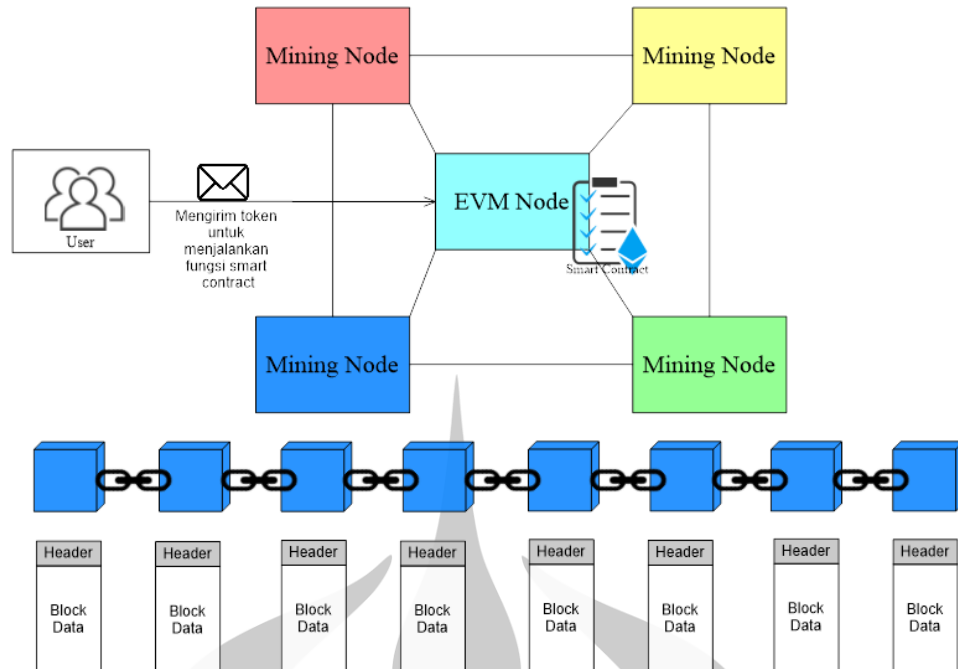


BAB 3

METODOLOGI PENELITIAN

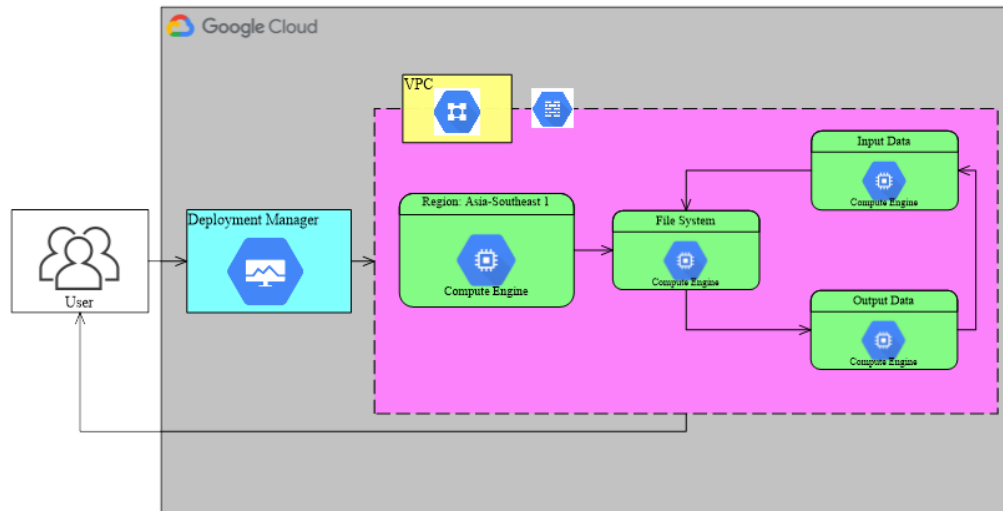
3.1 Arsitektur Ethereum



Gambar 3.1 Arsitektur Ethereum

Konsep dasar dari arsitektur Ethereum adalah bagaimana user menjalankan fungsi *smart contract* yang telah dibuat untuk mendukung keperluan bisnis di dalam blockchain yang telah ditulis di dalam *smart contract*. Arsitektur ini memberikan ilustrasi bagaimana teknologi-teknologi yang saling berkaitan satu dengan yang lain dalam menjaga ekosistem Ethereum tetap berjalan untuk menciptakan blok-blok di dalam blockchain yang berisi data transaksi dari pengguna. Mining node merupakan penambang yang bekerja memvalidasi transaksi yang berjalan di dalam jaringan blockchain Ethereum, setiap blok berisi tentang data yang sudah dijelaskan pada **Gambar 2.6** dan **Gambar 2.8**

