

WHAT IS BLOCKCHAIN?

Kelompok

BITCOIN

merupakan salah satu protokol pertama yang menggunakan teknologi revolusioner yang disebut dengan blockchain.

Buku putih Bitcoin dibuat oleh pseudo anonim Satoshi Nakamoto, dan menguraikan bagaimana Bitcoin dapat melakukan transaksi peer to peer.

Dalam jaringan terdesentralisasi jaringan ini didukung oleh kriptografi, Tality yang layak, dan memungkinkan orang untuk terlibat dalam keuangan tahan sensor, dengan cara yang terdesentralisasi karena fitur-fiturnya.

Pengembang Bitcoin memandang Bitcoin sebagai penyimpan nilai versus Aetherium, pengembang memandang Aetherium sebagai penyimpan nilai dan utilitas untuk memfasilitasi perjanjian terdesentralisasi ini. Sekarang, kontrak pintar di blockchain saja benar-benar luar biasa. Namun, mereka datang dengan masalah besar. Jika kita ingin perjanjian digital ini menggantikan perjanjian dalam kehidupan kita sehari-hari, mereka mungkin akan membutuhkan data dari blockchain dunia nyata sendiri yang sebenarnya tidak dapat berinteraksi dengan dan tidak dapat membaca atau mendengarkan data dari dunia nyata. Inilah yang dikenal sebagai masalah Oracle.

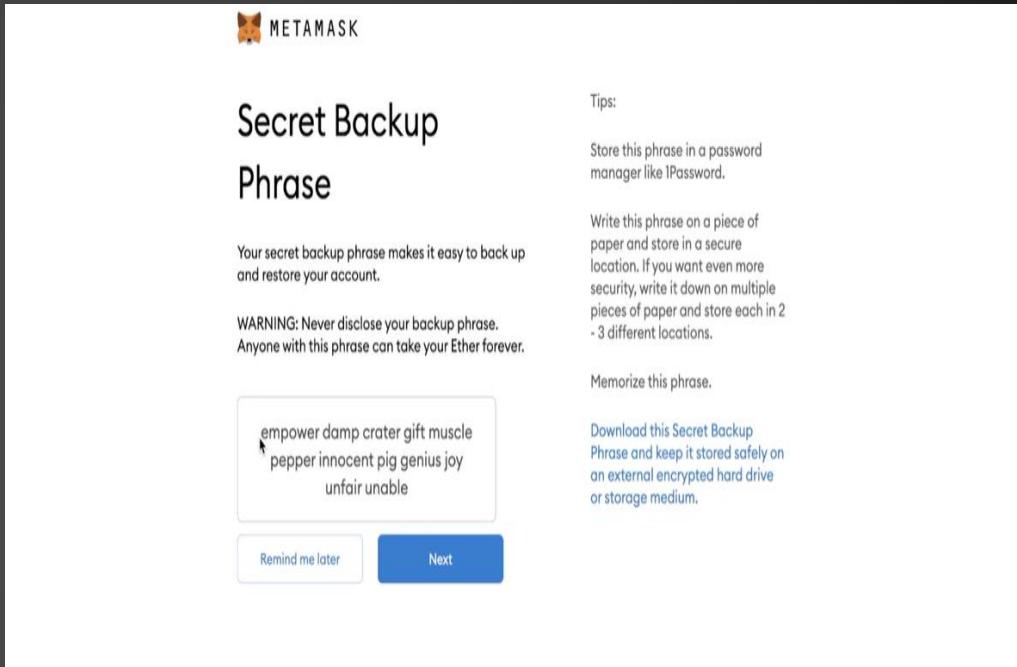
Blockchain ini adalah sistem deterministik dan mereka deterministik dengan sengaja. Oracle adalah perangkat apa pun yang mengirimkan data ke blockchain terdesentralisasi ini atau menjalankan komputasi eksternal. Jadi membutuhkan jaringan Oracle yang terdesentralisasi, maka menggunakan logika rantai akan terdesentralisasi, tetapi juga perlu data off chain dan komputasi Anda didesentralisasi, menggabungkan ini pada logika terdesentralisasi rantai.

Dengan off chain ini, data terdesentralisasi dan komputasi terdesentralisasi menimbulkan sesuatu yang disebut kontrak pintar hibrida. Dan sebagian besar protokol terbesar yang berinteraksi dengan kita saat ini adalah beberapa jenis kontrak pintar hybrid atau berinteraksi dengan kontrak pintar hybrid. Sampai batas tertentu, di sinilah rantai protokol berperan. Ini adalah jaringan Oracle modular dan terdesentralisasi yang dapat membawa data eksternal dan komputasi eksternal ke dalam kontrak pintar kami untuk memastikan mereka terdesentralisasi dari ujung ke ujung, sambil memberi mereka kekayaan fitur yang dibutuhkan untuk perjanjian ini.

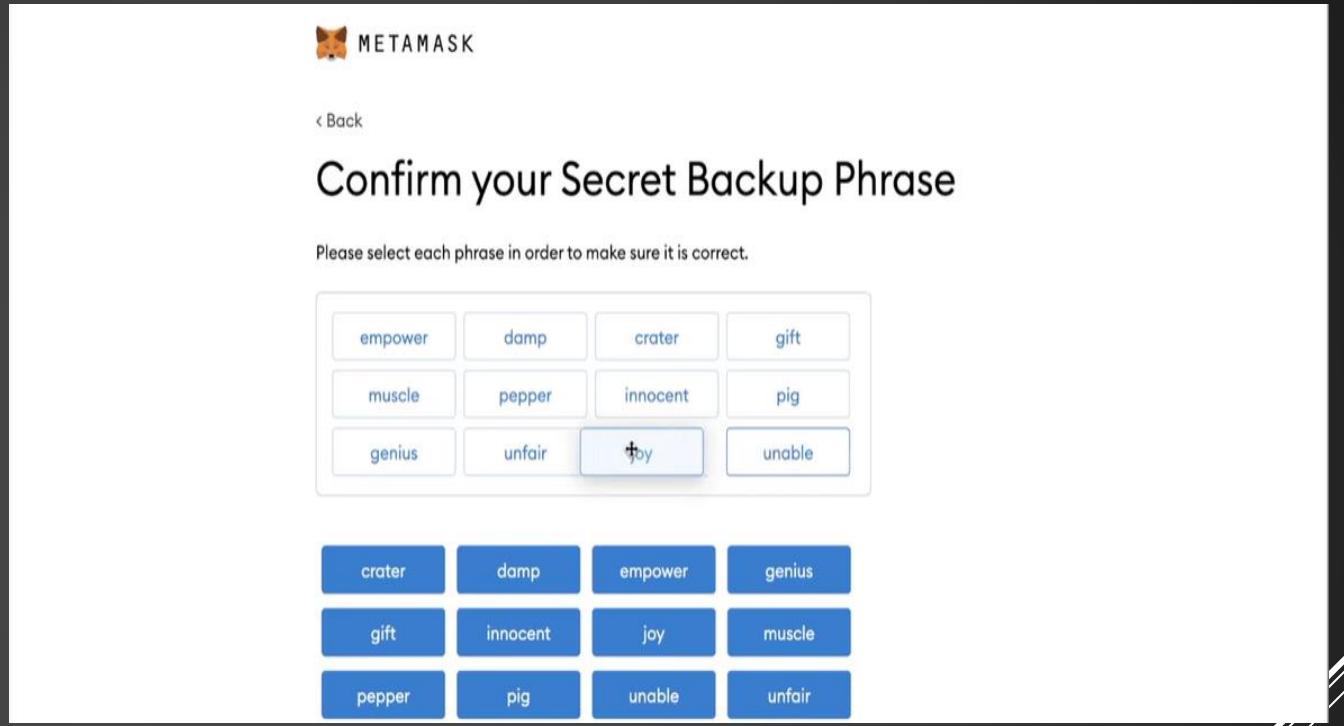
Variabel pada dasarnya adalah pemegang untuk nilai yang berbeda. Misalnya, kita dapat membuat variabel yang disebut memiliki nomor favorit untuk mewakili jika seseorang memiliki nomor favorit.

fungsi atau metode adalah modul mandiri yang akan menjalankan beberapa set instruksi khusus untuk kita, ketika kita menyebutnya jika Anda terbiasa dengan Java, atau Python, atau JavaScript atau semacamnya fungsi itu bekerja dengan cara yang sama persis dengan fungsi yang diidentifikasi oleh fungsi kata kunci,

KITA AKAN MEMBUAT META MASK
BARU.

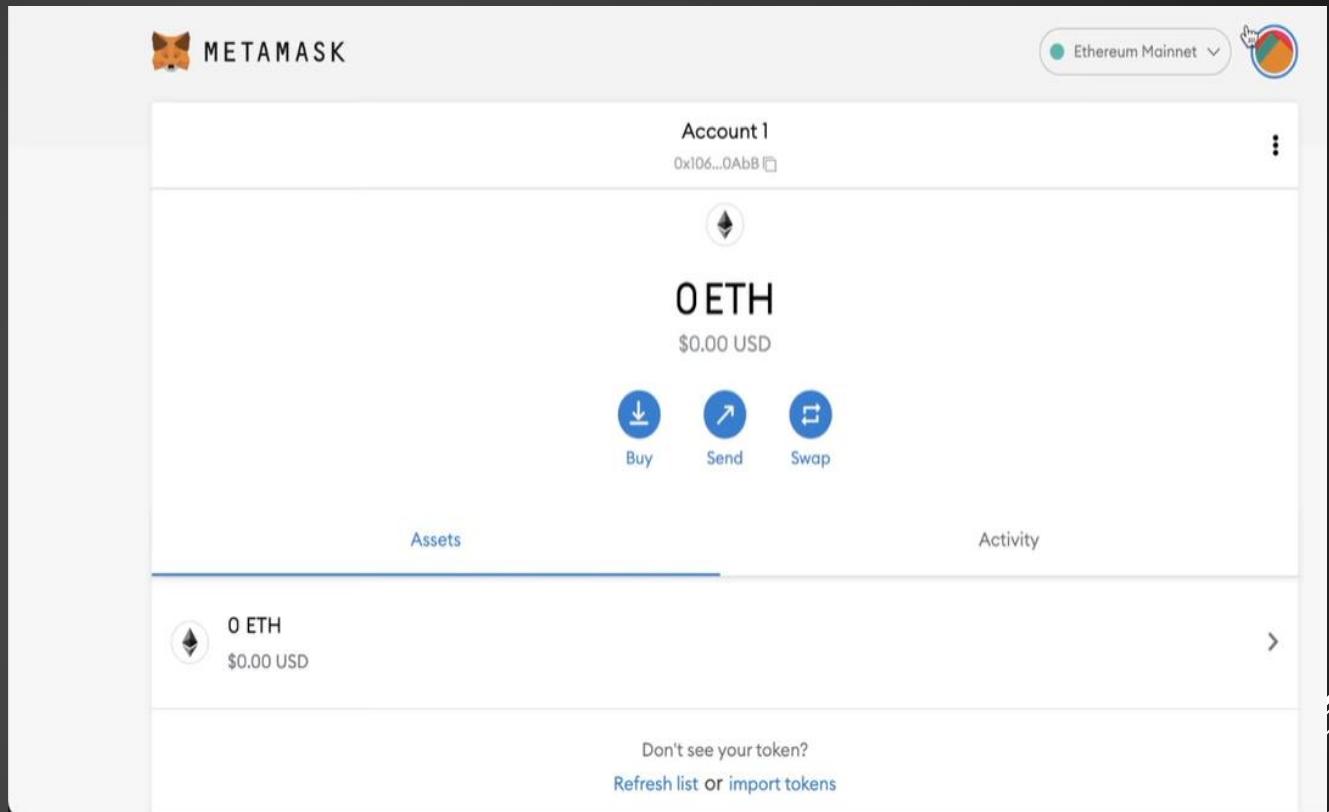


SUSUN MENJADI SEBUAH
KALIMAT UNTUK MEMVERIFIASI
AKUN



SETELAH MELEWATI BEBERAPA
VERIFIKASI AKUN METAMASK
MAKA KITA AKAN MEMILIKI
AKUN BARU





The Ethereum Blockchain Explorer

All Filters

ETHER PRICE
\$1,128.61 @ 0.05633 BTC (+3.30%)

MARKET CAP
\$134,954,004,988.00

TRANSACTIONS
1,629.90 M (14.0 TPS)

DIFFICULTY
11,654.66 TH

MED GAS PRICE
15 Gwei (\$0.38)

HASH RATE
893,897.67 GH/s

ETHERUM TRANSACTION HISTORY IN 14 DAYS

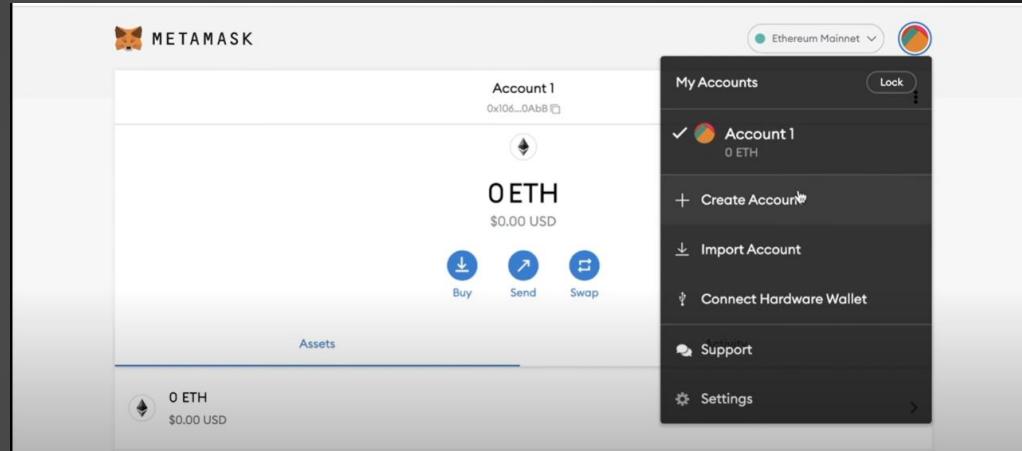
Latest Blocks

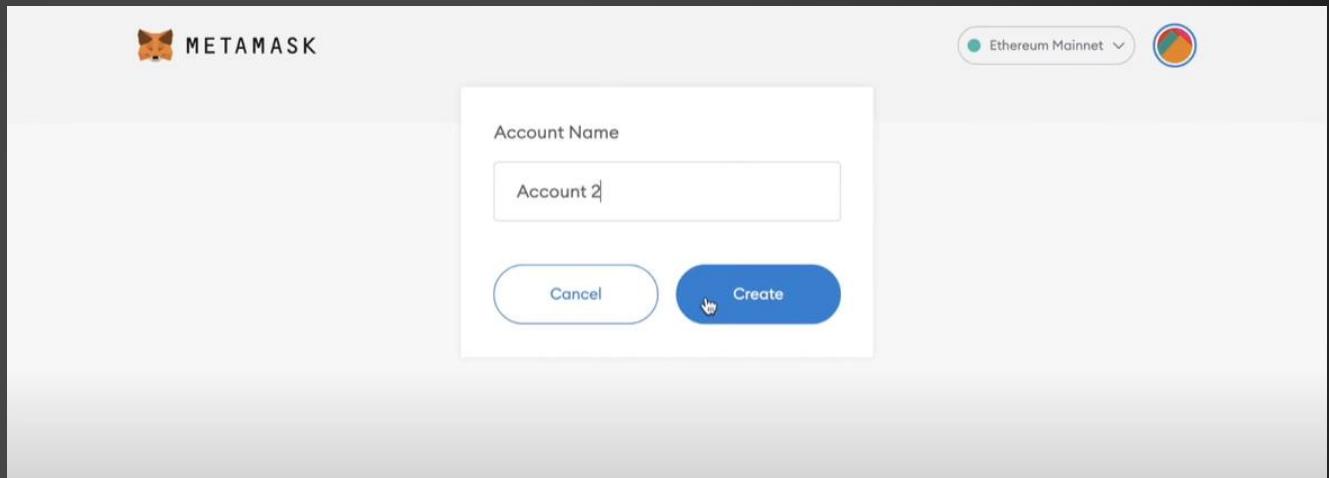
Bk	15088876	Miner MiningPoolHub	2.15121 Eth
Bk	15088875	Miner Poolin 2	2.19511 Eth
Bk	15088874	Miner 2Miner	139 bns in

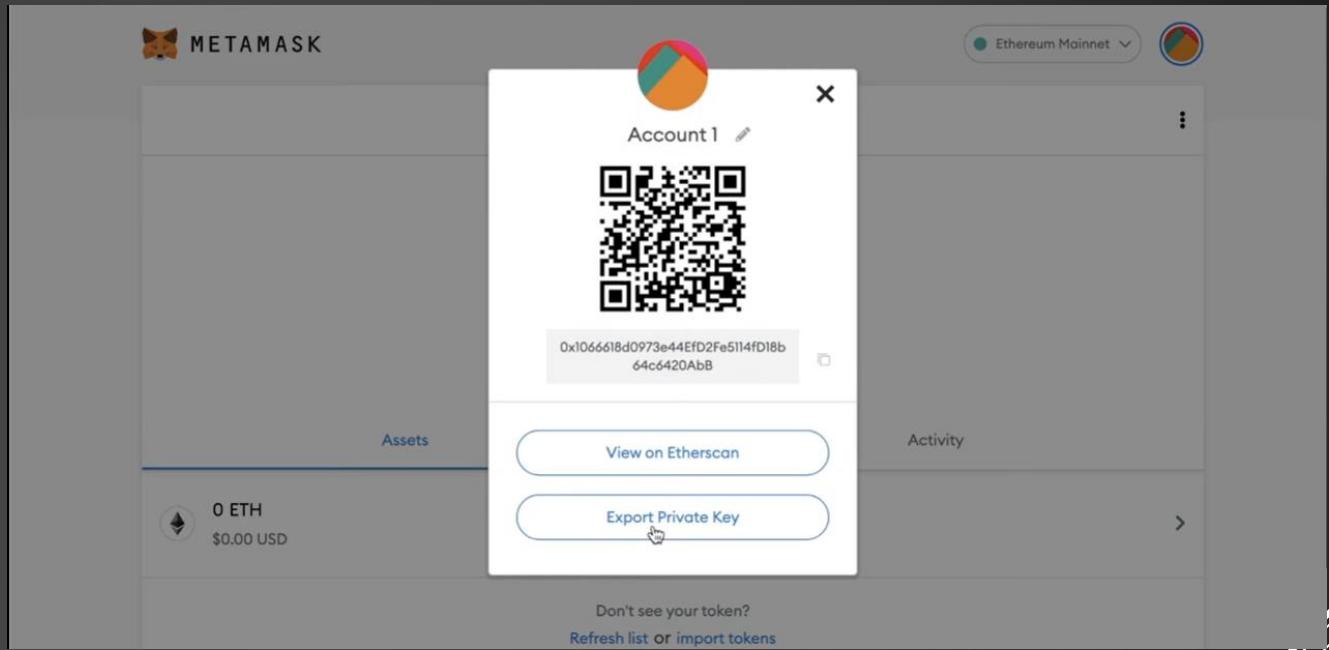
Latest Transactions

Tx	0xdbc489339761...	From 0xda801201925b1873e9...	0 Eth
Tx	0x552559a5c269...	From 0x39a708a6b505ba140b...	4.33899 Eth
Tx	0x1100	From 0x1100	0 Eth

This website uses cookies to improve your experience and has an updated Privacy Policy.







