Threat Intelligence Blockchain

IF4035 - Blockchain

Oleh

Christophorus Dharma Winata / 13521009
Muhammad Haidar Akita Tresnadi /13521028
Moch. Sofyan Firdaus / 13521083

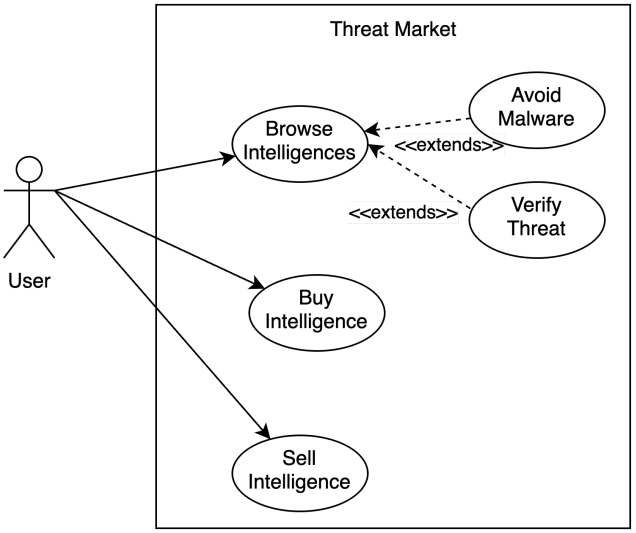
PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
2024

I. Problem Statement dan Use Case

A. Problem Statement

Blockchain-based Cyber Threat Marketplace adalah platform terdesentralisasi tempat organisasi dan peneliti bertukar informasi ancaman siber dengan scope aplikasi ini adalah sample malware, secara aman. Dengan memanfaatkan blockchain untuk kekekalan data dan transparansi, pasar ini menumbuhkan kepercayaan dan kolaborasi sekaligus memastikan integritas dan keandalan informasi bersama.

B. Use Case



Gambar 1 Use Case Diagram Threat Intelligence Marketplace

Pada aplikasi Threat Market, aktor berpusat pada user market yang saling bertukar informasi ancaman siber. User dapat membeli dan menjual informasi di dalam market untuk digunakan dalam kepentingan keamanan. Selain itu, User dapat melihat informasi-informasi yang tersedia pada market. Informasi ini dilengkapi dengan penanda kalau informasi yang berkaitan berisi malware dan/atau sudah terverifikasi.

II. Platform Blockchain

Platform blockchain yang digunakan adalah Private Ethereum dengan Geth yang berbasis Golang. Adapun bahasa pemrograman *smart contracts* yang digunakan adalah Solidity.

Pemilihan platform Private Ethereum dilakukan dengan justifikasi dukungan platform yang luas dari segi dokumentasi, komunitas, serta infrastruktur yang sudah diterapkan pada Ethereum seperti kakas Hardhat. Bahasa pemrograman Solidity dipilih karena bahasa ini merupakan bahasa pemrograman Blockchain yang lazim

digunakan sehingga memiliki dukungan komunitas serta dokumentasi yang besar.

Berikut adalah teknologi yang digunakan dalam aplikasi ini:

1. **Platform Blockchain - Geth**
   Geth memungkinkan implementasi dan pengelolaan blockchain Ethereum yang efisien, stabil, dan skalabel untuk aplikasi berbasis blockchain.

2. **Network Deployment - Kurtosis**
   Kurtosis mempermudah penyebaran dan pengujian jaringan blockchain, memastikan konfigurasi yang minim kesalahan dalam lingkungan terisolasi.

3. **Smart Contract - Solidity**
   Solidity dipilih sebagai bahasa pemrograman smart contract karena solidity sudah cukup banyak digunakan untuk membuat banyak dApps lain yang sudah banyak digunakan di seluruh dunia. Oleh karena itu, solidity merupakan pilihan bahasa yang bagus karena sudah banyak dokumentasi dan contoh-contoh di internet, sehingga dapat memudahkan proses development.

4. **Penyedia Data Oracle -**
   Penyedia data oracle yang digunakan adalah VirusTotal. Dengan VirusTotal, kami dapat menentukan apakah sebuah file merupakan malware atau bukan melalui hashnya.

5. **Backend IPFS -**
   Express digunakan sebagai backend karena mendukung middleware untuk menangani permintaan HTTP, mempermudah integrasi dengan layanan eksternal seperti IPFS dan blockchain.

