

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.

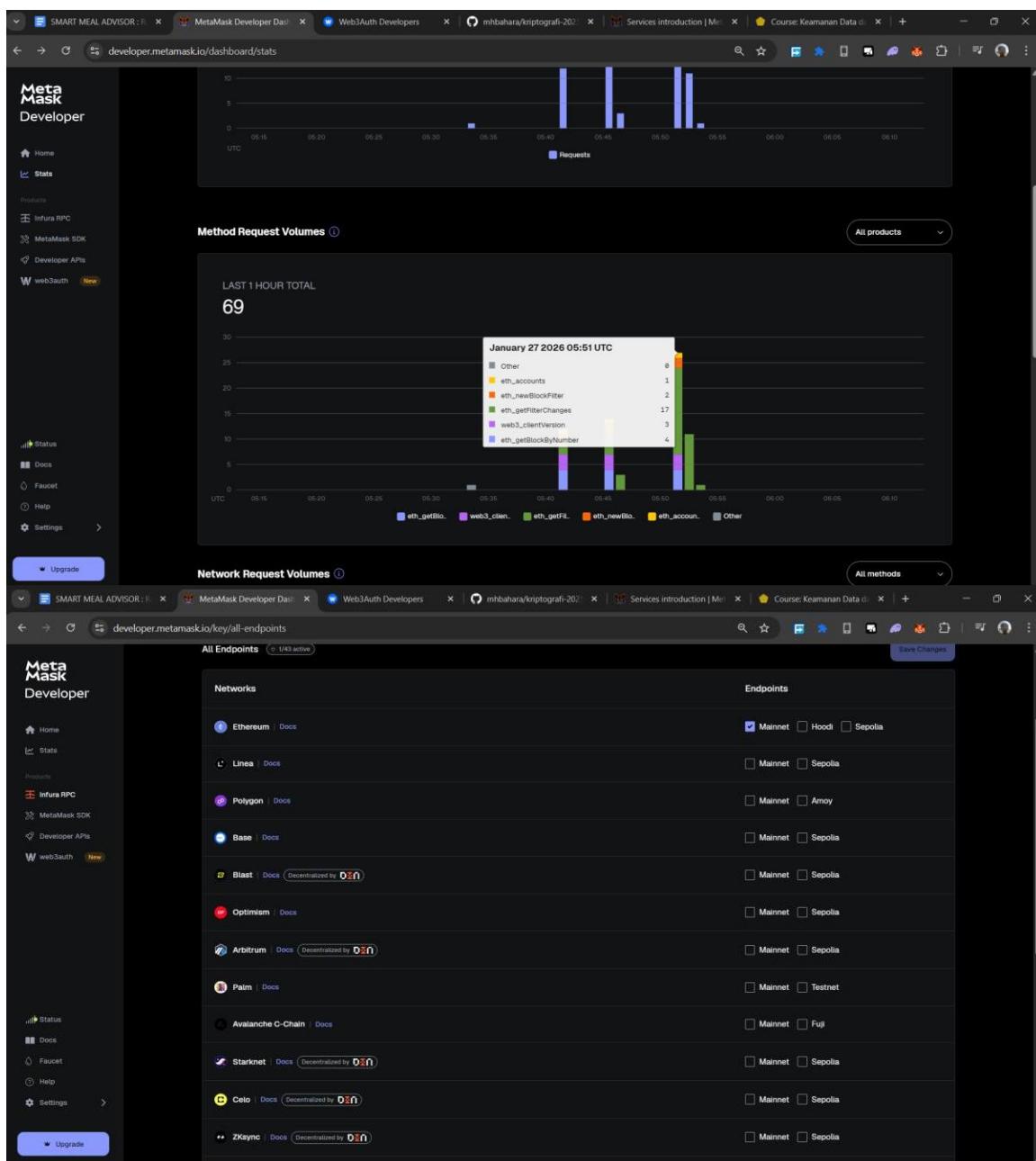
Kesimpulan

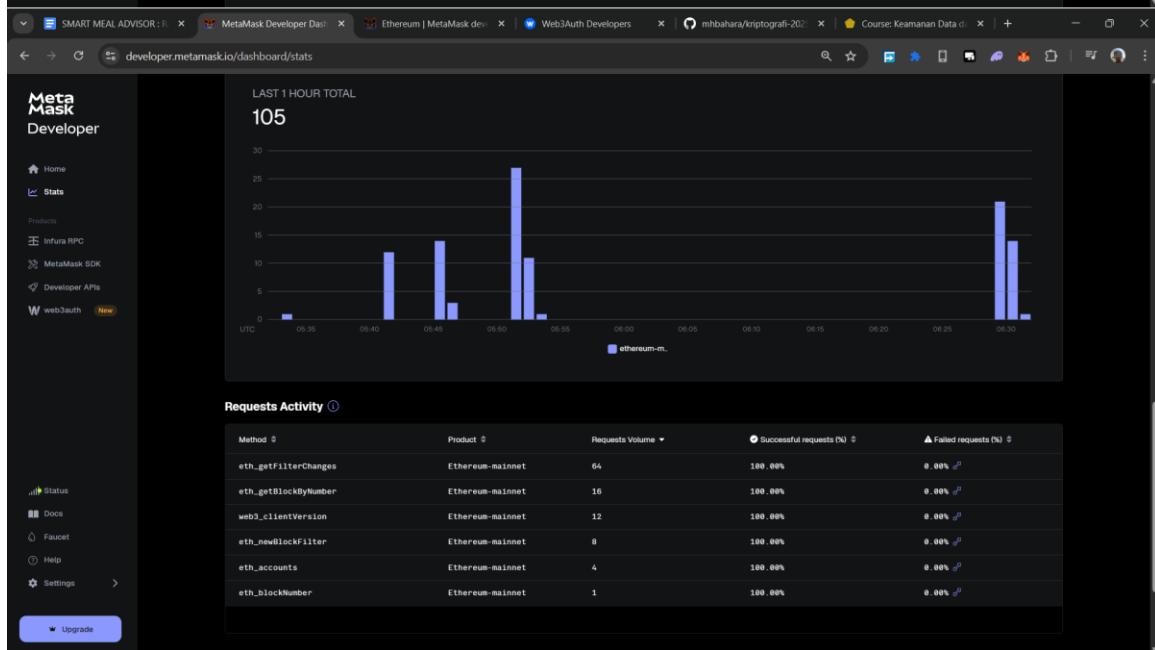
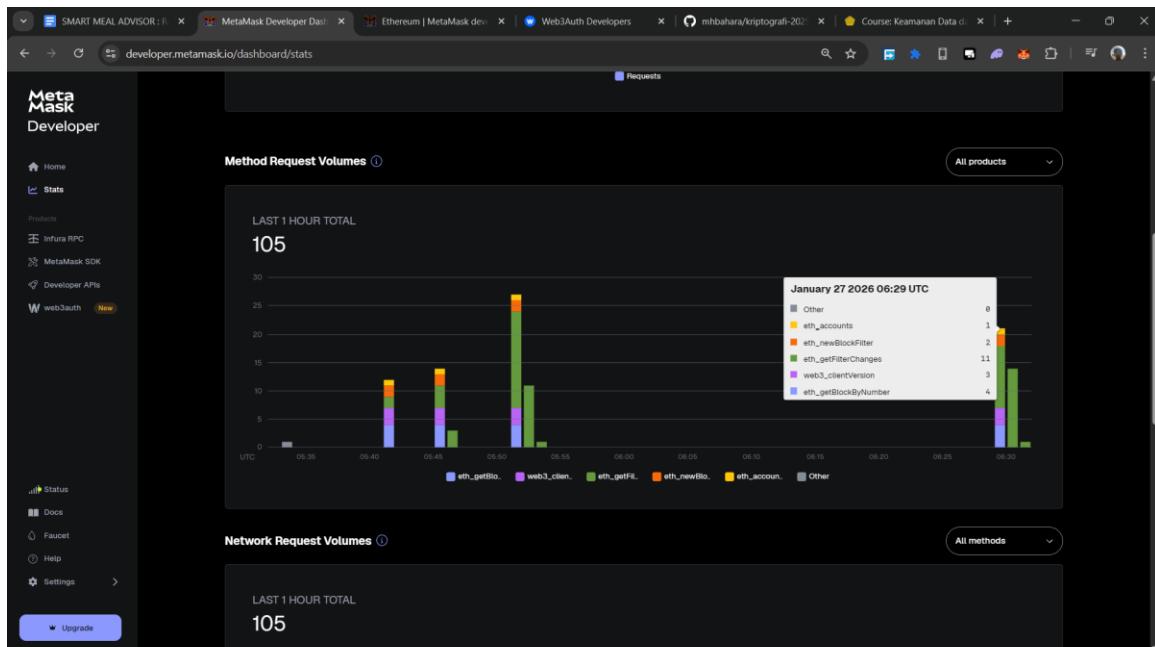
Proyek KoinAllIn berhasil dikembangkan sebagai implementasi nyata token ERC-20 yang terintegrasi dengan DApp berbasis web dan MetaMask. Proyek ini mencakup pengembangan smart contract, deployment ke Ethereum mainnet, serta penyediaan antarmuka pengguna yang memungkinkan interaksi blockchain secara langsung.