SmartContractSolidityBlockchainSolidityCourse.js

Code Issues Pull requests Actions Projects Wiki Security Insights

main 1 branch 0 tags Go to file Code About

PatrickAlphaC added faucet af815b4 now 12 commits

.outline.md wip 20 days ago

README.md added faucet 17 seconds ago

chronological-updates.txt added updates file 13 minutes ago

README.md

Welcome to the repository for the Ultimate Solidity, Blockchain, and Smart Contract - Beginner to Expert Full Course | Javascript Edition FreeCodeCamp course!

All code references have both a javascript and a typescript edition.

Recommended Testnet: Rinkeby

Testnet Faucets: <https://faucets.chain.link>

No description, website, or topics provided.

Readme 0 stars 1 watching 0 forks

No releases published

No packages published

The screenshot shows a GitHub repository page for a project named "SmartContractSolidityBlockchainSolidityCourse.js". The repository has 12 commits, 1 branch, and 0 tags. The README.md file contains the following text:

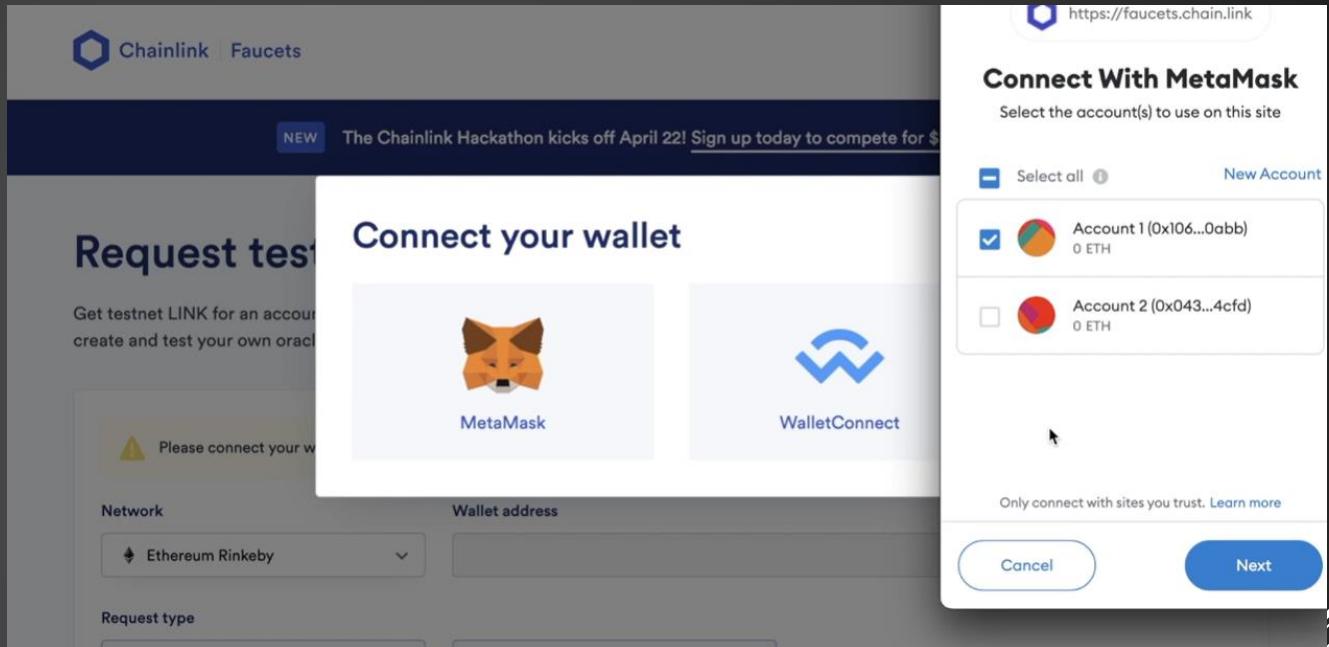
```
Welcome to the repository for the Ultimate Solidity, Blockchain, and Smart Contract - Beginner to Expert Full Course | Javascript Edition FreeCodeCamp course!

All code references have both a javascript and a typescript edition.

Recommended Testnet: Rinkeby

Testnet Faucets: https://faucets.chain.link
```

The repository has no description, website, or topics provided. It also has no releases or packages published.



Request testnet LINK

Get testnet LINK for an account on one of the supported blockchain testnets so you can create and test your own oracle and Chainlinked smart contract. [Learn more](#)

Network

Ethereum Rinkeby

Wallet address

0x1066618d0973e44efd2fe5114fd18b64c6420abb

Request type

10 test LINK

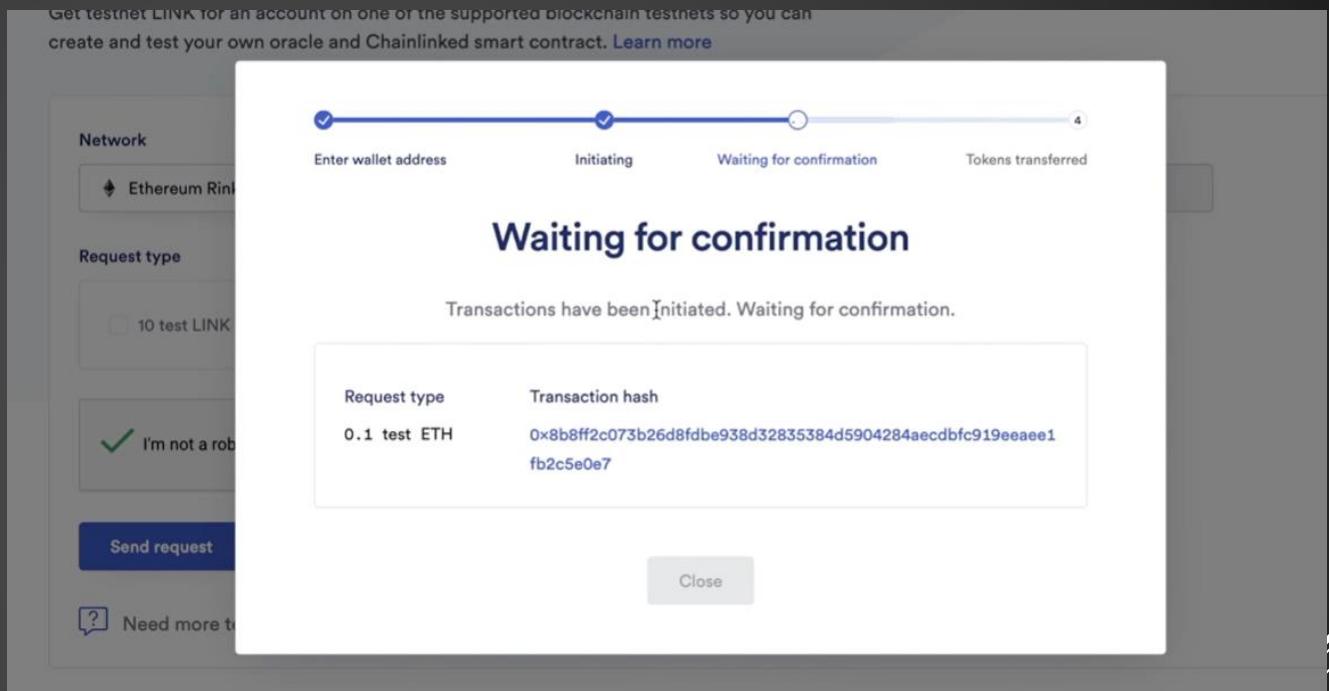
0.1 test ETH

I'm not a robot



Privacy - Terms

[Send request](#)



Overview	State
[This is a Rinkeby Testnet transaction only]	
⑦ Transaction Hash:	0x8b8ff2c073b26d8fbe938d32835384d5904284aecdbfc919eeaae1fb2c5e0e7 ⓘ
⑦ Status:	Success
⑦ Block:	10394763 45 Block Confirmations
⑦ Timestamp:	⑧ 11 mins ago (Mar-26-2022 04:09:16 PM +UTC)
⑦ From:	0xa7a82dd06901f29ab14af63faf3358ad101724a8 ⓘ
⑦ To:	0x1066618d0973e44efd2fe5114fd18b64c6420abb ⓘ
⑦ Value:	0.1 Ether (\$0.00)
⑦ Transaction Fee:	0.000052500000483 Ether (\$0.00)
⑦ Gas Price:	0.000000002500000023 Ether (2.500000023 Gwei)

Etherscan

Rinkeby Testnet Network

Address **0x1066618d0973e44EfD2Fe5114fD18b64c6420AbB**

Overview

Balance: 0.1 Ether

Transactions

Latest 1 from a total of 1 transactions

Txn Hash	Method ⓘ	Block	Age	From ↴
0x8b8ff2c073b26d8fdbe...	Transfer	10394763	13 mins ago	0xa7a82dd

A wallet address is a publicly available address that allows its owner to receive funds from another party. To

All  Rinkeby Test Network  ken / Ens

Not connected Account 1 0x106...0AbB

Copy to clipboard

0.1ETH

 Buy  Send  Swap

Assets Activity

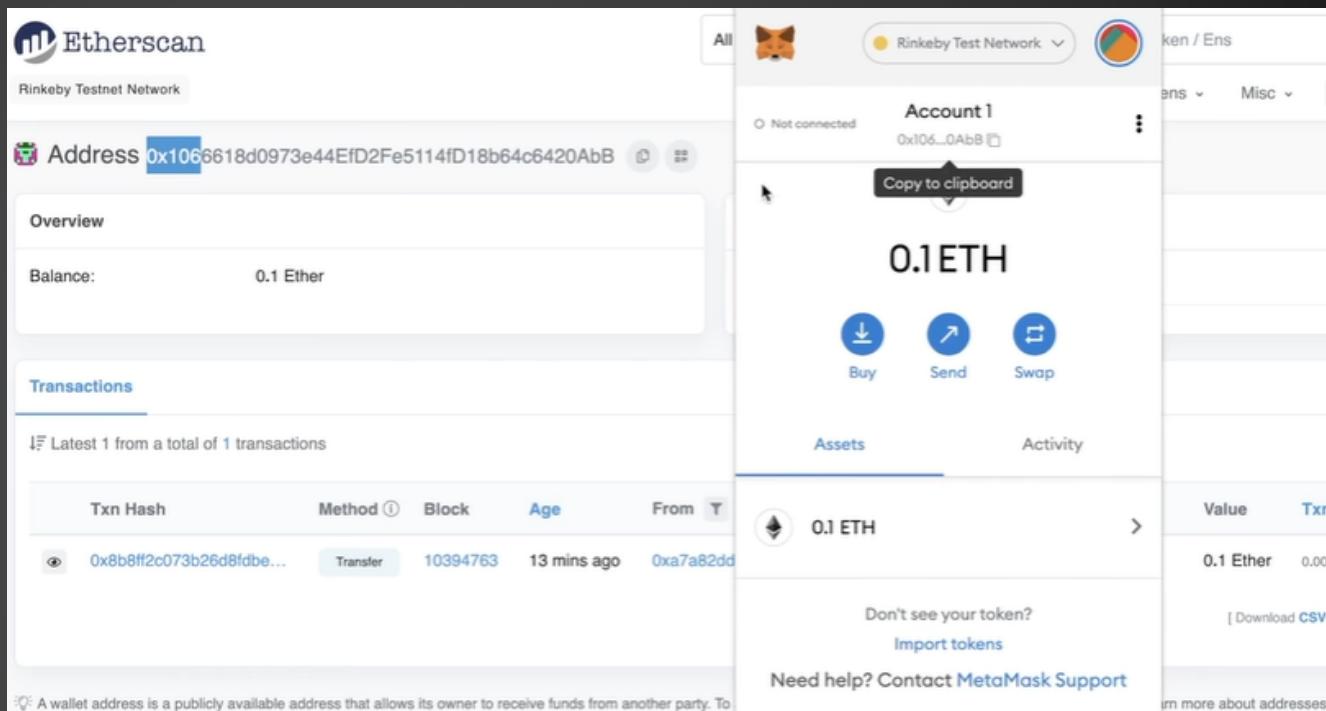
0.1 ETH

Don't see your token? Import tokens

Need help? Contact MetaMask Support

[Download CSV]

im more about addresses



Overview	State
② From:	0xa7a82dd06901f29ab14af63faf3358ad101724a8 Copy
② To:	0x1066618d0973e44efd2fe5114fd18b64c6420abb Copy
② Value:	0.1 Ether (\$0.00)
② Transaction Fee:	0.000052500000483 Ether (\$0.00)
② Gas Price:	0.000000002500000023 Ether (2.500000023 Gwei)
Click to see More ↓	

[Overview](#) [State](#)

Block: 10394763 45 Block Confirmation

Timestamp: 11 mins ago (Mar-26-2022 04:15:11)

From: 0xa7a82dd06901f29ab14af63faf3

To: 0x1066618d0973e44efd2fe5114f

Value: 0.1 Ether (\$0.00)

Transaction Fee: 0.000052500000483 Ether (\$0.00)

Gas Price: 0.000000002500000023 Ether (2)

Rinkeby Test Network

Account 1
0x106...0Ab8

Not connected

0.1 ETH

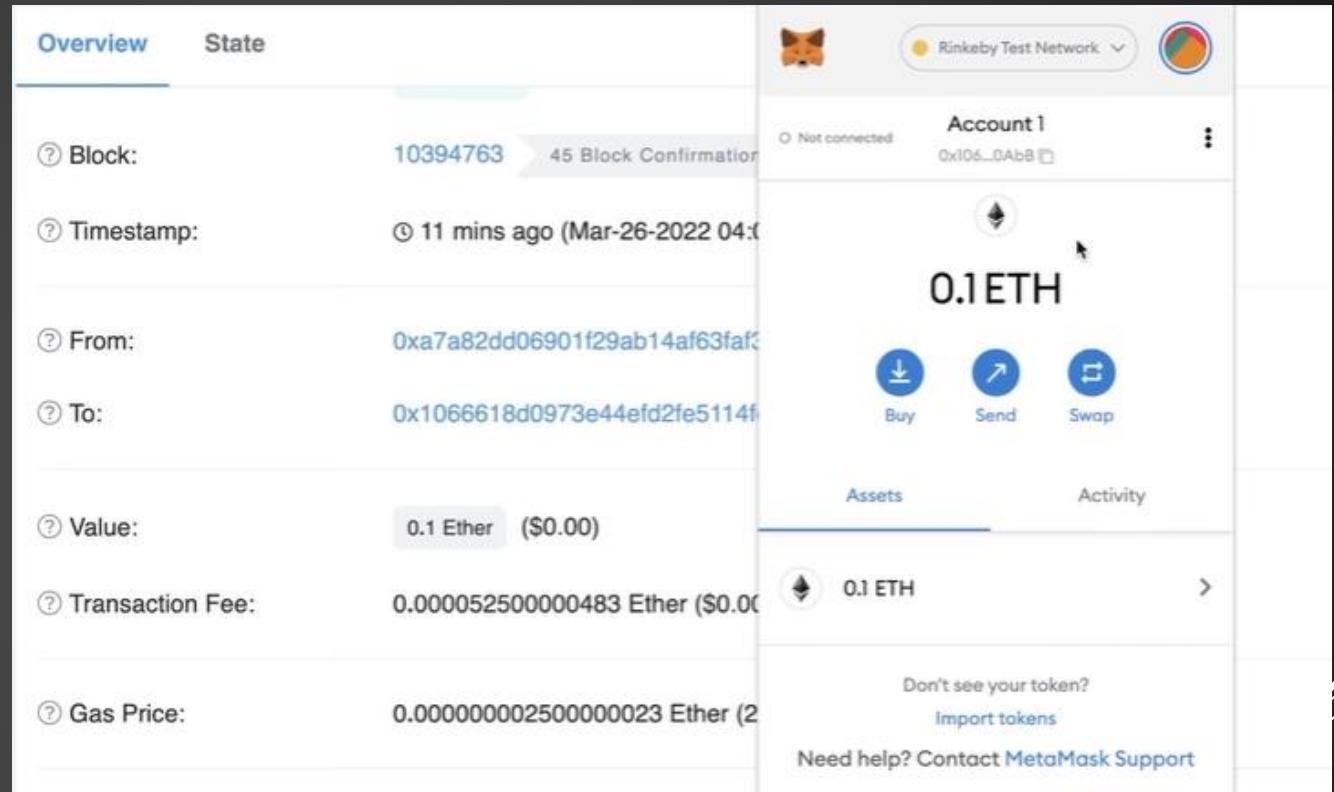
Buy Send Swap

Assets Activity

0.1 ETH

Don't see your token?
[Import tokens](#)

Need help? Contact [MetaMask Support](#)



Overview State

Block: 10394763 45 Block Confirmation

Timestamp: 11 mins ago (Mar-26-2022 04:00)

From: 0xa7a82dd06901f29ab14af63faf3

To: 0x1066618d0973e44efd2fe5114f

Value: 0.1 Ether (\$0.00)

Transaction Fee: 0.000052500000483 Ether (\$0.00)

Gas Price: 0.000000002500000023 Ether (2)

Rinkeby Test Network

Account 1 Account 2

New gas experience

We've updated how gas fee estimation and customization works.