6. **IPFS**
   Web3.Storage menggunakan IPFS untuk penyimpanan terdesentralisasi yang aman dan efisien, mendukung arsitektur aplikasi yang terdesentralisasi

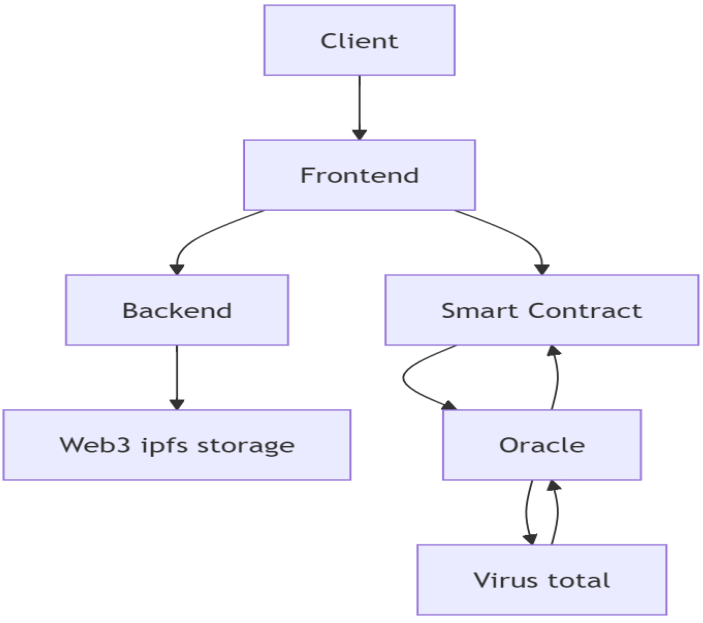
7. **Frontend - React.js**
   Framework pengembangan aplikasi dilakukan menggunakan React sesuai familiaritas pengembang dan dukungan pengembangan dApps pada framework React.

8. **Penyedia Wallet - Metamask**

Metamask digunakan sebagai penyedia sarana antarmuka antara pengguna dengan blockchain.

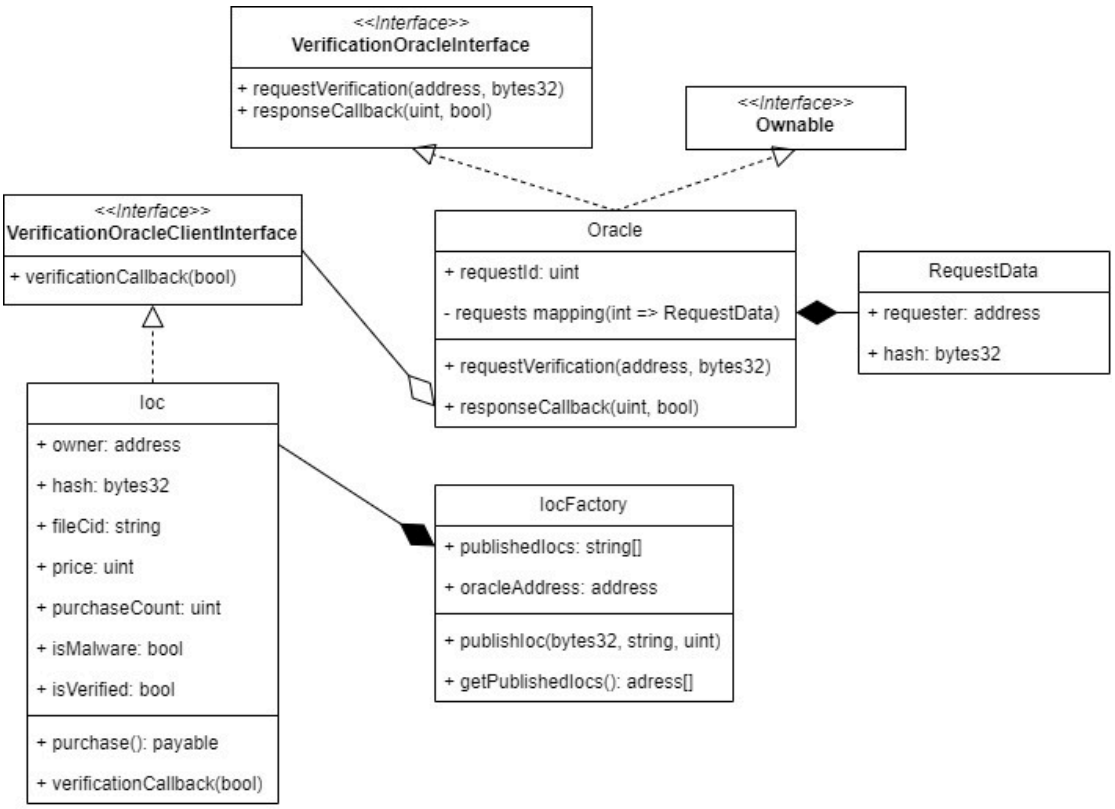
Aplikasi ini merupakan dApp (decentralized application) yang memungkinkan pengguna untuk membeli dan menjual sampel malware secara aman dan terdesentralisasi. Pada fitur jual, pengguna mengunggah file sampel malware melalui antarmuka frontend dApp. File tersebut akan dikirim ke backend, di mana backend mengenkripsi file tersebut dan menyimpannya di penyimpanan terdesentralisasi menggunakan IPFS Web3.Storage. Backend kemudian memberikan CID (Content Identifier) terenkripsi ke frontend, yang kemudian menyimpan CID tersebut ke smart contract di blockchain bersama dengan hash dari file untuk memastikan integritas data. Saat smart contract dibuat, oracle akan memverifikasi apakah sampel yang diunggah adalah malware yang valid atau bukan dengan menggunakan API dari virus total.

