

Laporan Praktikum: Pengembangan Token ERC-20 KoinAllIn (AIC) dan DApp Terintegrasi MetaMask



Disusun Guna Memenuhi Tugas Mata Kuliah

Sistem Informasi

230202736	Annis Zunaedhah Muthoharoh
230202821	Sri Rahayu
230202739	Ayunita Thalia Nugraeni
230202796	Adelya Ayu Virnanda
230202759	Jamaludin

**UNIVERSITAS PUTRA BANGSA
KEBUMEN PRODI INFORMATIKA
2023\20**

Link youtube :

<https://youtu.be/esCTDYcajWo?si=Lstqyrc9jd3U8O14>

Link repository :

Pendahuluan

Proyek KoinAllIn adalah implementasi lengkap token ERC-20 dengan fitur tambahan burning dan staking menggunakan bahasa pemrograman Solidity. Proyek ini dilengkapi dengan Decentralized Application (DApp) berbasis web yang terintegrasi dengan MetaMask wallet.

Pengembangan proyek memanfaatkan berbagai alat bantu seperti Brownie, Web3.js, dan Infura untuk proses pengembangan, pengujian, serta penyebaran ke jaringan utama Ethereum. Token yang dikembangkan dinamakan **KoinAllIn (AIC)** dengan pasokan awal sebesar **1.000.000 AIC**.

Proyek ini telah berhasil di-deploy ke Ethereum mainnet dan DApp yang dikembangkan telah berfungsi secara penuh (fully functional). Dengan demikian, token KoinAllIn dapat dikategorikan sebagai token yang “nyata” karena dapat diperdagangkan serta diinteraksikan melalui antarmuka web modern.

Apa yang Telah Kita Lakukan Sebelumnya

Dalam pengembangan proyek KoinAllIn, beberapa tahapan penting telah berhasil diselesaikan, antara lain:

- Membuat struktur direktori proyek yang terdiri dari folder kontrak, skrip, dan build.
- Mengubah nama file `TinyToken.sol` menjadi `TinyCoin.sol` di dalam folder kontrak.
- Membuat file konfigurasi `brownie-config.yaml`.
- Mengembangkan script `deploy.py` untuk proses penyebaran smart contract.
- Mengembangkan script `interact.py` untuk interaksi dengan kontrak, termasuk transfer token dan pengecekan saldo.
- Mengatasi permasalahan instalasi, seperti instalasi Microsoft Visual C++ Build Tools untuk menghindari error `cchecksum`.
- Memindahkan proyek ke drive D untuk mengatasi keterbatasan ruang disk.
- Membuat lingkungan virtual di WSL untuk instalasi dependensi Python.
- Menginstal dependensi Python seperti `eth-brownie` dan `web3`.
- Menginstal Ganache sebagai simulasi blockchain lokal.
- Menginstal dependensi tambahan Brownie.
- Menyusun dokumentasi proyek dalam file `README.md` yang berisi instruksi penggunaan.

Kekurangan

Beberapa kekurangan yang masih terdapat dalam proyek ini antara lain:

- Belum tersedia fitur tambahan lanjutan seperti burning token atau minting dinamis secara fleksibel.
- Pengujian unit (unit testing) belum dilakukan secara komprehensif dan masih terbatas pada interaksi dasar.
- Ketergantungan pada alat eksternal seperti Brownie dan Infura yang relatif sulit diinstal pada beberapa sistem.
- Kontrak pintar belum diverifikasi di Etherscan sehingga transparansi publik masih terbatas.

Kelebihan

Adapun kelebihan dari proyek KoinAllIn antara lain:

- Menggunakan standar ERC-20 yang telah terbukti dan kompatibel dengan berbagai wallet serta exchange.
- Penggunaan framework Brownie mempermudah proses deployment dan interaksi smart contract.
- Struktur proyek tertata dengan baik melalui pemisahan folder kontrak, skrip, dan build.
- Pasokan token awal yang terbatas menjadikan proyek ini cocok untuk keperluan demonstrasi dan pembelajaran.
- Kode Solidity yang digunakan relatif sederhana dan mudah dipahami, mencakup fungsi `transfer`, `approve`, dan `transferFrom`.

Kendala

Selama proses pengembangan, beberapa kendala yang dihadapi meliputi:

- Sistem operasi Windows memerlukan instalasi tambahan seperti Microsoft Visual C++ Build Tools yang cukup kompleks.
- Keterbatasan ruang penyimpanan pada drive utama sehingga proyek harus dipindahkan.
- Ketergantungan pada Python 3.8+, Node.js, dan npm yang harus terinstal terlebih dahulu.
- Penggunaan WSL sebagai lingkungan virtual yang belum familiar bagi sebagian pengguna Windows.
- Konfigurasi compiler Solidity (solc) yang memerlukan pengaturan path secara spesifik.