Turn on Enhanced Gas Fee UI in Settings

SENDING ETH

0.05

EDIT

Estimated gas fee 0.00004448
0.000046 ETH

Likely in < 30 seconds Max fee: 0.00004648 ETH

Total 0.05004648
0.05004648 ETH

Amount + gas fee Max amount: 0.05004648 ETH

Overview State

Block: 10394763 45 Block Confirmation

Timestamp: 11 mins ago (Mar-26-2022 04:00)

From: 0xa7a82dd06901f29ab14af63faf3

To: 0x1066618d0973e44efd2fe5114f

Value: 0.1 Ether (\$0.00)

Transaction Fee: 0.0000052500000483 Ether (\$0.00)

Gas Price: 0.000000002500000023 Ether (2 GWEI)

Rinkeby Test Network

Edit gas fee

Gas option	Time	Max fee
Low	30 sec	0.00003... ETH
Market	30 sec	0.00005... ETH
Aggressive	15 sec	0.00005... ETH
Advanced	--	--

Network status: 0 GWEI Base fee 2 - 3 GWEI Priority fee Busy

Want to learn more about gas?

Reject Confirm

SHA256 Hash

Data: asdfadfadfasfas

Hash: 7f50d069eb0ab8384fad2a3f85cc9213b37b369cc4bb0eb726de24c5f48fab

Blockchain Demo

Hash Block Blockchain Distributed Tokens Coinbase

Block

Block:	# 1
Nonce:	72608
Data:	(empty)
Hash:	0000f727854b50bb95c054b39c1fe5c92e5ebcfa4bcb5dc279f56aa96a365e5a

Mine

Blockchain Demo

Hash Block Blockchain Distributed Tokens Coinbase

Block

Block:	# 1
Nonce:	72608
Data:	asdfasdfasdfasdfasdfadsñdadsa
Hash:	5cafb422c23e133302ee6eae2059d27f4bac7ab903578701fa9e17f06b912d22

Mine

Blockchain Demo

Hash Block Blockchain Distributed Tokens Coinbase

Block

Block:	# 1
Nonce:	33128
Data:	asdfasdfadsfadfasdfad
Hash:	0000b75b1fc92d5a4207c86f81d9e2fdabc959ed31fea07b27538f5dde2a4c2

[Mine](#)

Demo Rantai Blok

Cincang Blok Rantai Blok Terdistribusi Token-token Coinbase

Rantai Blok

Blok:	#	1
Nonce:	11316	
Data:	[Empty]	
Sebelumnya:	00	
Cincang:	000015783b764259d382017d91a36d206d0600e2ccb3567748f	
<button>Tambang</button>		

Blok:	#	2
Nonce:	35230	
Data:	[Empty]	
Sebelumnya:	000015783b764259d382017d91a36d206d0600e2ccb3567748f	
Cincang:	000012fa9b916eb9078f8d98a7864e697ae83ed54f5146bd844	
<button>Tambang</button>		

Blok:	#	3
Nonce:	12937	
Data:	[Empty]	
Sebelumnya:	000012fa9b916eb9078f8d98a7864e697ae83ed54f5146bd844	
Cincang:	0000b9015ce2a08b61216ba5a0778545	
<button>Tambang</button>		

Demo Rantai Blok

Cincang Blok **Rantai Blok** Terdistribusi Token-token Coinbase

Rantai Blok

Blok: # 1	Blok: # 2	Blok: # 3
Nonce: 11316	Nonce: 35230	Nonce: 12937
Data: <input type="text"/>	Data: <input type="text"/> shely	Data: <input type="text"/>
Sebelumnya: 00	Sebelumnya: 000015783b764259d382017d91a36d206d0600e2cbb3567748f	Sebelumnya: 39305deea1fd41abd698afee2aa8dc166ddf6f4f4842a965c41
Cincang: 000015783b764259d382017d91a36d206d0600e2cbb3567748f	Cincang: 39305deea1fd41abd698afee2aa8dc166ddf6f4f4842a965c41	Cincang: 4beade654b8577b3c6407b12e42711dd
Tambang	Tambang	Tambang

Demo Rantai Blok

Cincang Blok Rantai Blok **Terdistribusi** Token-token Coinbase

Rantai Blok Terdistribusi

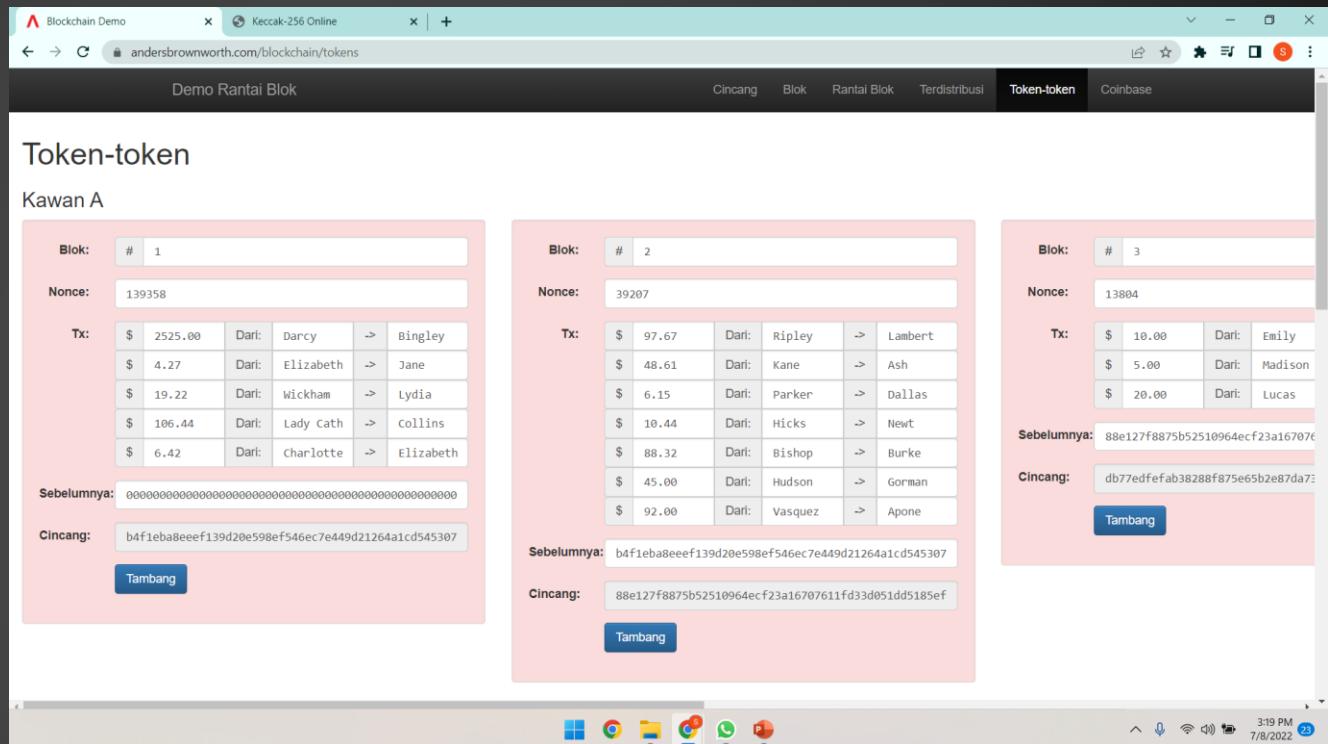
Kawan A

Blok:	# 1
Nonce:	11316
Data:	[Large empty box]
Sebelumnya:	00
Cincang:	000015783b764259d382017d91a36d206d0600e2ccb3567748
<button>Tambang</button>	

Blok:	# 2
Nonce:	35230
Data:	[Large empty box]
Sebelumnya:	000015783b764259d382017d91a36d206d0600e2ccb3567748
Cincang:	000012fa9b916eb9078f8d98a7864e697ae83ed54f5146bd84
<button>Tambang</button>	

Blok:	# 3
Nonce:	12937
Data:	[Large empty box]
Sebelumnya:	000012fa9b916eb9078f8d98a7864e697ae83ed54f5146bd84
Cincang:	0000b9015ce2a08b61216ba5a0778545
<button>Tambang</button>	

```
graph LR; B1["Block 1: #1, Nonce 11316, Cincang 000015783b764259d382017d91a36d206d0600e2ccb3567748"] --> B2["Block 2: #2, Nonce 35230, Cincang 000012fa9b916eb9078f8d98a7864e697ae83ed54f5146bd84"]; B2 --> B3["Block 3: #3, Nonce 12937, Cincang 0000b9015ce2a08b61216ba5a0778545"]
```



Blockchain Demo: Public / Private Keys & Signing

Keys Signatures Transaction Blockchain

Public / Private Key Pairs

Private Key

92853035924865024094188800129399842318596049436997263256614152630016656161875|

Random

Public Key

04508c991f6516fb032830fea7e9ac87761d8a6b96cf662938f1ec4d2840cba0b8ce7fb5b8bf285bd90f156ca8f5e3bdb36c2c23348e

Blockchain Demo: Public / Private Keys & Signing

Keys Signatures Transaction Blockchain

Signatures

Sign Verify

Message

hi world

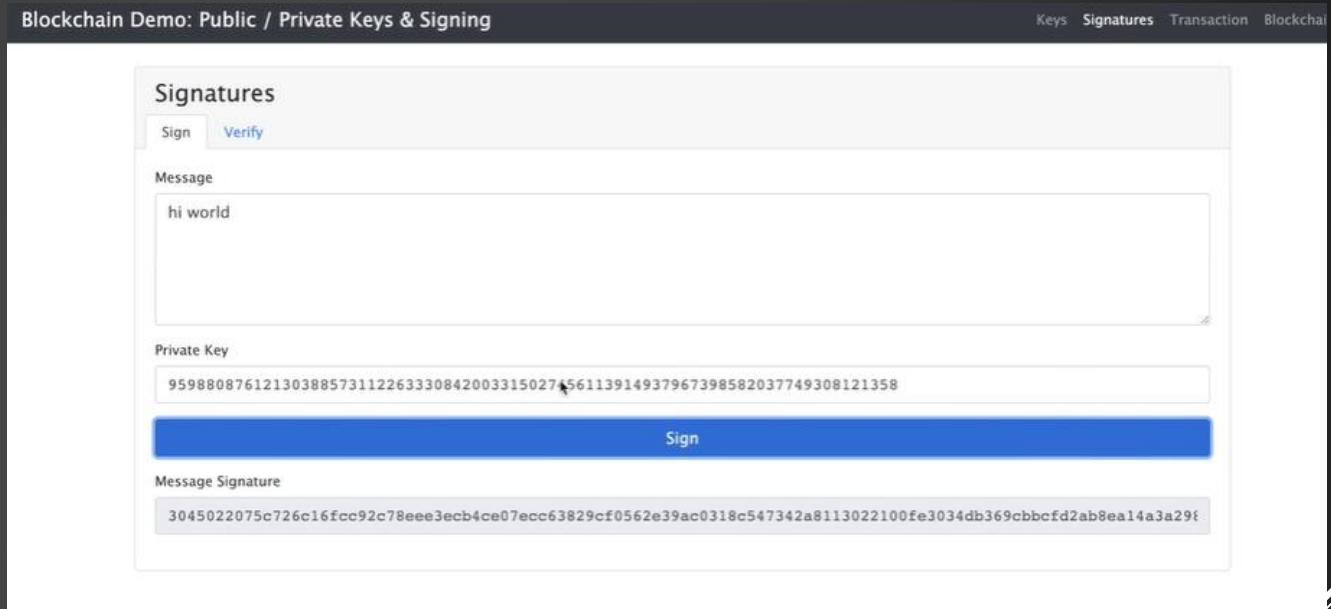
Private Key

9598808761213038857311226333084200331502756113914937967398582037749308121358

Sign

Message Signature

3045022075c726c16fcc92c78eee3ecb4ce07ecc63829cf0562e39ac0318c547342a8113022100fe3034db369cbccfd2ab8ea14a3a29f



Blockchain Demo: Public / Private Keys & Signing

Keys Signatures Transaction Blockchain

Signatures

Sign Verify

Message

hi world

Public Key

04033a93c65c54065907cle04126b6c41222243a1658c81b02e886d113dc175316eb96e119cc64fd429779277b17e80daa2e40a9b1da1

Signature

3045022075c726c16fcc92c78eafa356435245436tedfggs

Verify

The screenshot shows a web-based blockchain demonstration tool. The main title is "Blockchain Demo: Public / Private Keys & Signing". Below it is a navigation bar with links for "Keys", "Signatures" (which is currently selected), "Transaction", and "Blockchain". The main content area is titled "Signatures" and contains two tabs: "Sign" (selected) and "Verify". Under the "Sign" tab, there is a "Message" input field containing the text "hi world". Below it are two large input fields: "Public Key" and "Signature", both containing long hexadecimal strings. At the bottom of the "Sign" section is a large blue "Verify" button.

Transaction

Sign Verify

Message

\$ 400 From: 04033a93c65c54065907c1e04126b -> 04cc955bf8e359cc7ebbb66f4c2dc

Private Key

95988087612130388573112263330842003315027456113914937967398582037749308121358

Sign

Message Signature

30440220504f22140b6f4387fb66f7f8d589ce209b0d458c3f345328c47754bc145eb2d102202f05d5e9e3221f70e88a5a0421b61673c



[Edit](#) Rinkeby Test Network

[Overview](#) [State](#)

② Value: 0.05 Ether (\$0.00)

② Transaction Fee: 0.000046477873281 Ether (\$0.00)

② Gas Price: 0.000000002213232061 Ether

② Gas Limit & Usage by Txn: 21,000 | 21,000 (100%)

② Gas Fees: Base: 0.00000004 Gwei | Max: 0.00000004 Gwei

② Burnt & Txn Savings Fees: 🔥 Burnt: 0.0000000000084 Ether

Advanced gas fee [X](#)

Max base fee (GWEI) [?](#) 1.500000015 = 0 ↴ ↵

Current: 0 GWEI ↴ 12hr: 0 GWEI

Priority Fee (GWEI) [?](#) 1.5 = 0 ↴ ↵

Current: 1 - 2.5 GWEI ↴ 12hr: 1 - 191.21 GWEI

Save these new values as my default for "Advanced"

Gas limit 2000 ↴ ↵

Gas limit must be greater than 20999 and less than 213232061

[Save](#)

Ether (\$0.00)

After constantly using a calculator to convert between Ether units I thought it would be a good idea to implement the [EthereumJS-Units Project](#) on a website for easy accessibility. For clarity reasons there is a [simple](#) and [full](#) converter page, since the majority wouldn't bother for any other units than Ether, Gwei and Wei. This project is using the [BigNumber](#) library and the [EthereumJS-Units Project](#).

Wei 10000000000000000000

Gwei 1000000000 I

Ether

[Overview](#) [State](#)

[This is a Rinkeby **Testnet** transaction only]

② Transaction Hash: [0xfeae5a0a60dedd89dcef269d13ae8bc4abc53ed6f00abb0d6879287da654b695](#) 

② Status:  Success
Number of blocks mined since
[10395208](#)  5 Block Confirmations

② Block: [10395208](#)  5 Block Confirmations

② Timestamp:  1 min ago (Mar-26-2022 06:00:34 PM +UTC)

② From: [0x1066618d0973e44efd2fe5114fd18b64c6420abb](#) 

② To: [0x043202024eb087bd4149fd7f8a1e4a0caa124cfid](#) 

[Overview](#) [State](#)

Advanced A set of information that represents the current **state** is updated when a transaction takes place on the network. The below is a summary of those changes :

Address	Before	After	State [
0x00000000000000000000...	2,402.478198678369995942 Eth	2,402.478245156242436942 Eth	▲ 0.000...
0x043202024eb087bd41...	0 Eth	0.05 Eth	▲ 0.05
0x1066618d0973e44efd...	0.1 Eth	0.049953522126719 Eth	▼ 0.050...

BLOCKCHAIN BASICS

Kelompok

- ▶ Bitcoin adalah salah satu protokol pertama yang menggunakan teknologi revolusioner yang disebut blockchain ini. Buku putih Bitcoin dibuat oleh pseudo anonim Satoshi Nakamoto, dan menguraikan bagaimana Bitcoin dapat melakukan transaksi peer to peer.
- ▶ Pengembang Bitcoin memandang Bitcoin sebagai penyimpan nilai versus Aetherium, pengembang memandang Aetherium sebagai penyimpan nilai dan utilitas untuk memfasilitasi perjanjian terdesentralisasi ini.
- ▶ Sekarang, kontrak pintar di blockchain saja benar-benar luar biasa. Namun, mereka datang dengan masalah besar. Jika kita ingin perjanjian digital ini menggantikan perjanjian dalam kehidupan kita sehari-hari, mereka mungkin akan membutuhkan data dari blockchain dunia nyata sendiri yang sebenarnya tidak dapat berinteraksi dengan dan tidak dapat membaca atau mendengarkan data dari dunia nyata. Inilah yang dikenal sebagai masalah Oracle.

BITCOIN

ORACLE

- ▶ Oracle adalah perangkat apa pun yang mengirimkan data ke blockchain terdesentralisasi ini atau menjalankan komputasi eksternal. Namun, jika kami ingin aplikasi kami tetap benar-benar terdesentralisasi, kami tidak dapat bekerja dengan satu Oracle atau satu penyedia data atau satu sumber yang menjalankan komputasi eksternal ini. Jadi kita membutuhkan jaringan Oracle yang terdesentralisasi. Mirip dengan jaringan blockchain terdesentralisasi kami, Anda menggunakan logika rantai akan terdesentralisasi, tetapi Anda juga perlu data off chain dan komputasi Anda didesentralisasi, menggabungkan ini pada logika terdesentralisasi rantai. Dengan off chain ini, data terdesentralisasi dan komputasi terdesentralisasi menimbulkan sesuatu yang disebut kontrak pintar hibrida. Dan sebagian besar protokol terbesar yang berinteraksi dengan kita saat ini adalah beberapa jenis kontrak pintar hybrid atau berinteraksi dengan kontrak pintar hybrid. Sampai batas tertentu, di sinilah rantai protokol berperan.