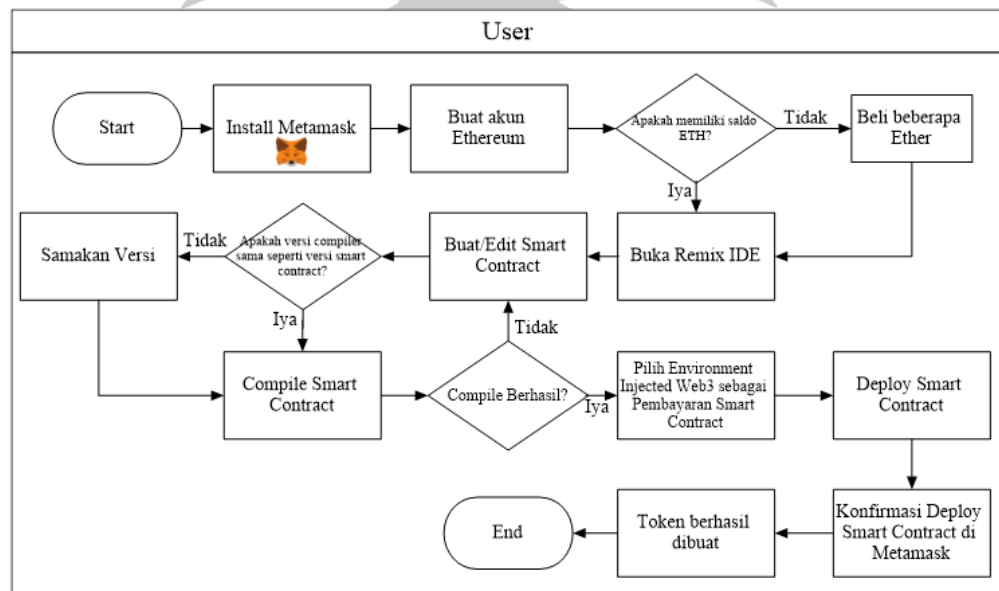
3.2 Arsitektur Cloud



Gambar 3.2 Arsitektur Cloud

Pada diagram di atas terlihat bahwa peneliti menggunakan layanan cloud GCP untuk men-deploy CMS. Peneliti menggunakan fitur Deployment Manager yang sudah terintegrasi dengan Compute Engine, VPC, dan Firewall agar aplikasi CMS peneliti bersifat online.

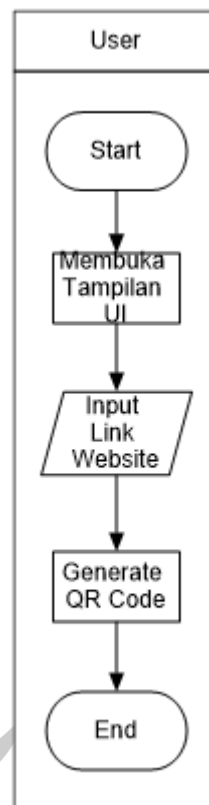
3.3 Membuat Smart Contract



Gambar 3.3 Alur Membuat Smart Contract

Pada diagram diatas pengguna wajib *install* aplikasi dompet kripto (seperti metamask) lalu membuat alamat Ethereum yang akan digunakan oleh pengguna. Saat membuat *smart contract* terdapat biaya untuk men-deploy di jaringan Ethereum jadi pastikan pengguna memiliki beberapa Ether (ETH) terlebih dahulu dengan cara mengikuti airdrop atau membeli di exchanger. Setelah memiliki beberapa Ether (ETH) pengguna bisa membuka IDE khusus milik Ethereum yaitu Remix Ethereum IDE pada tautan berikut (remix.ethereum.org) lalu pengguna bisa membuat dan memodifikasi *smart contract* yang akan digunakan sesuai keperluan bisnis pengguna. Contoh *smart contract* dari peneliti bisa diakses melalui tautan berikut (https://github.com/hanggaa/Thesis/blob/main/Token/remixbackup/.workspaces/default_workspace/ballot.sol) *smart contract* yang ditulis oleh peneliti berfungsi dalam membuat token baru yang berjalan pada jaringan Ethereum agar antar entitas bisa mengirim token sebagai syarat pencatatan data dalam blockchain. Setelah membuat atau memodifikasi *smart contract* langkah selanjutnya adalah mencocokkan versi solidity dengan compiler apakah sama atau tidak, jika tidak sama maka proses meng-*compile* akan mengalami kegagalan. Langkah selanjutnya adalah meng-*compile smart contract* jika proses *compile* berhasil maka *smart contract* bisa di-*deploy* ke dalam EVM node atau jaringan Ethereum. Proses *deploy* dengan cara memilih Injected Web 3 (MetaMask) sebagai sarana pembayaran *deploy*, lalu konfirmasi pembayaran melalui *Pop-up* MetaMask yang muncul setelah itu konfirmasi *deploy smart contract*. Selamat token dari *smart contract* telah berhasil dibuat dan bisa dijalankan guna pencatatan di dalam blockchain.