Dokumentasi

The screenshot shows a developer's workspace with the following components:

- Code Editor (VS Code):** The left sidebar shows project structures for `TINYCOIN` and `OPEN EDITORS`. The `install_solc.py` file in the `OPEN EDITORS` section is open, containing Python code to install Solc version 0.8.0 using `solcx`.
- Terminal:** The terminal window shows the command being run: `./scripts/deploy.py:main`. It displays transaction details, including the contract address (`0x3194cB0C3dbcd3E11a07892e7bA5c3394048Cc87`) and the deployed code hash.
- Browser Tabs:** The browser has several tabs open, including `MetaMask Developer Dashboard`, `mhbahara/kriptografi-202501`, and `KoinAllin (AIC) DApp`.
- Bottom Status Bar:** Shows the current file path as `PS D:\tinycoin`, along with other system information like Python version and file encoding.
- Staking Application:** A local application window titled "Stake Tokens" is open, showing a stake amount of 15 and a failed stake attempt.
- Staking Info:** Another part of the application shows "My Stake: 0 AIC" and a "Withdraw Stake" button.
- Rewards:** A third section shows "Reward: 0 AIC" and a "Check Reward" button.
- MetaMask Wallet:** A MetaMask wallet interface is shown, displaying a pending transfer from account `Account 1` to `0xD84E1_b...` with a value of 0 ETH.





Screenshot of a terminal window showing the deployment of a KoinAllIn DApp using Brownie and MetaMask.

```

brownie config.yaml U TODO.md U index.html U Laporan_Penuh.md U tinyConsole install_checksum.bat U settings.json
File Edit Selection View Go Run Terminal Help ← → Tanycoin [Administrator] 08 - + ×
EXPLORER OPEN EDITORS
Laporan_Penuh.md > # Laporan Penuh Projek KoinAllIn ERC-20 dengan DApp MetaMask > ## Kesimpulan
1 # Laporan Penuh Projek KoinAllIn ERC-20 dengan DApp MetaMask
94 ## Pengembangan DApp MetaMask
140 ### Keamanan DApp
144 - Validasi input di client-side dan contract-side
145
146 ## Kesimpulan
147 proyek KoinAllIn telah berhasil dikembangkan menjadi ekosistem lengkap yang terdiri dari:
148 - Smart contract ERC-20 dengan fitur burning dan staking
149 - Deployment script untuk mainnet
150 - DApp web modern dengan MetaMask integration
151 - Infura connection untuk reliability
152 - Comprehensive error handling dan user experience
153 - Real-time blockchain interactions
154
155 Token AIC dapat diperdagangkan di Ethereum mainnet dengan interface yang user-friendly untuk interaksi sehari-hari.
156

TINYCON
reports skip tests venv
brownie-config... HelloWorld.sol index.html install_checksum... install_solidity...
JuanBlanco.solidity Laporan_Penuh... Miniconsole.exe New Text Docu... package-lock.json package.json README.md solc TODO.md

OUTLINE
TIMELINE
CHECKPOINTS
HIDDEN ITEMS
LOCAL HISTORY
main* 501 1 Live Share Git Graph
MetaMask Developer Dash MetaMask Developer Dashboard Konten channel - YouTube KoinAllIn (AIC) DApp KoinAllIn (AIC): Membangun Laporan Projek KoinAllIn + - ×
developer.metamask.io

MetaMask Developer
Home Stats Products Infura RPC MetaMask SDK Developer APIs Web3Auth Now
Analytics Overview
Status Docs Faucet Help Settings Upgrade

```

TERMINAL

```

PORTS Forwarded Add... Port 0 8000 https:// Add Port
file "/usr/lib/python3.12/importlib/_init_.py", line 98, in import_module
    return _bootstrap._gcd_import(name[level:], package, level)
file "<frozen>", line 1, in in
from brownie import network, accounts, KoinAllIn # Import dari Brownie untuk jaringan, akun, dan kontrak
ImportError: cannot import name 'KoinAllIn' from 'brownie' (/mnt/d/tinycoint-venv/lib/python3.12/site-packages/brownie/_init_.py)
(tinycoint-venv) mal@DESKTOP-1C21F02:/mnt/d/tinycoint-venv$ Deploy KoinAllIn ke mainnet
(tinycoint-venv) mal@DESKTOP-1C21F02:/mnt/d/tinycoint-venv$ 

```

API Keys

Name	Daily Request Health %	Daily Requests	Daily Credit Usage	Updated	Actions
My First Key	100%	188	22,300	Jan 27, 2026	Configure Copy Key

Let's Get Started

Start building. Explore the Resource Hub below to start building a winning Web3 app in no time.

Send Requests
Test sending a variety of requests directly from our documentation.
[Send requests →](#)

Integrate MetaMask
Use MetaMask SDK to connect user wallets to your app.
[Integrate metamask →](#)

MetaMask Developer Center
A series of interactive lessons to begin your Web3 journey.
[Start learning →](#)

Need help getting started?
Share your needs on Discord to get personalized support.
[Access our community →](#)

Analytics Overview

Daily Request Health

188 Total

Success (100%) Failed (0%)

Daily Credit Usage

22,300 of 3M used

2.98M available

eth_getFilter... eth_getBlock... web3_client... eth_newBla... eth_accounts eth_blockN...