- chainlink adalah jaringan Oracle terdesentralisasi yang paling populer dan kuat adalah salah satu yang akan kita fokuskan untuk kursus ini juga. chainlink juga agnostik blockchain. Jadi untuk bekerja pada Aetherium, longsoran salju, Polygon Solana Terra, atau benar-benar blockchain lain di luar sana. Sekarang, sepanjang kursus ini, Anda akan mendengar istilah DAP atau Protokol terdesentralisasi, atau Protokol kontrak pintar, atau aplikasi terdesentralisasi. Dan mereka semua memiliki arti yang sama. Aplikasi terdesentralisasi biasanya merupakan kombinasi dari banyak kontrak pintar.

CHAINLINK

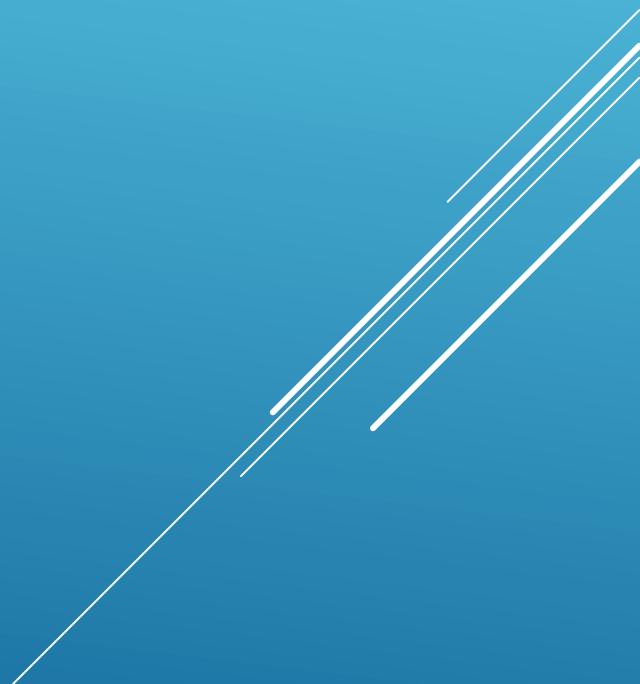
- ▶ Dow adalah kelompok yang diatur sepenuhnya terdesentralisasi oleh serangkaian instruksi atau kontrak pintar pada rantai. Ada beberapa manfaat besar di sini di mana keterlibatan jauh lebih mudah. Aturannya hitam dan putih. Dan Anda dapat melihat segala sesuatu secara langsung pada teknologi pemungutan suara dan tata kelola berantai yang sepenuhnya terdesentralisasi di ruang blockchain adalah salah satu yang besar yang mendorong bagaimana kita dapat mengembangkan politik dan bagaimana kita dapat mengembangkan tata kelola untuk membuatnya lebih efisien, adil, dan masuk akal

DOW

- ▶ Fungsi hashing dan hash akan sama. Tidak peduli di mana Anda melihat blockchain terdesentralisasi, itu akan sama di mana pun Anda melihat, bagaimana itu benar-benar diterapkan, akan sedikit berbeda.
- ▶ Kombinasi dari node-node ini berinteraksi satu sama lain, yang menciptakan seluruh blockchain ini. Apa yang membuat ini begitu kuat juga, adalah bahwa siapa pun dapat bergabung dengan jaringan. Dan itulah mengapa ada desentralisasi penghalang untuk masuk adalah sedikit persyaratan perangkat keras untuk mendapatkan bahan yang benar untuk menjalankan perangkat lunak. Dan kemudian Anda menjalankan perangkat lunak, siapa pun dapat bergabung dengan jaringan ini dan berpartisipasi. Dan itulah yang membuatnya benar-benar terdesentralisasi.

- ▶ Beberapa contoh lapisan dua akan menjadi arbitrase tautan berantai, atau optimisme. Arbitrase dan optimisme sangat menarik karena mereka adalah lapisan dua yang juga ingin menyelesaikan masalah skalabilitas ini. Arbitrase dan optimisme adalah apa yang dikenal sebagai roll up dan mereka menggulung transaksi mereka menjadi lapisan satu seperti Aetherium, kita tidak akan terlalu jauh ke dalam roll up dan bagaimana mereka benar-benar bekerja. Tetapi yang benar-benar perlu Anda ketahui adalah bahwa roll up seperti rantai pecahan, mereka memperoleh keamanannya dari lapisan dasar dari lapisan satu seperti Aetherium.

SELESAI

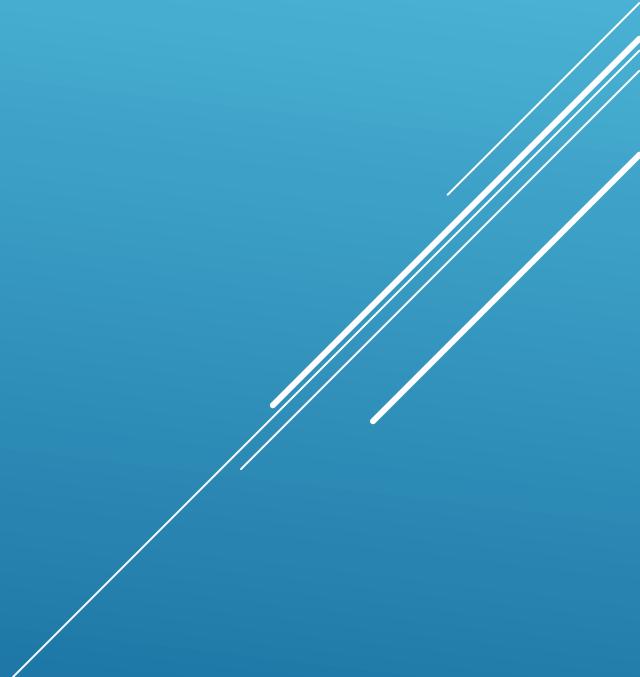




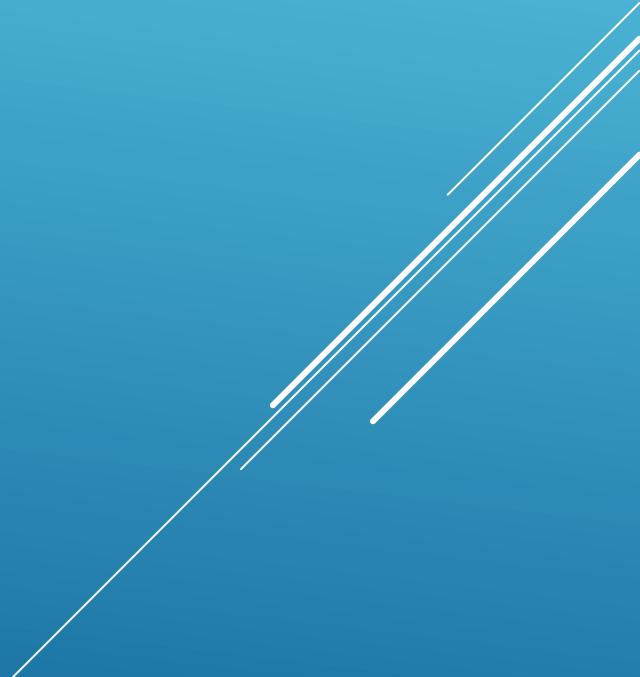
WELCOME TO REMIX! SIMPLE STORAGE

Kelompok

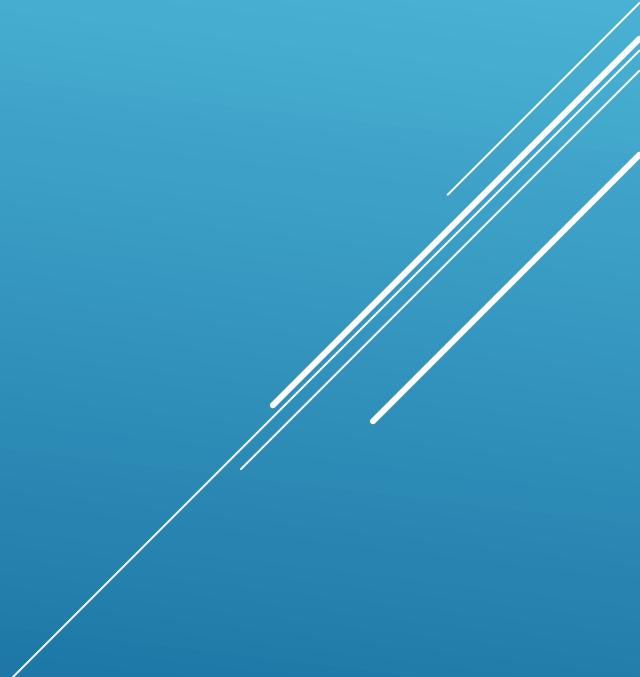
Membangun aplikasi kontrak pintar terdesentralisasi adalah membangun dunia yang lebih bertanggung jawab atas dunia dengan janji-janji yang tidak dapat dilanggar, dunia yang lebih cepat, lebih efisien, bebas secara finansial, komunitas kolaboratif, menggabungkan kecakapan filosofi dan teknologi ke dalam sistem baru



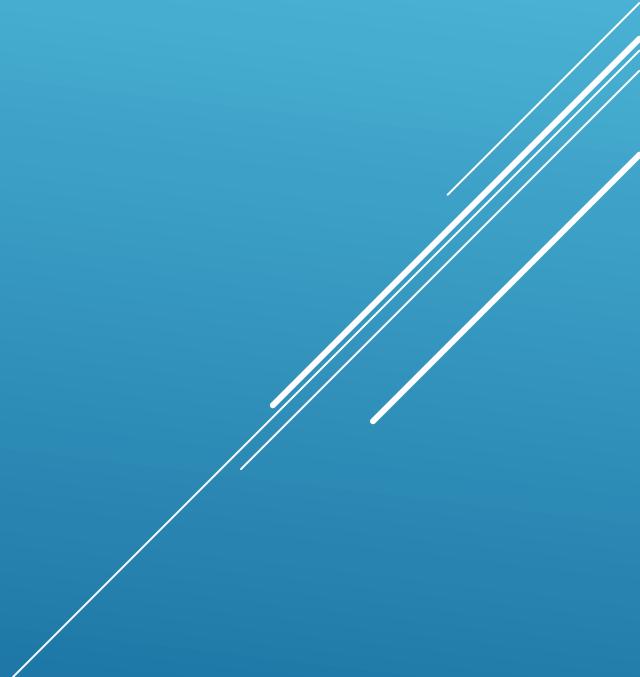
Dunia blockchain dan kontrak pintar sangat kolaboratif. Jadi pastikan untuk menggunakan alat seperti tab diskusi repositori GitHub kami, mengajukan pertanyaan tentang Stack Overflow dan Aetherium Stack Exchange dan menandai teknologi relatif, membuat masalah pada berbagai repositori GitHub yang Anda kerjakan dengan melompat ke perselisihan, Reddit, Twitter, dan tempat lain yang berkumpul oleh komunitas dan teknologi ini.



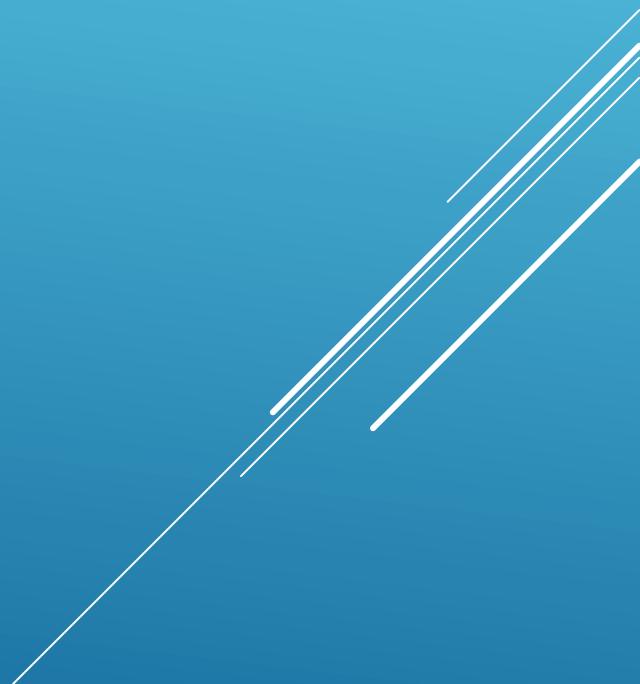
Hackathon adalah salah satu tempat terbaik untuk terhubung dengan insinyur lain, chainlink hackathon eath, hackathon global, dan Dev folio hackathon adalah tiga suite hackathon yang bagus untuk terhubung.



Memahami ideologi ini dan dasar-dasar ini sangat penting, karena ini akan menentukan bagaimana Anda merancang aplikasi terdesentralisasi Anda, mempelajari dasar-dasar blockchain dan soliditas sangat penting.



SELESAI



Blockchain

Lessons 3 Remix Storage Factory

Membuat file pada REMIX

hal yang pertama dilakukan adalah membuka website remix(<https://remix.ethereum.org>) lalu kita membuat workspace baru agar bersih dari template yang default diberikan oleh remix.

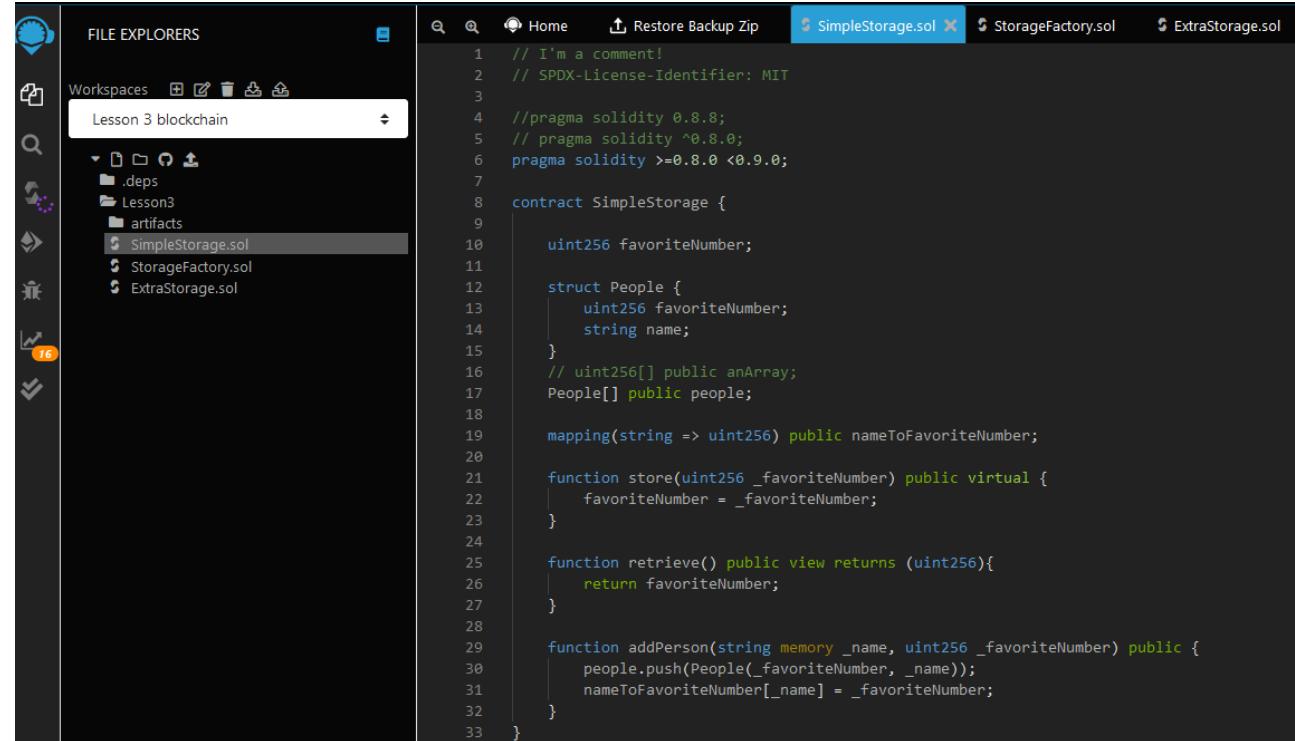
lalu yang kita dapat lakukan adalah membuat tiga file yang berekstensi ".sol" yang di antara lainnya adalah :