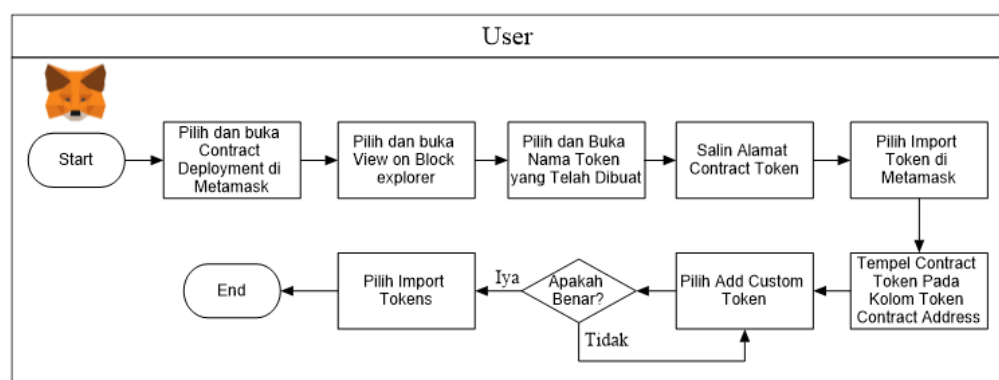
3.4 QR Code



Gambar 3.4 Alur Kerja Mencetak QR Code

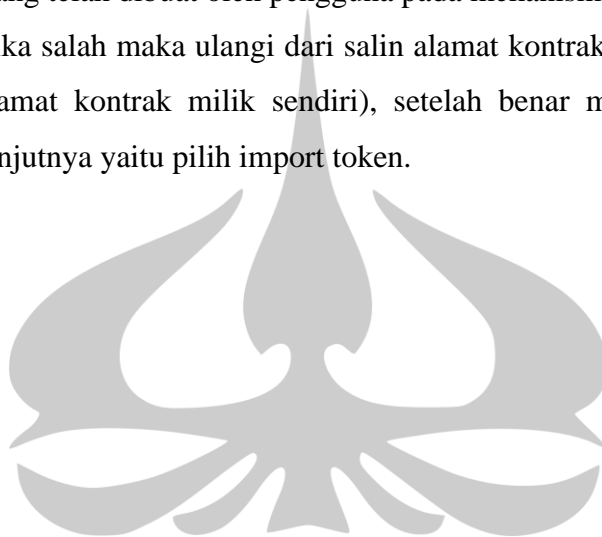
Pada **Gambar 3.4** user menyalin tautan website yang akan disimpan di dalam QR Code lalu tautan tersebut ditempelkan di sistem generate QR Code untuk diubah menjadi QR Code sebelum dicetak dan dipasang menjadi label.

