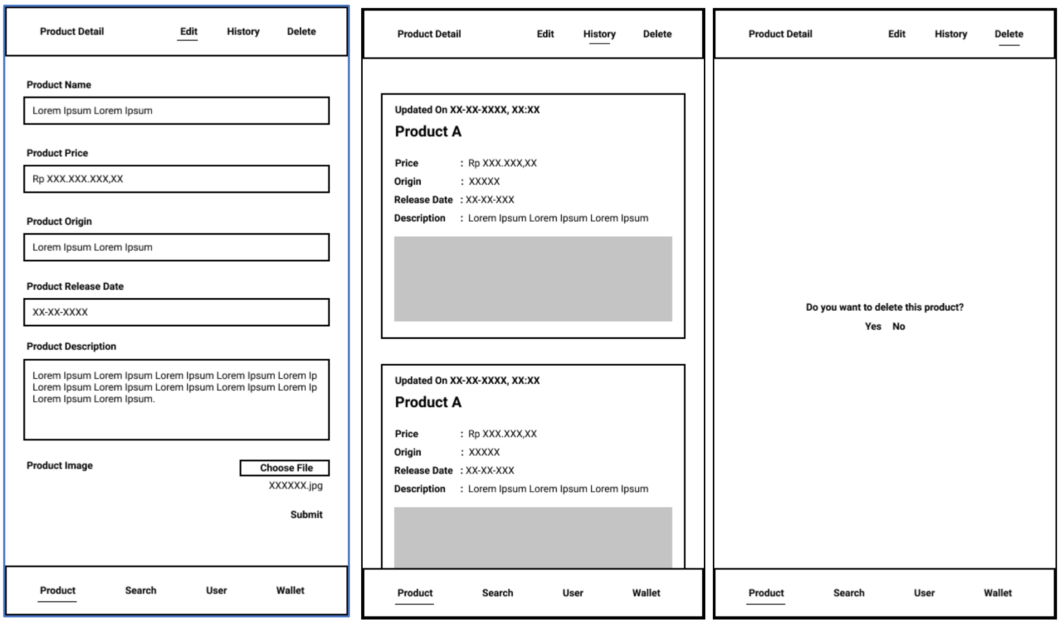
Gambar 3.9 Rancangan Login dan Enrollment

2. Halaman Product Catalog

Pada halaman ini, pengguna dapat melihat daftar produk, melihat detail produk, dan menambahkan produk. Pada halaman detail produk, pengguna dapat memperbaharui produk, melihat riwayat produk, dan menghapus produk. Rancangan halaman Product Catalog, Add Product dan Detail Product dapat dilihat pada Gambar 3.10. Rancangan halaman Edit Product, History Product, dan Delete Product dapat dilihat pada Gambar 3.11.



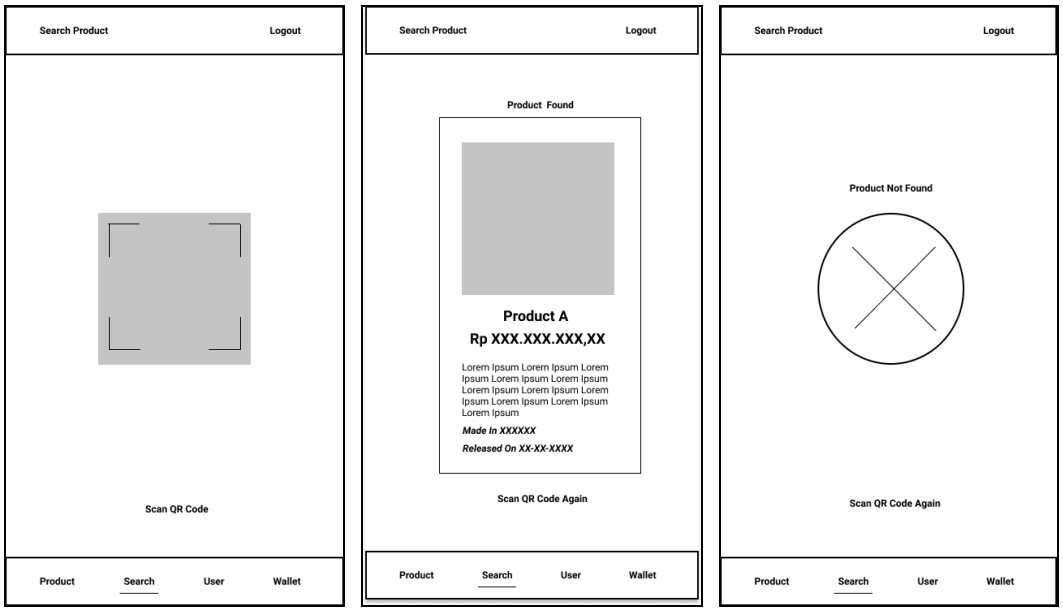
Gambar 3.10 Rancangan Product Catalog, Add Product, Detail Product