SimpleStorage.sol

StorageFactory.sol

ExtraStorage.sol

semua file ini kita akan gunakan seiring waktu.



```
// I'm a comment!
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;
pragma solidity >=0.8.0 <0.9.0;

contract SimpleStorage {
    uint256 favoriteNumber;

    struct People {
        uint256 favoriteNumber;
        string name;
    }
    // uint256[] public anArray;
    People[] public people;

    mapping(string => uint256) public nameToFavoriteNumber;

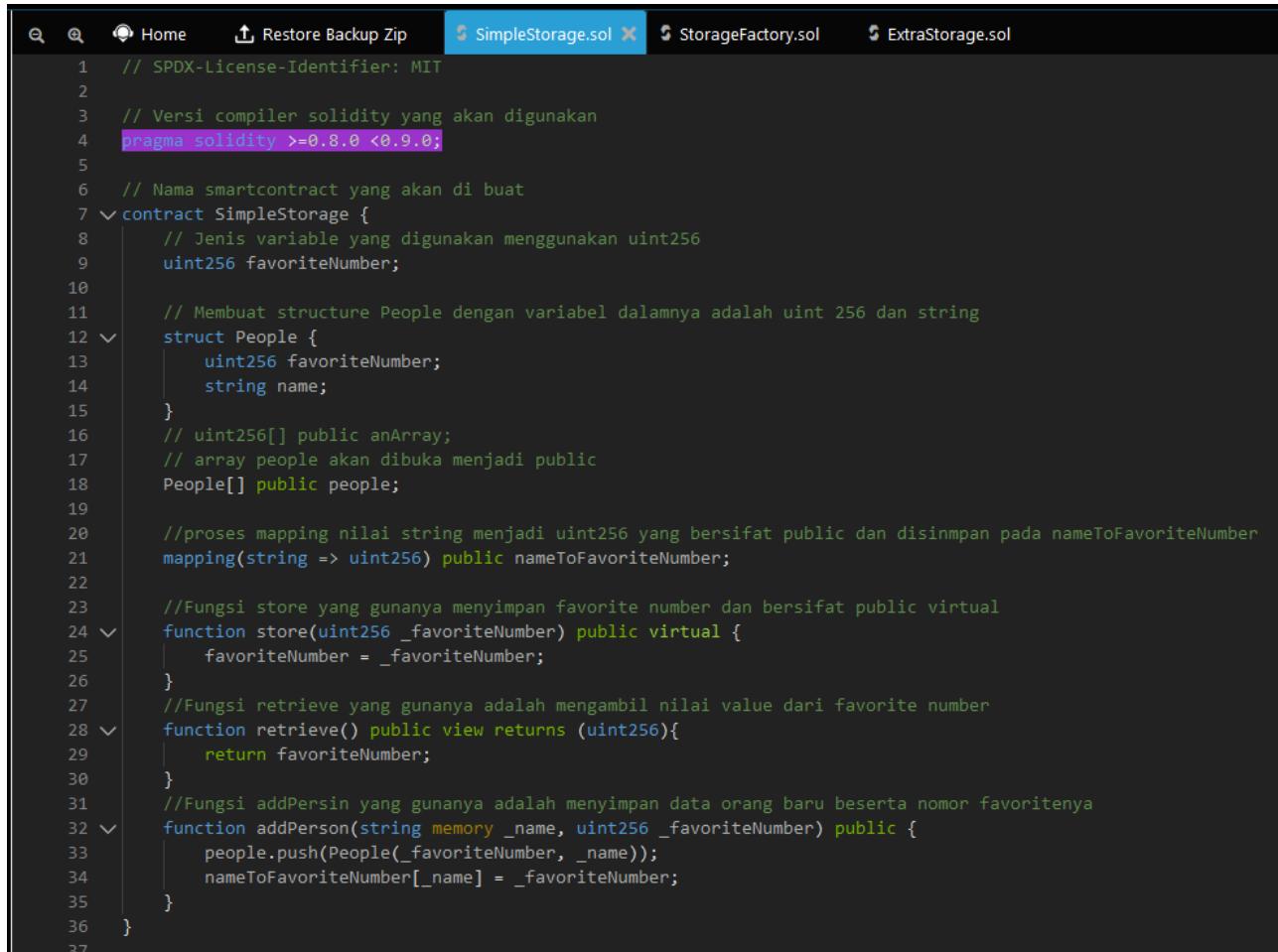
    function store(uint256 _favoriteNumber) public virtual {
        favoriteNumber = _favoriteNumber;
    }

    function retrieve() public view returns (uint256){
        return favoriteNumber;
    }

    function addPerson(string memory _name, uint256 _favoriteNumber) public {
        people.push(People(_favoriteNumber, _name));
        nameToFavoriteNumber[_name] = _favoriteNumber;
    }
}
```

Mempersiapkan Simple Storage

lalu kita akan mempersiapkan code kita pada SimpleStorage. Pada code disamping saya menggunakan compiler solidity lebih dari 0.7.0 hingga kurang dari versi 0.9.0. Lalu kita pun membuat nama kontrak dengan SimpleStorage dengan menyimpan variable uint256, mapping variable, dan fungsi dari smart contract yang kita buat



The screenshot shows a Solidity IDE interface with three tabs at the top: 'Home', 'Restore Backup Zip', and 'SimpleStorage.sol'. The 'SimpleStorage.sol' tab is active, indicated by a blue background and white text. The code editor below contains the Solidity code for the SimpleStorage contract. The code includes comments explaining the use of SPDX license, compiler version, and data types. It defines a struct 'People' with fields 'favoriteNumber' (uint256) and 'name' (string). It also defines a mapping 'nameToFavoriteNumber' (string to uint256). Three functions are implemented: 'store' (public virtual), 'retrieve' (public view), and 'addPerson' (public).

```
// SPDX-License-Identifier: MIT
// Versi compiler solidity yang akan digunakan
pragma solidity >=0.8.0 <0.9.0;

// Nama smartcontract yang akan di buat
contract SimpleStorage {
    // Jenis variable yang digunakan menggunakan uint256
    uint256 favoriteNumber;

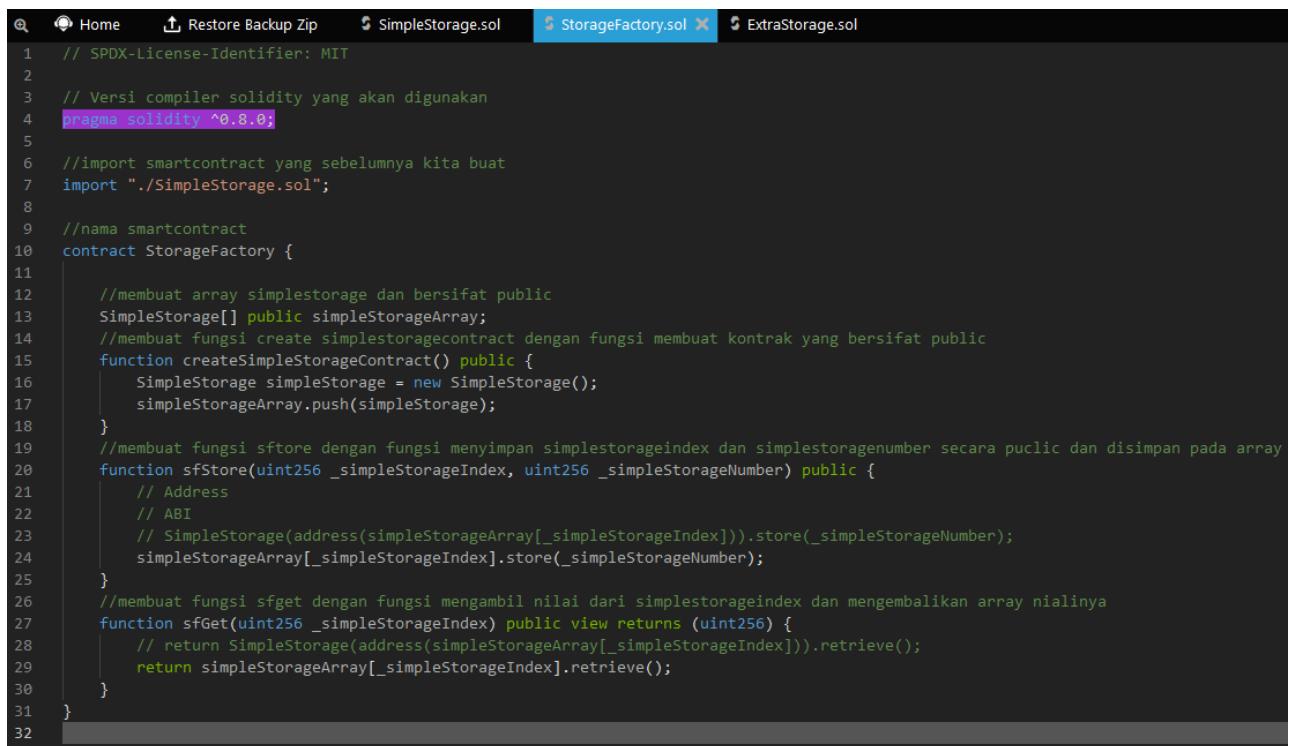
    // Membuat structure People dengan variabel dalamnya adalah uint 256 dan string
    struct People {
        uint256 favoriteNumber;
        string name;
    }
    // uint256[] public anArray;
    // array people akan dibuka menjadi public
    People[] public people;

    //proses mapping nilai string menjadi uint256 yang bersifat public dan disimpan pada nameToFavoriteNumber
    mapping(string => uint256) public nameToFavoriteNumber;

    //Fungsi store yang gunanya menyimpan favorite number dan bersifat public virtual
    function store(uint256 _favoriteNumber) public virtual {
        favoriteNumber = _favoriteNumber;
    }
    //Fungsi retrieve yang gunanya adalah mengambil nilai value dari favorite number
    function retrieve() public view returns (uint256){
        return favoriteNumber;
    }
    //Fungsi addPerson yang gunanya adalah menyimpan data orang baru beserta nomor favoritnya
    function addPerson(string memory _name, uint256 _favoriteNumber) public {
        people.push(People(_favoriteNumber, _name));
        nameToFavoriteNumber[_name] = _favoriteNumber;
    }
}
```

Mempersiapkan Storage Factory

lalu kita akan mempersiapkan code kita pada StorageFactory. Pada code disamping saya mengguna compiler solidity lebih dari lebih dari 0.8.0. Lalu kita pun membuat nama kontrak dengan StorageFactory dengan menyimpan SimpleStorageArray ke SimpleStorage dan bersifat public, selain itu kita membuat fungsi pada kontrak seperti fungsi createsimplestoragecontract yang bersifat public dengan fungsinya adalah membuat SC, lalu fungsi sfstore untuk menyimpan simplestorageindex dan simplestoragenumber secara public pada array dan fungsi sfget untuk mengambil nilai simplestorage index



The screenshot shows a Solidity IDE interface with three tabs at the top: 'SimpleStorage.sol' (selected), 'StorageFactory.sol' (highlighted in blue), and 'ExtraStorage.sol'. The code editor displays the following Solidity code:

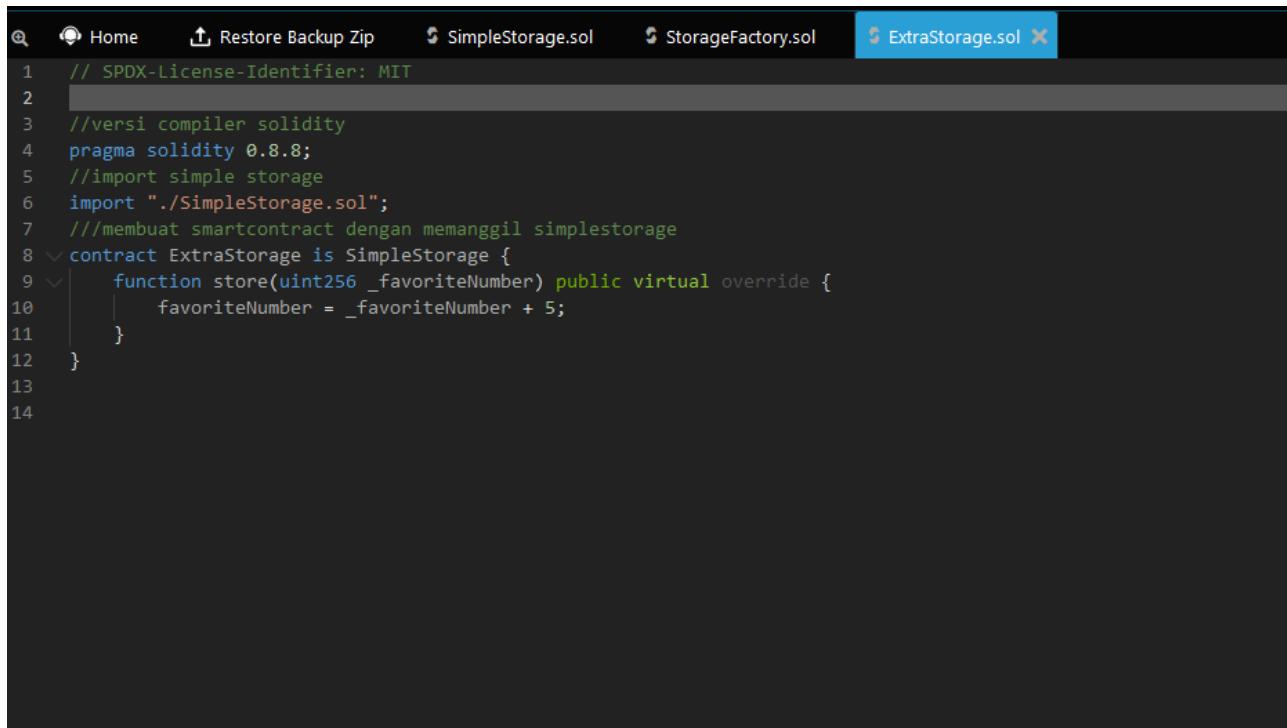
```
// SPDX-License-Identifier: MIT
// Versi compiler solidity yang akan digunakan
pragma solidity ^0.8.0;

//import smartcontract yang sebelumnya kita buat
import "./SimpleStorage.sol";

//nama smartcontract
contract StorageFactory {
    //membuat array simplestorage dan bersifat public
    SimpleStorage[] public simpleStorageArray;
    //membuat fungsi create simplestoragecontract dengan fungsi membuat kontrak yang bersifat public
    function createSimpleStorageContract() public {
        SimpleStorage simpleStorage = new SimpleStorage();
        simpleStorageArray.push(simpleStorage);
    }
    //membuat fungsi sfstore dengan fungsi menyimpan simplestorageindex dan simplestoragenumber secara public dan disimpan pada array
    function sfStore(uint256 _simpleStorageIndex, uint256 _simpleStorageNumber) public {
        // Address
        // ABI
        // SimpleStorage(address(simpleStorageArray[_simpleStorageIndex])).store(_simpleStorageNumber);
        simpleStorageArray[_simpleStorageIndex].store(_simpleStorageNumber);
    }
    //membuat fungsi sfget dengan fungsi mengambil nilai dari simplestorageindex dan mengembalikan array nialinya
    function sfGet(uint256 _simpleStorageIndex) public view returns (uint256) {
        // return SimpleStorage(address(simpleStorageArray[_simpleStorageIndex])).retrieve();
        return simpleStorageArray[_simpleStorageIndex].retrieve();
    }
}
```

Mempersiapkan Extra Storage

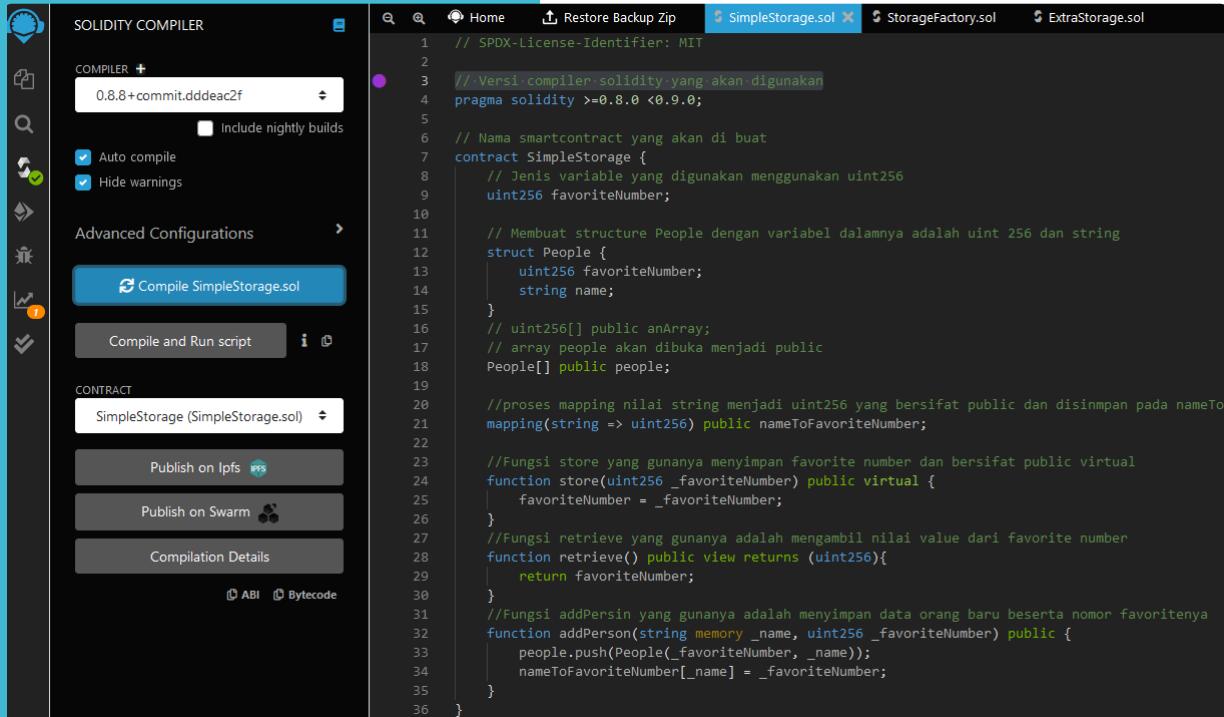
lalu kita akan mempersiapkan code kita pada ExtraStorage. Pada code disamping saya gunakan compiler solidity lebih dari lebih dari 0.8.8. Lalu kita pun membuat nama kontrak dengan ExtraStorage dengan memanggil SimpleStorage terlebih dahulu. lalu pada smart contract nya sendiri akan mengoverride nilai favoritenumber dengan menambahkan lima pada nilai favoritenumber sebelumnya



The screenshot shows a code editor interface with a dark theme. At the top, there is a navigation bar with icons for Home, Restore Backup Zip, and three tabs: SimpleStorage.sol, StorageFactory.sol, and ExtraStorage.sol (which is currently active). The main area displays the following Solidity code:

```
// SPDX-License-Identifier: MIT
pragma solidity 0.8.8;
//import simple storage
import "./SimpleStorage.sol";
//membuat smartcontract dengan memanggil simplestorage
contract ExtraStorage is SimpleStorage {
    function store(uint256 _favoriteNumber) public virtual override {
        favoriteNumber = _favoriteNumber + 5;
    }
}
```