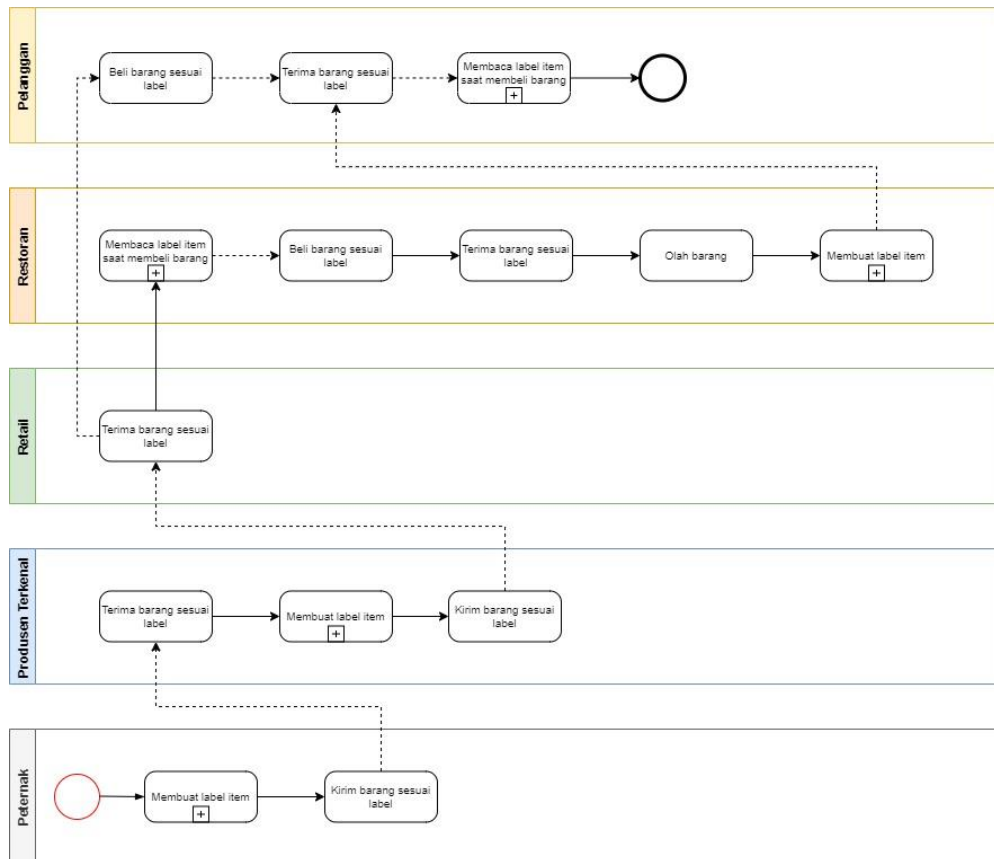
3.5 Memasang Token di MetaMask



Gambar 3.5 Pasang Token *Smart Contract* di MetaMask

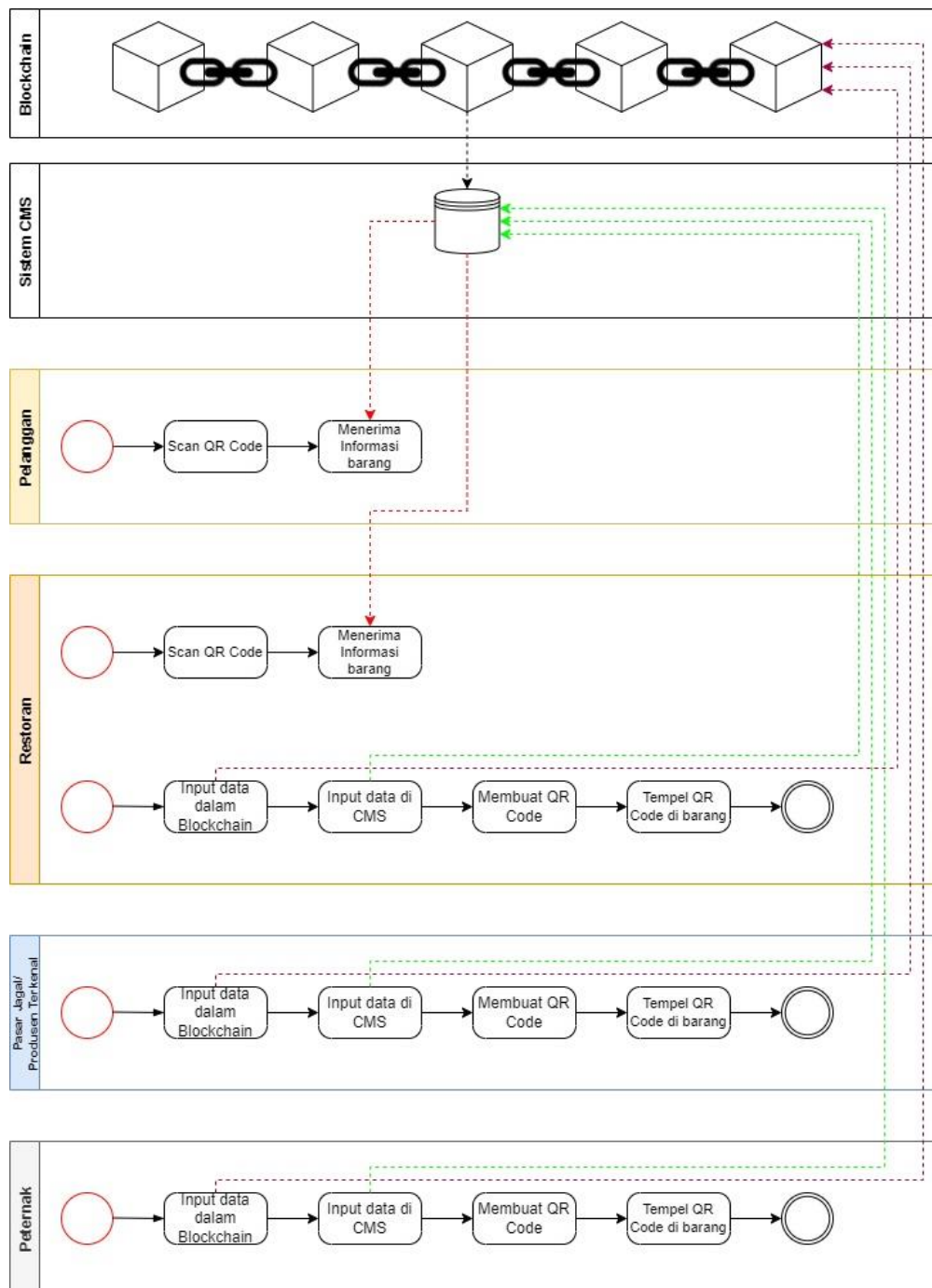
Pengguna bisa memasang token yang baru saja dibuat melalui mekanisme sebelumnya ke akun Ethereum di MetaMask dengan cara memilih Contract Deployment untuk melihat detail dari transaksi *smart contract* yang baru saja di-deploy, pilih *View on block explorer* untuk melihat data transaksi dalam blockchain. Setelah membuka *View on block explorer* maka terlihat detail transaksi pembuatan token yang telah dibuat oleh pengguna, selanjutnya salin alamat kontrak token lalu pilih *Import Token* di MetaMask untuk mengimpor data token tersebut ke dalam akun Ethereum MetaMask pengguna dan tempelkan alamat kontrak yang telah disalin tadi. Jika benar maka akan muncul data token yang telah dibuat oleh pengguna pada mekanisme sebelumnya, dan sebaliknya jika salah maka ulangi dari salin alamat kontrak yang telah dibuat (Pastikan alamat kontrak milik sendiri), setelah benar muncul data token langkah selanjutnya yaitu pilih import token.





Gambar 3.7 Proses Bisnis Supply Chain Menggunakan Blockchain (Produsen Terkenal/PT) Level 1

Terlihat bahwa pada **Gambar 3.6** dan **Gambar 3.7** sebelum barang dikirim peternak membuat label item barang tersebut agar tercatat di dalam blockchain, setelah itu entitas selanjutnya produsen terkenal atau pasar jagal menerima barang