Pada fitur beli, pengguna dapat membeli sampel malware yang tersedia di marketplace. Setelah pembelian berhasil, pengguna akan menerima CID terenkripsi dari Smart Contract. CID ini kemudian dikirim ke backend melalui permintaan download. Backend akan mendekripsi file yang terkait dengan CID tersebut dan mengembalikannya ke pengguna. Dengan arsitektur ini, aplikasi memastikan keamanan data, privasi pengguna, dan keandalan sistem menggunakan blockchain dan teknologi desentralisasi.



III. Smart Contract dan Oracle

Berikut class diagram dari smart contract yang dibuat.



Oracle yang dibuat pada aplikasi ini bertugas untuk memverifikasi apakah file yang diupload oleh seller betul-betul merupakan sebuah malware. Oracle memanfaatkan informasi dari VirusTotal untuk menentukan apakah file tersebut merupakan malware.

Alur proses dari verifikasi diawali dengan seller yang mengupload file baru di aplikasi, lalu front-end akan memanggil fungsi “publishIoc” dari IocFactory. IocFactory kemudian akan membuat sebuah Ioc baru dengan parameter yang diberikan oleh front-end. Di dalam constructor Ioc, request verifikasi malware akan diberikan kepada oracle. Lalu oracle akan memanggil VirusTotal API dan memanggil fungsi callback yang telah dibuat.

IV. Design Pattern

Kontrak-kontrak Solidity ini menggunakan beberapa pola desain yang umum digunakan dalam pengembangan aplikasi blockchain untuk menciptakan sistem yang modular, fleksibel, dan mudah dikelola.

- Sistem ini menggunakan Factory Pattern yang diimplementasikan dalam kontrak IocFactory. Pola ini bertujuan untuk mempermudah pembuatan dan pengelolaan banyak instance kontrak Ioc. Setiap instance Ioc mewakili aset unik dan dicatat dalam array publishedIocs. Dengan pendekatan ini, proses pembuatan kontrak baru menjadi terpusat, terstruktur, dan mudah dilacak.

- Sistem ini mengadopsi Modular Contract Design dengan menggunakan interface pada oracle. Pendekatan ini memastikan fleksibilitas dan memisahkan logika implementasi, sehingga memungkinkan pengembang untuk mengganti atau memperbarui oracle tanpa memodifikasi kontrak Ioc.

V. Pembagian Tugas

Nama	NIM	Pembagian Tugas
Christophorus Dharma Winata	13521009	1. Pengembangan Frontend Aplikasi 2. Integrasi dengan Wallet
Muhammad Haidar Akita Tresnadi	13521025	1. Pengembangan backend Aplikasi 2. Integrasi dengan IPFS web3 storage
Moch. Sofyan Firdaus	13521083	1. Pengembangan smart contract 2. Integrasi Oracle dengan smart contract