Mengcompile Simple Storage



The image shows the Solidity Compiler interface. On the left, there's a sidebar with icons for Home, Restore Backup Zip, SOLIDITY COMPILER, COMPILER + (version 0.8.8+commit.dddeac2f), Auto compile, Hide warnings, Advanced Configurations, Compile SimpleStorage.sol (which is highlighted in blue), Compile and Run script, CONTRACT (SimpleStorage (SimpleStorage.sol)), Publish on Ipfs, Publish on Swarm, Compilation Details, ABI, and Bytecode. The main area displays the Solidity code for SimpleStorage.sol:

```
// SPDX-License-Identifier: MIT
pragma solidity >=0.8.0 <0.9.0;

// Nama smartcontract yang akan di buat
contract SimpleStorage {
    // Jenis variable yang digunakan menggunakan uint256
    uint256 favoriteNumber;

    // Membuat structure People dengan variabel dalamnya adalah uint 256 dan string
    struct People {
        uint256 favoriteNumber;
        string name;
    }
    // uint256[] public anArray;
    // array people akan dibuka menjadi public
    People[] public people;

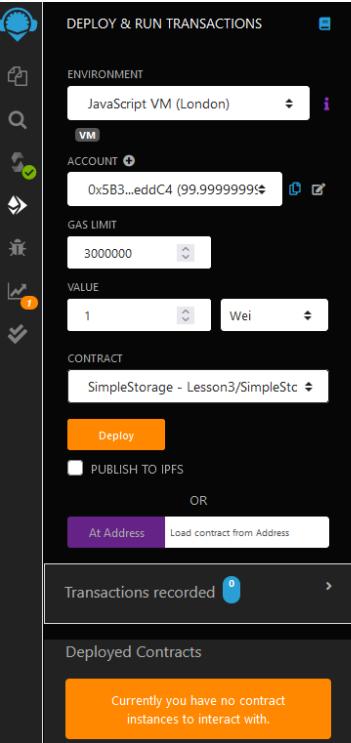
    //proses mapping nilai string menjadi uint256 yang bersifat public dan disimpan pada nameToFavoriteNumber
    mapping(string => uint256) public nameToFavoriteNumber;

    //Fungsi store yang gunanya menyimpan favorite number dan bersifat public virtual
    function store(uint256 _favoriteNumber) public virtual {
        favoriteNumber = _favoriteNumber;
    }
    //Fungsi retrieve yang gunanya adalah mengambil nilai value dari favorite number
    function retrieve() public view returns (uint256){
        return favoriteNumber;
    }
    //Fungsi addPerson yang gunanya adalah menyimpan data orang baru beserta nomor favoritnya
    function addPerson(string memory _name, uint256 _favoriteNumber) public {
        people.push(People(_favoriteNumber, _name));
        nameToFavoriteNumber[_name] = _favoriteNumber;
    }
}
```

pada tahapan ini kita akan meng compile terlebih dahulu code kita yang telah dibuat sebelumnya

Mendeploy Simple Storage

pada tahapan ini kita akan mendeploy smart contract kita ke dalam jaringan blockchain tetapi hanya masih pada test network saja. dan pada environment kita dapat menggunakan yang tersedia pada remix, tetapi pada test ini saya menggunakan environment london



```
// SPDX-License-Identifier: MIT
// Versi compiler-solidity yang akan digunakan
pragma solidity >=0.8.0 <0.9.0;

// Nama smartcontract yang akan di buat
contract SimpleStorage {
    // Jenis variable yang digunakan menggunakan uint256
    uint256 favoriteNumber;
}

// Membuat structure People dengan variabel dalamnya adalah uint 256 dan string
struct People {
    uint256 favoriteNumber;
    string name;
}
// uint256[] public anArray;
// array people akan dibuka menjadi public
People[] public people;

//proses mapping nilai string menjadi uint256 yang bersifat public dan disimpan pada nameToFavoriteNumber
mapping(string => uint256) public nameToFavoriteNumber;

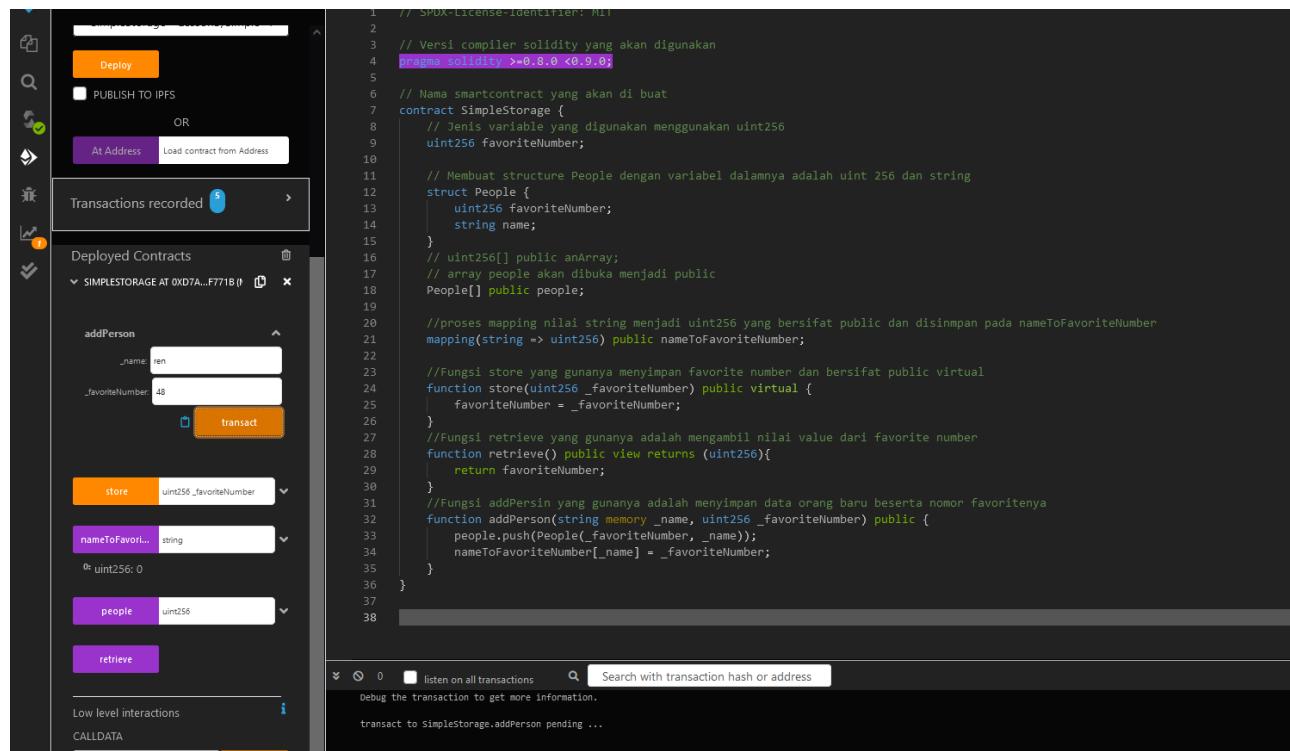
//Fungsi store yang gunanya menyimpan favorite number dan bersifat public virtual
function store(uint256 _favoriteNumber) public virtual {
    favoriteNumber = _favoriteNumber;
}

//Fungsi retrieve yang gunanya adalah mengambil nilai value dari favorite number
function retrieve() public view returns (uint256){
    return favoriteNumber;
}

//Fungsi addPerson yang gunanya adalah menyimpan data orang baru beserta nomor favoritnya
function addPerson(string memory _name, uint256 _favoriteNumber) public {
    people.push(People(_favoriteNumber, _name));
    nameToFavoriteNumber[_name] = _favoriteNumber;
}
```

Explore Simple Storage SC

pertama-tama kita akan mencoba menyimpan value nama orang ke dalam blockchain melalui smart contract kita

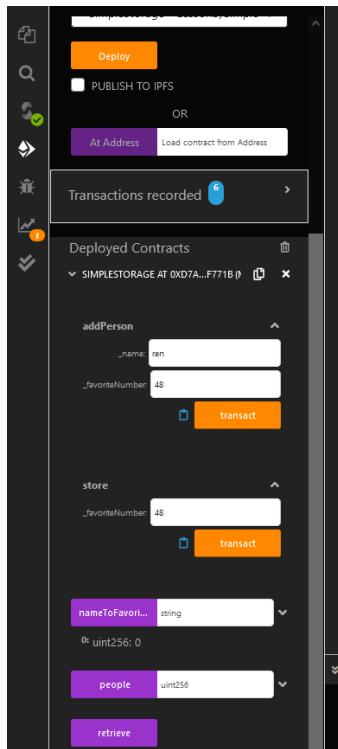


The screenshot shows the Truffle UI interface. On the left, there's a sidebar with icons for file, search, and deployment. The main area has tabs for 'Deploy' (highlighted in orange), 'PUBLISH TO IPFS', 'At Address' (selected), and 'Load contract from Address'. Below these are sections for 'Transactions recorded' and 'Deployed Contracts'. Under 'Deployed Contracts', it shows 'SIMPLESTORAGE AT 0xD7A...F771B()' with an 'addPerson' button. The 'addPerson' section has fields for '_name' (set to 'ren') and '_favoriteNumber' (set to '48'), with a 'transact' button. Below this are buttons for 'store' (with parameter 'uint256 _favoriteNumber'), 'nameToFavorit...', and 'retrieve'. At the bottom, there are sections for 'Low level interactions' and 'CALLDATA'. On the right side of the interface, the Solidity code for the 'SimpleStorage' contract is displayed:

```
1 // SPDX-License-Identifier: MIT
2
3 // Versi compiler solidity yang akan digunakan
4 // solidity >=0.8.0 <0.9.0;
5
6 // Nama smartcontract yang akan dibuat
7 contract SimpleStorage {
8     // Jenis variable yang digunakan menggunakan uint256
9     uint256 favoriteNumber;
10
11    // Membuat structure People dengan variabel dalamnya adalah uint 256 dan string
12    struct People {
13        uint256 favoriteNumber;
14        string name;
15    }
16    // uint256[] public anArray;
17    // array people akan dibuka menjadi public
18    People[] public people;
19
20    //proses mapping nilai string menjadi uint256 yang bersifat public dan disimpan pada nameToFavoriteNumber
21    mapping(string => uint256) public nameToFavoriteNumber;
22
23    //Fungsi store yang gunanya menyimpan favorite number dan bersifat public virtual
24    function store(uint256 _favoriteNumber) public virtual {
25        favoriteNumber = _favoriteNumber;
26    }
27    //Fungsi retrieve yang gunanya adalah mengambil nilai value dari favorite number
28    function retrieve() public view returns (uint256){
29        return favoriteNumber;
29    }
30    //Fungsi addPerson yang gunanya adalah menyimpan data orang baru beserta nomor favoritnya
31    function addPerson(string memory _name, uint256 _favoriteNumber) public {
32        people.push(People(_favoriteNumber, _name));
33        nameToFavoriteNumber[_name] = _favoriteNumber;
34    }
35}
36
37
38
```

Explore Simple Storage SC

lalu kita akan mencoba menyimpan value nomor favorite orang tadi ke blockchain melalui smart contract



```
// Versi compiler solidity yang akan digunakan
pragma solidity >=0.8.0 <0.9.0;

// Nama smartcontract yang akan di buat
contract SimpleStorage {
    // Jenis variable yang digunakan menggunakan uint256
    uint256 favoriteNumber;

    // Membuat structure People dengan variabel dalamnya adalah uint 256 dan string
    struct People {
        uint256 favoriteNumber;
        string name;
    }

    // uint256[] public anArray;
    // array people akan dibuka menjadi public
    People[] public people;

    //proses mapping nilai string menjadi uint256 yang bersifat public dan disimpan pada nameToFavoriteNumber
    mapping(string => uint256) public nameToFavoriteNumber;

    //Fungsi store yang gunanya menyimpan favorite number dan bersifat public virtual
    function store(uint256 _favoriteNumber) public virtual {
        favoriteNumber = _favoriteNumber;
    }

    //Fungsi retrieve yang gunanya adalah mengambil nilai value dari favorite number
    function retrieve() public view returns (uint256){
        return favoriteNumber;
    }

    //Fungsi addPerson yang gunanya adalah menyimpan data orang baru beserta nomor favoritennya
    function addPerson(string memory _name, uint256 _favoriteNumber) public {
        people.push(People(_favoriteNumber, _name));
        nameToFavoriteNumber[_name] = _favoriteNumber;
    }
}

//listen on all transactions
[vm] From: 0x5B3...eddC4 to: SimpleStorage.addPerson(string,uint256) 0x07A...F771B value: 0 wei data: 0x6f7...00000 logs: 0 hash: 0xf52...56a63
```

The screenshot shows the Truffle UI interface. At the top, there are options to 'Deploy' or 'PUBLISH TO IPFS'. Below that, a section titled 'Transactions recorded' shows a single transaction with a blue checkmark and a link. The main area is titled 'Deployed Contracts' and shows a single contract named 'SIMPLESTORAGE AT 0xD7A...F771B'. Under this contract, there are two functions: 'addPerson' and 'store'. The 'addPerson' function has fields for '_name' (set to 'ren') and '_favoriteNumber' (set to '48'). The 'store' function also has a '_favoriteNumber' field set to '48'. At the bottom, there are buttons for 'retrieve' and 'people' (which is a dropdown menu). On the right side of the interface, the Solidity code for the 'SimpleStorage' contract is displayed, showing the implementation of the 'addPerson' and 'store' functions along with their mappings and structures.

Explore Simple Storage SC

lalu kita coba memanggil hasil return jika kita menginputkan nama orang tadi ke dalam blockchain, apakah nilai yang dibalikin sama seperti nilai yang kita inputkan? jika benar berarti smart contract yang telah dibuat sebelumnya sudah benar

```
// SPDX-License-Identifier: MIT
pragma solidity >=0.8.0 <0.9.0;

// Nama smartcontract yang akan di buat
contract SimpleStorage {
    // Jenis variable yang digunakan menggunakan uint256
    uint256 favoriteNumber;

    // Membuat structure People dengan variabel dalamnya adalah uint 256 dan string
    struct People {
        uint256 favoriteNumber;
        string name;
    }

    // uint256[] public anArray;
    // array people akan dibuka menjadi public
    People[] public people;

    //proses mapping nilai string menjadi uint256 yang bersifat public dan disimpan pada nameToFavoriteNumber
    mapping(string => uint256) public nameToFavoriteNumber;

    //Fungsi store yang gunanya menyimpan favorite number dan bersifat public virtual
    function store(uint256 _favoriteNumber) public virtual {
        favoriteNumber = _favoriteNumber;
    }

    //Fungsi retrieve yang gunanya adalah mengambil nilai value dari favorite number
    function retrieve() public view returns (uint256){
        return favoriteNumber;
    }

    //Fungsi addPerson yang gunanya adalah menyimpan data orang baru beserta nomor favoritnya
    function addPerson(string memory _name, uint256 _favoriteNumber) public {
        people.push(People(_favoriteNumber, _name));
        nameToFavoriteNumber[_name] = _favoriteNumber;
    }
}
```

Explore Simple Storage SC

lalu kita coba memanggil hasil return jika kita menginputkan nomor favorite orang, jika outputnya adalah yang kita inputkan sebelumnya maka code kita sudah benar

The screenshot shows the Truffle UI interface with the SimpleStorage smart contract. On the left, there are four interaction panels: 'addPerson' (with inputs '_name: ren' and '_favoriteNumber: 48'), 'store' (with input '_favoriteNumber: "48"'), 'nameToFavoriteNumber' (with input '_name: ren'), and 'people' (with input '_name: "48"'). On the right, the Solidity code for the SimpleStorage contract is displayed:

```
3 // Versi compiler solidity yang akan digunakan
4 pragma solidity >=0.8.0 <0.9.0;
5
6 // Nama smartcontract yang akan di buat
7 contract SimpleStorage {
8     // Jenis variabel yang digunakan menggunakan uint256
9     uint256 favoriteNumber;
10
11    // Membuat structure People dengan variabel dalamnya adalah uint 256 dan string
12    struct People {
13        uint256 favoriteNumber;
14        string name;
15    }
16    // uint256[] public anArray;
17    // array people akan dibuka menjadi public
18    People[] public people;
19
20    //proses mapping nilai string menjadi uint256 yang bersifat public dan disimpan pada nameToFavoriteNumber
21    mapping(string => uint256) public nameToFavoriteNumber;
22
23    //Fungsi store yang gunanya menyimpan favorite number dan bersifat public virtual
24    function store(uint256 _favoriteNumber) public virtual {
25        favoriteNumber = _favoriteNumber;
26    }
27    //Fungsi retrieve yang gunanya adalah mengambil nilai value dari favorite number
28    function retrieve() public view returns (uint256){
29        return favoriteNumber;
30    }
31    //Fungsi addPerson yang gunanya adalah menyimpan data orang baru beserta nomor favoritenya
32    function addPerson(string memory _name, uint256 _favoriteNumber) public {
33        people.push(People(_favoriteNumber, _name));
34        nameToFavoriteNumber[_name] = _favoriteNumber;
35    }
36}
37
38
```

At the bottom, there is a search bar with the placeholder 'Search with transaction hash or address' and a note 'Debug the transaction to get more information.'