Gambar 3.11 Rancangan Edit Product, History Product, Delete Product

3. Halaman Search Product

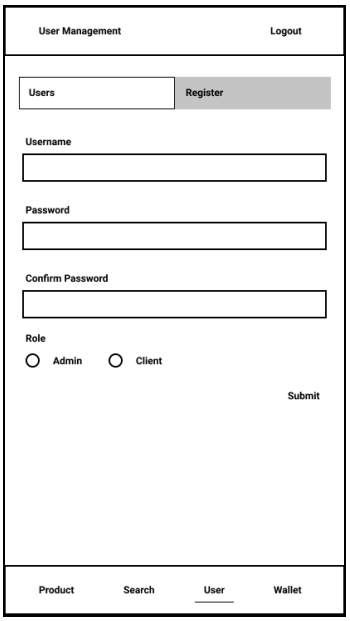
Pada halaman ini, pengguna dapat mencari produk dengan memindai kode QR. Rancangan halaman Search Product dapat dilihat pada Gambar 3.12.



Gambar 3.12 Rancangan Search Product

4. Halaman User Management

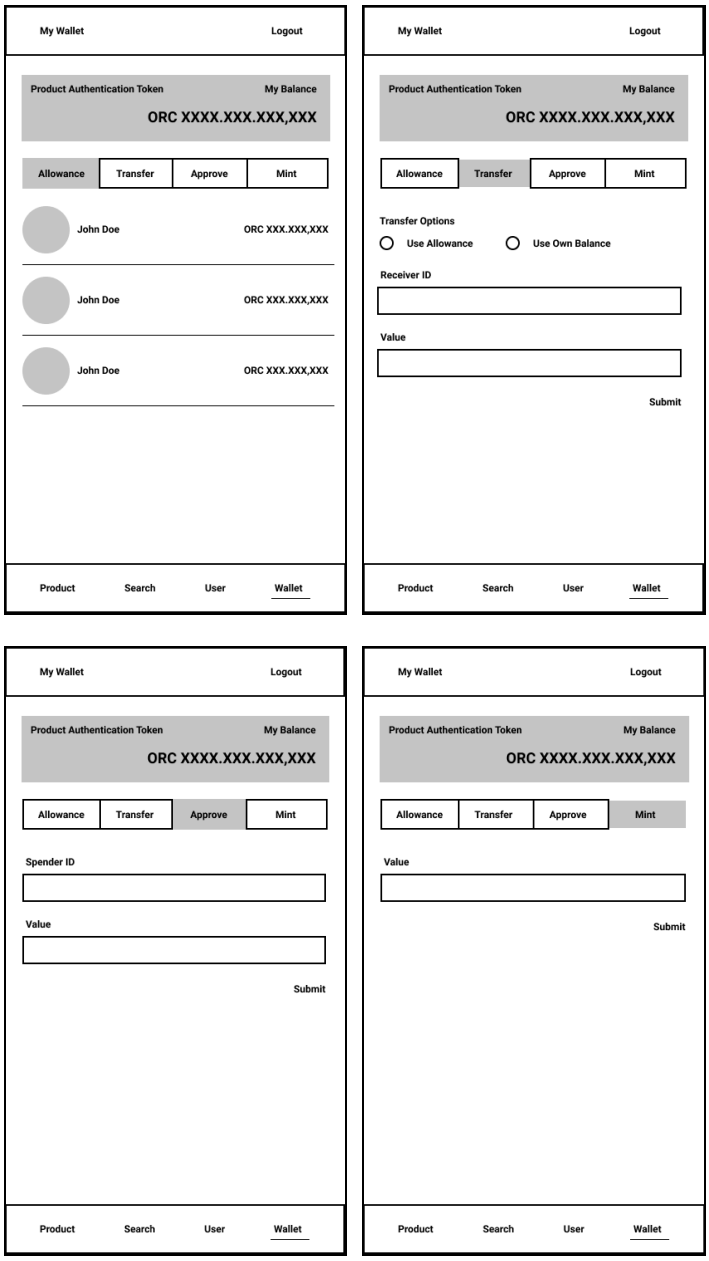
Pada halaman ini, pengguna dapat melihat daftar seluruh pengguna dan meregistrasi pengguna baru. Rancangan halaman User Management dapat dilihat pada Gambar 3.13.



Gambar 3.13 Rancangan User Management

5. Halaman Token Wallet

Pada halaman ini, pengguna dapat melihat *allowances* yang diberikan, melakukan transfer, memberi *allowances*, dan menambah jumlah token. Rancangan halaman Token Wallet dapat dilihat pada Gambar 3.14.



Gambar 3.14 Rancangan Token Wallet