Langkah-Langkah Pembuatan dan Progres

1. Persiapan lingkungan pengembangan dengan menginstal Python, Node.js, npm, dan Visual C++ Build Tools.
2. Pembuatan struktur direktori proyek serta file konfigurasi.
3. Pengembangan smart contract KoinAllIn dengan fitur ERC-20, burning, dan staking.
4. Pembuatan script deployment untuk penyebaran ke jaringan Ethereum.
5. Pembuatan script interaksi untuk transfer token dan pengecekan saldo.
6. Konfigurasi Brownie melalui file `brownie-config.yaml`.
7. Instalasi seluruh dependensi yang dibutuhkan.
8. Pengujian kontrak menggunakan Ganache dan MetaMask.
9. Penyusunan dokumentasi proyek.

Cara Menjalankan Proyek

Instalasi Prasyarat

- Python 3.8 atau lebih baru
- Node.js dan npm
- Microsoft Visual C++ Build Tools

Instalasi Dependensi

```
pip install eth-brownie web3
```

Atau menggunakan virtual environment:

```
python -m venv venv
venv\Scripts\activate
pip install eth-brownie web3
```

Menjalankan Ganache

```
ganache
```

Deployment Kontrak

```
brownie run skrip/deploy_patched.py --network mainnet
```

Interaksi Kontrak

```
brownie run skrip/interaksi.py --network mainnet
```

Pengembangan DApp MetaMask

Fitur DApp Utama

DApp KoinAllIn menyediakan beberapa fitur utama, yaitu integrasi MetaMask, pengecekan saldo, transfer token, burning token, staking token, pengecekan reward, serta penarikan staking.

DApp dilengkapi dengan validasi input, error handling, serta tampilan antarmuka yang responsif dan user-friendly.

Penting: Kebutuhan Gas Fee

Transaksi belum dapat dilakukan apabila wallet MetaMask tidak memiliki ETH sebagai gas fee. Setiap transaksi blockchain memerlukan ETH untuk diproses.

Tanpa saldo ETH, seluruh fungsi transaksi seperti transfer, burn, stake, dan withdraw akan gagal.

Kesimpulan

Proyek KoinAllIn berhasil dikembangkan sebagai implementasi nyata token ERC-20 yang terintegrasi dengan DApp berbasis web dan MetaMask. Proyek ini mencakup pengembangan smart contract, deployment ke Ethereum mainnet, serta penyediaan antarmuka pengguna yang memungkinkan interaksi blockchain secara langsung.

Dokumentasi

The image displays a development environment with a code editor, a terminal, and a web browser. The code editor shows a Python script named `install_solc.py` that installs Solc version 0.8.0. The terminal output shows the successful execution of the script and the deployment of a contract. The web browser shows a web interface for staking tokens, with a 'Stake' button and a 'Staking Info' section. A MetaMask wallet is also visible, showing a transaction for 0 ETH.

```
1 from solcx import install_solc
2
3 # Install solc version 0.8.0
4 install_solc("0.8.0")
5
6 print("solc 0.8.0 installed successfully!")
7
```

OUTPUT

```
- /usr/local/lib/node_modules/ganache/node_modules/@trufflesuite/usb-js-unofficial/src/usb.js
- /usr/local/lib/node_modules/ganache/dist/node/cli.js
Falling back to a NodeJS implementation; performance may be degraded.

Launching 'ganache-cli --chain.vmErrorsOnRPCResponse true --server.port 8545 --miner.blockGasLimit 12000000 --wallet.totalAccounts 10 --hardfork istanbul --wallet.mnemonic brownie'...

Running 'scripts/deploy.py:main'...
Transaction sent: 0xd6de65e13cf8b20ee94c97efc5854c33405a7f5c4c05a474d8ac3f10398e3d05
Gas prices: 0.0 gwei Gas limit: 6721973 Nonce: 0
TinyCoin.constructor confirmed Block: 1 Gas used: 566583 (8.43%)
TinyCoin deployed at: 0xc3194c8dc3dbcd3e11a07892e7ba5c3394048cc87

Contract deployed at: 0xc3194c8dc3dbcd3e11a07892e7ba5c3394048cc87
Deployer balance: 1e+21 TNC
Terminating local RPC client...
PS D:\Tinycoin>
```

MetaMask Developer Dashboard

Stake Tokens

15 Stake

Stake failed

Staking Info

Check My Stake

My Stake: 0 AIC

Withdraw Stake 8

Rewards

Check Reward

Reward: 0 AIC

MetaMask

Permintaan transfer

0 ETH

Dari Account 1 0xD84E1_b...

Estimasi perubahan Tidak ada perubahan

Jaringan Ethereum

Permintaan dari t0hlcdbn-8000.asse.devttunnels.ms

Biaya jaringan US\$0.16 ETH

Kecepatan Situs yang disarankan ~12 dtk

Batal Tinjau peringatan