Mendeploy Storage Factory

jika berhasil kita akan melihat smart contract yang kita telah buat pada tab deployed contracts. pada deployed contracts kita pun dapat melihat fitur-fitur yang kita buat pada smart contract

```
// Versi compiler solidity yang akan digunakan
pragma solidity ^0.8.0;

//import smartcontract yang sebelumnya kita buat
import "./SimpleStorage.sol";

//nama smartcontract
contract Storagefactory {
    //membuat array simplestorage dan bersifat public
    SimpleStorage[] public simpleStorageArray;
    //membuat fungsi create simplestoragecontract dengan fungsi membuat kontrak yang bersifat public
    function createSimpleStorageContract() public {
        SimpleStorage simpleStorage = new SimpleStorage();
        simpleStorageArray.push(simpleStorage);
    }
    //membuat fungsi sfstore dengan fungsi menyimpan simplestorageindex dan simplestoragenumber secara public dan disimpan
    function sfStore(uint256 _simpleStorageIndex, uint256 _simpleStorageNumber) public {
        // Address
        // ABI
        SimpleStorage(address(simpleStorageArray[_simpleStorageIndex])).store(_simpleStorageNumber);
        simpleStorageArray[_simpleStorageIndex].store(_simpleStorageNumber);
    }
    //membuat fungsi sfget dengan fungsi mengambil nilai dari simplestorageindex dan mengembalikan array nialinya
    function sfGet(uint256 _simpleStorageIndex) public view returns (uint256) {
        // return SimpleStorage(address(simpleStorageArray[_simpleStorageIndex])).retrieve();
        return simpleStorageArray[_simpleStorageIndex].retrieve();
    }
}
```

Explore Storage Factory

pada smart contract storage factory, kita membuat kode yang dimana kita dapat membuat SC lalu menyimpan kan nilai ke blockchain. pada langkah pertama ini kita mencoba untuk membuat SC dengan index yang dimulai dengan nol

The screenshot shows a web-based Ethereum development interface. On the left, there's a sidebar with icons for file operations, search, and deployment. The main area has tabs for "DEPLOY & RUN TRANSACTIONS" and "CONTRACTS". Under "DEPLOY & RUN TRANSACTIONS", there are fields for "VALUE" (set to 0 Wei), "CONTRACT" (set to "StorageFactory - Lesson3/Storage"), and a "Deploy" button. Below this, there are options for "PUBLISH TO IPFS" and "At Address". A "Transactions recorded" section shows 25 transactions. Under "CONTRACTS", it lists "SIMPLESTORAGE AT 0x0FC...9A836" and "STORAGEFACTORY AT 0x5A8...C4D01". The "STORAGEFACTORY" entry has a "createSimpleS..." button and two dropdown menus: "sfStore" (set to "uint256 _simpleStorageIndex") and "sfGet" (set to "0"). Below these are fields for "simpleStorage..." (set to "0") and "address" (set to "0xbdb5b354220B250DF257ed5e988Fe81Cd6235"). At the bottom, there are filters for "listen on all transactions" and a search bar, along with a log of recent transactions.

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

//import smartcontract yang sebelumnya kita buat
import "./SimpleStorage.sol";

//nama smartcontract
contract StorageFactory {
    //membuat array simplestorage dan bersifat public
    SimpleStorage[] public simpleStorageArray;
    //membuat fungsi create simplestoragecontract dengan fungsi membuat kontrak yang bersifat public
    function createSimpleStorageContract() public {
        SimpleStorage simpleStorage = new SimpleStorage();
        simpleStorageArray.push(simpleStorage);
    }
    //membuat fungsi sfstore dengan fungsi menyimpan simplestorageindex dan simplestoragenumber secara public dan disimpan pada array
    function sfStore(uint256 _simpleStorageIndex, uint256 _simpleStorageNumber) public {
        // Address
        // ABI
        // SimpleStorage(address(simpleStorageArray[_simpleStorageIndex])).store(_simpleStorageNumber);
        simpleStorageArray[_simpleStorageIndex].store(_simpleStorageNumber);
    }
    //membuat fungsi sfget dengan fungsi mengambil nilai dari simplestorageindex dan mengembalikan array nialinya
    function sfGet(uint256 _simpleStorageIndex) public view returns (uint256) {
        // return SimpleStorage(address(simpleStorageArray[_simpleStorageIndex])).retrieve();
        return simpleStorageArray[_simpleStorageIndex].retrieve();
    }
}
```

Explore Storage Factory

lalu kita dapat menyimpan nilai nomor favorite kita dan disimpan pada index SC tertentu pada blockchain. disini saya menggunakan index nol dan menyimpan nomor favorit delapan belas. jika kita memanggil fungsi retrieve seharusnya kita mendapatkan nilai delapan belas tadi yang kita inputkan sebelumnya.

The screenshot shows the Truffle UI interface. On the left, the 'Deploy' screen is visible, showing a 'Value' input field set to 3000000 Wei, a 'Contract' dropdown set to 'StorageFactory - Lesson3/StorageFactory', and a 'Deploy' button. Below this, the 'Transactions recorded' section shows 26 transactions. To the right, the 'Deployed Contracts' section lists 'SIMPLESTORAGE AT 0x0FC...9A836' and 'STORAGEFACTORY AT 0x5A8...C4d01'. Under 'STORAGEFACTORY', there are buttons for 'createSimpleS...', 'sfStore', and 'sfGet'. The 'sfGet' button has a value of 0. Below these buttons, it shows the address 0xbd5b35422082500f257ed5e988fe8f81cd86235. At the bottom, there is a 'Low level interactions' section. On the right side of the interface, the Solidity code for the StorageFactory contract is displayed:

```
2
3 // Versi compiler solidity yang akan digunakan
4 //solc solidity ^0.8.0;
5
6 //import smartcontract yang sebelumnya kita buat
7 import "./SimpleStorage.sol";
8
9 //nama smartcontract
10 contract StorageFactory {
11
12     //membuat array simplestorage dan bersifat public
13     SimpleStorage[] public simpleStorageArray;
14     //membuat fungsi create simplestoragecontract dengan fungsi membuat kontrak yang bersifat public
15     function createSimpleStorageContract() public {
16         SimpleStorage simpleStorage = new SimpleStorage();
17         simpleStorageArray.push(simpleStorage);
18     }
19     //membuat fungsi sfstore dengan fungsi menyimpan simplestorageindex dan simplestoragenumber secara public dan disimpan pada array
20     function sfStore(uint256 _simpleStorageIndex, uint256 _simpleStorageNumber) public {
21         // Address
22         // ABI
23         // SimpleStorage(address(simpleStorageArray[_simpleStorageIndex])).store(_simpleStorageNumber);
24         simpleStorageArray[_simpleStorageIndex].store(_simpleStorageNumber);
25     }
26     //membuat fungsi sfget dengan fungsi mengambil nilai dari simplestorageindex dan mengembalikan array nialinya
27     function sfGet(uint256 _simpleStorageIndex) public view returns (uint256) {
28         // return SimpleStorage(address(simpleStorageArray[_simpleStorageIndex])).retrieve();
29         return simpleStorageArray[_simpleStorageIndex].retrieve();
30     }
31 }
```

Below the code, the transaction history shows two entries:

- [vm] from: 0x5B3...eddC4 to: StorageFactory.sfStore(uint256,uint256) 0x5A8...C4d01 value: 0 wei data: 0x156...00012 logs: 0 hash: 0xe9c...02c88 call to StorageFactory.sfGet
- [call] from: 0x5B30Da6a701c568545dCfcB03FcB875f56beddC4 to: StorageFactory.sfGet(uint256) data: 0xc5f...00900

Mendeploy Extra Storage

jika berhasil kita akan melihat smart contract yang kita telah buat pada tab deployed contracts. pada deployed contracts kita pun dapat melihat fitur-fitur yang kita buat pada smart contract

```
// Versi compiler solidity yang akan digunakan
pragma solidity ^0.8.0;

//import smartcontract yang sebelumnya kita buat
import "./SimpleStorage.sol";

//nama smartcontract
contract StorageFactory {
    //membuat array simplestorage dan bersifat public
    SimpleStorage[] public simpleStorageArray;
    //membuat fungsi create simplestoragecontract dengan fungsi membuat kontrak yang bersifat public
    function createSimpleStorageContract() public {
        SimpleStorage simpleStorage = new SimpleStorage();
        simpleStorageArray.push(simpleStorage);
    }
    //membuat fungsi sfstore dengan fungsi menyimpan simplestorageindex dan simplestoragenumber secara puclic dan disimpan
    function sfStore(uint256 _simpleStorageIndex, uint256 _simpleStorageNumber) public {
        // Address
        // ABI
        // SimpleStorage(address(simpleStorageArray[_simpleStorageIndex])).store(_simpleStorageNumber);
        simpleStorageArray[_simpleStorageIndex].store(_simpleStorageNumber);
    }
    //membuat fungsi sfget dengan fungsi mengambil nilai dari simplestorageindex dan mengembalikan array nialinya
    function sfGet(uint256 _simpleStorageIndex) public view returns (uint256) {
        // return SimpleStorage(address(simpleStorageArray[_simpleStorageIndex])).retrieve();
        return simpleStorageArray[_simpleStorageIndex].retrieve();
    }
}
```

Explore Extra Storage

pada smart contract extra storage kita memiliki fungsi override nilai yang kita miliki, pada SC ini kita menambahkan nilai favorite number ditambah lima dari nilai awal, dapat dilihat ketika saya menyimpan value 69 pada nama shel ketika di get masih bernilai 69.

The screenshot shows the Truffle UI interface. On the left, there's a sidebar with icons for file, search, deployment, and network. The main area has tabs for 'CONTRACT' and 'DEPLOYMENT'. In the 'CONTRACT' tab, the code for the ExtraStorage contract is displayed:

```
2 //versi compiler solidity
3 pragma solidity 0.8.8;
4 //import simple storage
5 import "./SimpleStorage.sol";
6 //membuat smartcontract dengan memanggil simplestorage
7 contract ExtraStorage is SimpleStorage {
8     function store(uint256 _favoriteNumber) public virtual override {
9         favoriteNumber = _favoriteNumber + 5;
10    }
11 }
12
13
14
```

Below the code, there are sections for 'Transactions recorded' (with 32 pending transactions), 'Deployed Contracts' (listing SIMPLESTORAGE, STORAGEFACTORY, and EXTRASTORAGE contracts), and a detailed view of the EXTRASTORAGE contract with its methods: addPerson, store, nameToFavoriteNumber, people, and retrieve. The 'store' method is shown with a value of 69. The 'nameToFavoriteNumber' method is shown with a value of 0x0: uint256: 69. The 'retrieve' method is shown with a value of 0x0: uint256.

On the right, the terminal window shows the transaction details:

```
[vm] from: 0x5B3...eddC4 to: SimpleStorage.store(uint256) 0x38c...24C73 value: 0 wei data: 0x605...00045 logs: 0 hash: 0x6b8...eafa4
call to ExtraStorage.nameToFavoriteNumber
```

Explore Extra Storage

tetapi jika retrieve nilai dari SC extra storage kita akan mendapatkan nilai baru yaitu 74 yang berasal $69 + 5$

The screenshot shows the Truffle UI interface. On the left, the 'DEPLOY & RUN TRANSACTIONS' panel displays the 'ExtraStorage - Lesson3/ExtraSto' contract with a 'Deploy' button. Below it, the 'Transactions recorded' section shows 32 transactions. The 'Deployed Contracts' section lists three contracts: SIMPLESTORAGE at 0x0fc...94836, STORAGEFACTORY at 0x5a8...c4d01, and EXTRASTORAGE at 0x38c...24c73. The EXTRASTORAGE contract has several interactions visible: 'addPerson' with value 'ren, 69', 'store' with value '69', 'nameToFavor...' with value 'ren', and 'retrieve' with value '74'. On the right, the code editor shows the Solidity source code for 'ExtraStorage.sol':

```
// SPDX-License-Identifier: MIT
pragma solidity 0.8.8;
import simple_storage;
contract ExtraStorage is SimpleStorage {
    function store(uint256 _favoriteNumber) public virtual override {
        favoriteNumber = _favoriteNumber + 5;
    }
}
```

Below the code editor, the transaction history shows two calls to the 'retrieve' function:

- CALL [call] from: 0x58380a6a701c568545dCfcB03FcB875f56beddC4 to: SimpleStorage.nameToFavoriteNumber(string) data: 0x8ba...00000
- CALL [call] from: 0x58380a6a701c568545dCfcB03FcB875f56beddC4 to: SimpleStorage.retrieve() data: 0x2e6...4cec1

Course 4: FundMe Contract

1. FundMe.sol

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.8;

error NotOwner();

contract FundMe {

    uint256 public constant MINIMUM_USD = 50 * 10 ** 18;

    function fund() public payable {
        require(msg.value.getConversionRate() >= MINIMUM_USD, "You need to
spend more ETH!");
}
```

Pada tahap pertama ini kita akan menambahkan contract di atas ini. Contract ini berfungsi mengirim fund/dana ke suatu akun, MINIMUM_USD adalah jumlah minimum yang harus di kirim oleh contract owner.

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.8;

contract FundMe {

    address public /* immutable */ i_owner;
    uint256 public constant MINIMUM_USD = 50 * 10 ** 18;

    function fund() public payable {
        require(msg.value.getConversionRate() >= MINIMUM_USD, "You need to
spend more ETH!");

    constructor() {
        i_owner = msg.sender;
    }
}
```

Tambahan address i_owner dan constructor untuk i_owner = msg.sender untuk mendata address yang menggunakan contract tersebut.

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.8;

mapping(address => uint256) public addressToAmountFunded;
address[] public funders;

contract FundMe {

    address public /* immutable */ i_owner;
```

```

uint256 public constant MINIMUM_USD = 50 * 10 ** 18;

function fund() public payable {
    require(msg.value.getConversionRate() >= MINIMUM_USD, "You need to
spend more ETH!");
    addressToAmountFunded[msg.sender] += msg.value;

    funders.push(msg.sender);
}
constructor() {
    i_owner = msg.sender;
}
modifier onlyOwner {

    // require(msg.sender == owner);
    if (msg.sender != i_owner) revert NotOwner();
}

function withdraw() payable onlyOwner public {
    for (uint256 funderIndex=0; funderIndex < funders.length;
funderIndex++){
        address funder = funders[funderIndex];
        addressToAmountFunded[funder] = 0;
    }
    funders = new address[](0);
    (bool callSuccess, ) = payable(msg.sender).call{value:
address(this).balance}("");
    require(callSuccess, "Call failed");
}

```

Kita akan menambahkan beberapa fungsi, pertama ada yang diluar kontrak adalah mapping untuk menghitung berapa banyaknya akun yang harus didanakan, dan membuat array address funders yang akan digunakan dalam contract. Dalam function fund kita akan menambahkan jumlah akun yang harus didanakan setiap kali function fund dipanggil dan address akan dimasukkan ke dalam array funders.

Kemudian kita membuat function baru yaitu function withdraw. Fungsi ini berfungsi untuk menarik semua dana yang telah diisi dalam fungsi fund. Kita ingin hanya contract owner yang dapat menggunakan function ini, cara yang akan kita pakai untuk itu adalah membuat modifier onlyOwner yang akan mengecek apabila yang menggunakan fungsi adalah owner, apabila bukan owner, maka akan ditolak.

2. PriceConverter.sol

Kita membutuhkan cara untuk membaca nilai mata uang usd tanpa merusak desentralized system blockchain, salah satu solusinya adalah menggunakan AggregatorV3Interface sebagai library yang akan mengeluarkan output harga mata uang usd dalam bentuk wei. Source codenya adalah sebagai berikut:

```

pragma solidity ^0.8.8;

import "@chainlink/contracts/src/v0.8/interfaces/AggregatorV3Interface.sol";

// Why is this a library and not abstract?
// Why not an interface?
library PriceConverter {
    // We could make this public, but then we'd have to deploy it
    function getPrice() internal view returns (uint256) {
        // Rinkeby ETH / USD Address
        // https://docs.chain.link/docs/ethereum-addresses/
        AggregatorV3Interface priceFeed = AggregatorV3Interface(
            0x8A753747A1Fa494EC906cE90E9f37563A8AF630e
        );
        (, int256 answer, , , ) = priceFeed.latestRoundData();
        // ETH/USD rate in 18 digit
        return uint256(answer * 1000000000000000000);
    }

    // 1000000000
    function getConversionRate(uint256 ethAmount)
        internal
        view
        returns (uint256)
    {
        uint256 ethPrice = getPrice();
        uint256 ethAmountInUsd = (ethPrice * ethAmount) / 1000000000000000000;
        // the actual ETH/USD conversion rate, after adjusting the extra 0s.
        return ethAmountInUsd;
    }
}

```

Kita akan mengimport library AggregatorV3Interface dari github, kemudian kita menambahkan fungsi getPrice yang berfungsi untuk menarik harga uang mata usd pada pasar, kemudian dengan fungsi getConversionRate kita akan mengkonversi nilai mata uang usd ke nilai mata uang crypto wei.

3. Import PriceConverter.sol ke FundMe.sol

Sekarang kita akan mengimport library yang telah kita buat kedalam kontrak kita dengan menambahkan dua baris kode ini:

```

import "@chainlink/contracts/src/v0.8/interfaces/AggregatorV3Interface.sol";
import "./PriceConverter.sol";

```

dan menambahkan satu baris kode ini di dalam kontrak FundMe pada baris pertama:

```

using PriceConverter for uint256;

```

serta menambahkan fungsi getVersion untuk AggregatorV3Interface di dalam kontrak FundMe:

```
function getVersion() public view returns (uint256){  
    AggregatorV3Interface priceFeed =  
AggregatorV3Interface(0x8A753747A1Fa494EC906cE90E9f37563A8AF630e);  
    return priceFeed.version();  
}
```

Terakhir kita akan menambahkan fallback dan receive untuk penyelesaikan kontrak di paling bawah kontrak:

```
fallback() external payable {  
    fund();  
}  
receive() external payable {  
    fund();  
}
```

Course 5: Ethers.js Simple Storage

1. Installasi

Sebelum kita mulai, ada beberapa hal yang harus kita install agar source code kita bisa berjalan dengan lancar, di antara itu adalah:

- [vscode](#): sebagai code editor
- [git](#): untuk mengupload kode online
- [node.js](#): untuk meng-compile javascript
- [yarn](#): untuk installasi beberapa module
- [ganache](#): untuk mendeploy fake etherium account secara lokal

2. SimpleStorage.sol

Buatlah folder baru untuk course baru ini dan di dalam folder tersebut buatlah file “SimpleStorage.sol” dan copypaste kode berikut:

```
// I'm a comment!
// SPDX-License-Identifier: MIT

pragma solidity 0.8.7;

// pragma solidity ^0.8.0;
// pragma solidity >=0.8.0 <0.9.0;

contract SimpleStorage {
    uint256 favoriteNumber;

    struct People {
        uint256 favoriteNumber;
        string name;
    }
    // uint256[] public anArray;
    People[] public people;

    mapping(string => uint256) public nameToFavoriteNumber;

    function store(uint256 _favoriteNumber) public {
        favoriteNumber = _favoriteNumber;
    }

    function retrieve() public view returns (uint256) {
        return favoriteNumber;
    }

    function addPerson(string memory _name, uint256 _favoriteNumber)
public {
```

```

        people.push(People(_favoriteNumber, _name));
        nameToFavoriteNumber[_name] = _favoriteNumber;
    }
}

```

Ini adalah smart contract yang akan kita pakai untuk course simple storage menggunakan ethers.js.