sekaligus label item dari peternak untuk diproses lagi dan membuat label item yang baru sebelum barang dikirimkan ke retail. Setelah barang di retail maka entitas seperti pelanggan (termasuk restoran) bisa membaca label item barang dengan cara memindai QR Code sebelum membeli barang tersebut untuk melihat entitas yang bekerja pada proses supply chain sebelumnya.



Gambar 3.8 Proses Bisnis Supply Chain Menggunakan Blockchain (Pasar Jagal/Produsen Terkenal) Level 2

Gambar 3.8 merupakan rincian dari tugas membuat label item dan membaca label item saat membeli barang, dalam proses membuat label item

para entitas meng-*input* data ke dalam Blockchain dengan cara mengirim token yang telah dibuat pada mekanisme sebelumnya ke alamat yang digunakan oleh entitas agar data transaksi ter-*input* ke dalam blockchain. Setelah data sukses ter-*input* ke dalam blockchain langkah selanjutnya adalah para entitas meng-*input* data ke dalam CMS sesuai dengan bisnisnya masing-masing. Tautan situs web hasil dari *input* data ke dalam CMS diubah ke dalam QR Code oleh para entitas lalu ditempelkan ke barang agar konsumen bisa melihat situs web yang berisi Txn (Wajib) dan lainnya.

Pada proses membaca label item konsumen cukup memindai QR Code yang menempel pada barang untuk melihat transaksi proses supply chain pada blockchain. Tentunya konsumen bisa melihat asal mula dari barang yang akan dibeli (Tergantung kesepakatan antar pihak).