Apabila ingin mengganti formatting solidity untuk lebih rapih setiap kali kita save, install “solidity + hardhat” dari tab Extension vscode, kemudian klik View>Command Palletes...>settings dan pilih settings(JSON) dan tambahkan kode berikut di baris paling bawah

```

"[solidity]": {
    "editor.defaultFormatter": "NomicFoundation.hardhat-solidity"
}

```

Setelah itu balik lagi ke command palattes dan ketik settings dan pilih user settings, cari format to save dan centrang box tersebut.

3. Deploy.js

Buatlah file baru bernama “deploy.js” dan copypaste kode berikut:

```

async function main() {

}

main()
    .then(() => process.exit(0))
    .catch((error) => {
        console.error(error)
        process.exit(1)
})

```

Ini adalah fondasi dari kode yang akan kita pakai untuk mendeploy SimpleStorage.sol kita, tetapi sebelum kita dapat deploy kode solidity kita, kita harus install terlebih dahulu compiler solidity yaitu solc dengan memasukkan perintah berikut kedalam terminal

yarn add solc@0.8.7-fixed

setelah instalasi selesai, aka noda file baru bernama ”package.json”, di dalam file tersebut tambahkan baris kode berikut dibawah “dependencies”. Scripts ini akan meng-compile file solidity kita setiap kali kita mendeploy file js ini.

```

"scripts": {
    "compile": "yarn solcjs --bin --abi --include-path node_modules/ --
base-path . -o . SimpleStorage.sol"
}

```

Setelah itu install lagi menggunakan yarn dua module yang akan kita pakai selanjutnya, yaitu:

yarn add ethers

yarn add fs-extra

Kemudian kita akan menambahkan baris kode berikut diatas file deploy.js kita

```
const ethers = require("ethers")
const fs = require("fs-extra")
```

Sekarang kita akan menambahkan kode di dalam fungsi main() kita, kodenya seperti ini:

```
let provider = new ethers.providers.JsonRpcProvider(process.env.RPC_URL)
let wallet = new ethers.Wallet(process.env.PRIVATE_KEY, provider)

const abi = fs.readFileSync("./SimpleStorage_sol_SimpleStorage.abi",
"utf8")
const binary = fs.readFileSync("./SimpleStorage_sol_SimpleStorage.bin",
"utf8")

const contractFactory = new ethers.ContractFactory(abi, binary, wallet)
console.log("Deploying, please wait...")
const contract = await contractFactory.deploy()

const deploymentReceipt = await contract.deployTransaction.wait(1)
console.log(`Contract deployed to ${contract.address}`)
```

keterangan:

provider: block yang kita deploy dari aplikasi genache

wallet: private key dari salah satu fake account dari aplikasi genache

abi: file abi yang kita dapat saat file sol tercompile

bin: file bin yang kita dapat saat file sol tercompile

contractFactory: menggunakan module ethers untuk mendeploy file sol kita

contract: menunggu hasil contractFactory

deploymentReceipt: mengambil receipt dari transaksi

Sekarang kita akan menambahkan kode yang akan menampilkan hasil transaksi dengan lebih sederhana.

```
let currentFavoriteNumber = await contract.retrieve()

console.log(`Current Favorite Number: ${currentFavoriteNumber}`)
console.log("Updating favorite number...")
let transactionResponse = await contract.store(7)
let transactionReceipt = await transactionResponse.wait()
currentFavoriteNumber = await contract.retrieve()
console.log(`New Favorite Number: ${currentFavoriteNumber}`)
```

Seperti file sol kita sebelumnya, apabila kita ingin editing format yang raph untuk file js kita, kita dapat mnгинstall extension “Prettier – code formatter”, dan seperti sebelumnya kita akan menambahkan baris kode dalam settings.json kita yaitu seperti ini:

```
"[javascript]": {
    "editor.defaultFormatter": "esbenp.prettier-vscode"
}
```

4. .env

Untuk mengisi beberapa info yang kurang seperti block genesis dan private key kita agar aman dari publik, kita akan menggunakan module dotenv yang akan menyimpan data private kita. Pertama, kita akan menginstall dotenv dengan perintah pada terminal

yarn add dotenv

setelah itu kita akan membuat file baru bernama “.env”, dalam file ini kita akan mengisi informasi private kita untuk di import ke dalam deploy.js kita. Isi file .env adalah sebagai berikut:

```
PRIVATE_KEY=/*isi dengan private key yang akan dipakai*/
RPC_URL=http://{ip address block yang akan dipakai}
PRIVATE_KEY_PASSWORD=/*password yang akan digunakan untuk enkripsi
di bagian selanjutnya*/
```

Agar file .env kita tidak terupload oleh git apabila kita akan menpublish kode kita maka kita akan membuat file baru bernama “.gitignore” dimana isi dari file ini adalah list file atau folder yang kita tidak ingin untuk di tarik oleh git.

5. (optional) encryptKey.js

Apabila kita menginginkan keamanan yang lebih kuat kita dapat meng-enkripsi private key kita dengan membuat file js baru bernama “encryptKey.js”. isi dari file ini adalah

```
const ethers = require("ethers")
const fs = require("fs-extra")
require("dotenv").config()

async function main() {
    console.log(process.env.PRIVATE_KEY)
    console.log(process.env.PRIVATE_KEY_PASSWORD)
    const wallet = new ethers.Wallet(process.env.PRIVATE_KEY)
    const encryptedJsonKey = await wallet.encrypt(
        process.env.PRIVATE_KEY_PASSWORD,
        process.env.PRIVATE_KEY
    )
    console.log(encryptedJsonKey)
    fs.writeFileSync("./.encryptedKey.json", encryptedJsonKey)
}

main()
    .then(() => process.exit(0))
    .catch((error) => {
        console.error(error)
        process.exit(1)
})
```

Kita akan langsung mendeploy file ini dengan perintah terminal

Deploy encryptKey.js

Setalah selesai berjalan kita akan mendapatkan file baru bernama “encryptedKey.json”. file ini berisi hasil enkripsi kita yang akan kita akan pakai mulai sekarang, dan apabila kita akan memakai private key baru, kita tinggal mendeploy file encryptKey.js lagi.

Cara menggunakan file hasil enkripsi kita adalah dengan merubah isi file deploy.js kita, hasil edit yang menggunakan file enkripsi kita dalam sebagai berikut:

```
let provider = new
ethers.providers.JsonRpcProvider(process.env.RPC_URL)
// let wallet = new ethers.Wallet(process.env.PRIVATE_KEY, provider)
const encryptedJson = fs.readFileSync("./.encryptedKey.json", "utf8");
let wallet = new ethers.Wallet.fromEncryptedJsonSync(
    encryptedJson,
    process.env.PRIVATE_KEY_PASSWORD
);
wallet = wallet.connect(provider);
const abi = fs.readFileSync("./SimpleStorage_sol_SimpleStorage.abi",
"utf8")
const binary = fs.readFileSync(
    "./SimpleStorage_sol_SimpleStorage.bin",
    "utf8"
)
```

6. (optional) .prettierrc

Bagian ini adalah bagian opsional apabila kita ingin membuat kode kita terlihat lebih rapi dengan cara yang mudah. Kita akan menginstall module prettier dari terminal dengan perintah `yarn add prettier prettier-plugin-solidity`

setelah terinstall, maka kita akan membuat file baru bernama “.prettierrc” yang berisi setting style kode kita seperti berikut:

```
{
  "tabWidth": 4,
  "useTabs": false,
  "semi": false,
  "singleQuote": false
}
```

7. Alchemy testnet

-Registrasi ke [Alchemy](#) dan buatlah etherium blockcahin ecosystem (contoh menggunakan network rinkeby)

-setelah registrasi selesai copy RCP HTTP URL dan paste kan ke dalam file .env menggantikan RCP_URL yang sebelumnya menggunakan genache

-untuk private key rinkeby kita dapat menggunakan fake private key apabila untuk percobaan saja atau menggunakan service seperti metamask yang menggunakan network rinkeby untuk menggantikan PRIVATE_KEY di dalam file .env.

-setelah semua yang diatas telah selesai, run deploy.js

Course 6: Hardhat Simple Storage

1. Hardhat

Pada course ini kita akan membuat Hardhat Simple Storage dengan menggunakan beberapa hal yang kita pelajari dari course sebelumnya. Pertama kita akan menginisialisasi package.json dengan perintah `yarn init` dan menginstall hardhat dengan perintah dan `yarn add --dev hardhat`. setelah instalasi selesai kita akan mengganti greeter.sol yang ada di folder contracts menjadi SimpleStorage.sol dengan isi kode sama seperti course sebelumnya yaitu:

```
// I'm a comment!
// SPDX-License-Identifier: MIT
pragma solidity 0.8.8;

// pragma solidity ^0.8.0;
// pragma solidity >=0.8.0 <0.9.0;

contract SimpleStorage {
    uint256 favoriteNumber;

    struct People {
        uint256 favoriteNumber;
        string name;
    }

    // uint256[] public anArray;
    People[] public people;

    mapping(string => uint256) public nameToFavoriteNumber;

    function store(uint256 _favoriteNumber) public {
        favoriteNumber = _favoriteNumber;
    }

    function retrieve() public view returns (uint256) {
        return favoriteNumber;
    }

    function addPerson(string memory _name, uint256 _favoriteNumber) public
{
    people.push(People(_favoriteNumber, _name));
    nameToFavoriteNumber[_name] = _favoriteNumber;
}
}
```

Pastikan versi solidity di dalam hardhat.config.js sesuai dengan versi solidity yang ada dalam kode SimpleStorage.sol agar tidak mendapatkan error.

2. Deploy.js

Pada folder scripts kita akan menemukan file sample-scripts.js, kita akan mengubah nama file tersebut menjadi deploy.js dan menghapus semua isis kode tersebut dan mengganti isi file dengan kode berikut:

```
async function main() {  
  
}  
  
main()  
.then(() => process.exit(0))  
.catch((error) => {  
  console.error(error)  
  process.exit(1)  
})
```

Kemudian kita akan menginstall prettier plugins menggunakan perintah `yarn add prettier prettier-plugin-solidity`, membuat file baru bernama “.prettierrc” dan mengisi file dengan kode berikut:

```
{  
  "tabWidth": 2,  
  "useTabs": false,  
  "semi": false,  
  "singleQuote": false  
}
```

Pada file deploy.js kita akan menambahkan beberapa baris kode, yaitu

```
const { ethers, run, network } = require("hardhat")
```

kode di atas akan mengimport ethers, run, dan network dari hardhat yang kana membantu memudahkan pembuatan simple storage ini. Kemudian dalam fungsi main() kita akan menambahkan kode berikut:

```
const SimpleStorageFactory =  
  await ethers.getContractFactory("SimpleStorage")  
console.log("Deploying contract...")  
const simpleStorage = await SimpleStorageFactory.deploy()  
await simpleStorage.deployed()  
console.log(`Deployed contract to: ${simpleStorage.address}`)
```

dengan menggunakan hardhat kita dapat mendeploy smart contract ini tanpa harus mengisi private key atau RPC url, tetapi apabila kita ingin memakai network tertentu dan private key tertentu kita dapat melakukannya dengan meng-edit file hardhat.config.js pada bagian module.exports kiat akan menambahkan

```
defaultNetwork: "hardhat",
networks: {
  hardhat: {},
  rinkeby: {
    url: RINKEBY_RPC_URL,
    accounts: [PRIVATE_KEY],
    chainId: 4,
  },
}
```

Kita akan menambahkan file .env dengan menginstall dotenv dengan perintah `yarn add dotenv` kemudian membuat file .env yang berisi data rcp url dan private key kita. Dan jangan lupa untuk mengimportnya ke hardhat.config.js dengan menambahkan baris ini di bagian paling atas

```
require("dotenv").config()
```

Dan di atas module.exports kita juga akan menambahkan baris ini

```
const RINKEBY_RPC_URL =
  process.env.RINKEBY_RPC_URL

const PRIVATE_KEY =
  process.env.PRIVATE_KEY
```

setelah itu kita akan menambahkan fitur verify kedalam kode kita dengan menggunakan etherscan. Pertama kita harus install etherscan menggunakan perintah `yarn add --dev @nomiclabs/hardhat-etherscan` setelah instalasi selesai kita akan membuat akun di etherscan untuk mendapatkan API key. Saat kita telah mendapatkan API key kita akan menambahkannya dalam file .env kita dengan nama variabel ETHERSCAN_API_KEY dan mengimportnya seperti di atas dengan menambahkan

```
require("@nomiclabs/hardhat-etherscan")
  dan dalam module.exports kita juga menambahkan
```

```
etherscan: {
  apiKey: ETHERSCAN_API_KEY,
},
```

Untuk menggunakan fitur verify ini kita akan meng-edit file deploy.js kita menambahkan fungsi baru dibawah fungsi main() kita bernama “verify”. Dalam fungsi verify ini kita isi dengan kode berikut:

```
const verify = async (contractAddress, args) => {
  console.log("Verifying contract...")
  try {
    await run("verify:verify", {
      address: contractAddress,
      constructorArguments: args,
    })
  }
```

```

} catch (e) {
  if (e.message.toLowerCase().includes("already verified")) {
    console.log("Already Verified!")
  } else {
    console.log(e)
  }
}

```

Dan menambahkan dalam fungsi main kita

```

const SimpleStorageFactory =
  await ethers.getContractFactory("SimpleStorage")
console.log("Deploying contract...")
const simpleStorage = await SimpleStorageFactory.deploy()
await simpleStorage.deployed()
console.log(`Deployed contract to: ${simpleStorage.address}`)
// what happens when we deploy to our hardhat network?

if (network.config.chainId === 4 && process.env.ETHERSCAN_API_KEY)
{
  console.log("Waiting for block confirmations...")
  await simpleStorage.deployTransaction.wait(6)
  await verify(simpleStorage.address, [])
}

const currentValue = await simpleStorage.retrieve()
console.log(`Current Value is: ${currentValue}`)

// Update the current value
const transactionResponse = await simpleStorage.store(7)
await transactionResponse.wait(1)
const updatedValue = await simpleStorage.retrieve()
console.log(`Updated Value is: ${updatedValue}`)

```

fungsi if akan melakukan verifikasi transaksi kita dan kemudian currentValue akan menampilkan value saat ini dan kemudian akan diperbarui dengan value yang baru oleh updatedValue.

3. (optional) Block-number.js

Di bagian ini kita akan mencoba membuat task yang merupakan semacam perintah yang bisa kita panggil tanpa harus memanggil main script kita. Contoh yang kali ini kita coba buat adalah task untuk mendapatkan block number dari network kita. Pertama kita akan membuat folder baru bernama “tasks” dan di dalamnya menambahkan file task baru kita yaitu “block-number.js” yang berisi kode berikut:

```

const { task } = require("hardhat/config")

task("block-number", "Prints the current block number").setAction(

```

```
async (taskArgs, hre) => {
  const blockNumber = await hre.ethers.provider.getBlockNumber()
  console.log(`Current block number: ${blockNumber}`)
}

module.exports = {}
```

kemudian pada hardhat.config.js kita akan menambahkan import berikut diatas kode

```
require("./tasks/block-number")
```

4. Localhost dan console

Kita dapat menjalankan virtual blockchain lokal pada komputer kita mirip seperti genache dengan menggunakan hardhat. Cara nya adalah dengan menggunakan perintah *yarn hardhat node* yang kemudian akan menunjukkan link http localhost yang akan kita tambahkan ke dalam hardhat.config.js dalam module.exports di bagian networks seperti ini:

```
localhost: {
  url: "http://localhost:8545",
  chainId: 31337,
},
```

Kita juga dapat memanggil fungsi apapun dalam smart contract kita menggunakan hardhat console. Cara kita untuk mengakses hardhat consol tersebut adalah dengan menggunakan perintah *yarn hardhat console --network {network names}* yang akan memasukkan kita ke dalam shell hardhat. Dalam shell hardhat ini kita dalam memanggil fungsi apapun dari project kita, hal ini digunakan untuk testing apakah suatu fungsi berjalan edngan benar tanpa menjalankan seluruh file .js.

5. Test-Deploy.js dan gasReporter

Untuk meningkatkan keamanan dari smart contract kita, kita akan membuat test-deploy.js pada folder test yang akan menggantikan sample-test.js. file test ini berguna untuk kita melakukan testing pada contract kita agar setiap fungsi berjalan dengan benar saat kita mempublish kode kita ke publik. Dalam file test-deploy.js ini kita akan isi dengan kumpulan kode berikut:

```
const { ethers } = require("hardhat")
const { expect, assert } = require("chai")

// describe("SimpleStorage", () => {})
describe("SimpleStorage", function () {
  // let simpleStorageFactory
  // let simpleStorage
  let simpleStorageFactory, simpleStorage
  beforeEach(async function () {
    simpleStorageFactory = await
    ethers.getContractFactory("SimpleStorage")
```

```
    simpleStorage = await simpleStorageFactory.deploy()
  })

it("Should start with a favorite number of 0", async function () {
  const currentValue = await simpleStorage.retrieve()
  const expectedValue = "0"
  // assert
  // expect
  assert.equal(currentValue.toString(), expectedValue)
  // expect(currentValue.toString()).to.equal(expectedValue)
})
it("Should update when we call store", async function () {
  const expectedValue = "7"
  const transactionResponse = await
simpleStorage.store(expectedValue)
  await transactionResponse.wait(1)

  const currentValue = await simpleStorage.retrieve()
  assert.equal(currentValue.toString(), expectedValue)
})

// Extra - this is not in the video
it("Should work correctly with the people struct and array", async function () {
  const expectedPersonName = "Patrick"
  const expectedFavoriteNumber = "16"
  const transactionResponse = await simpleStorage.addPerson(
    expectedPersonName,
    expectedFavoriteNumber
  )
  await transactionResponse.wait(1)
  const { favoriteNumber, name } = await simpleStorage.people(0)
  // We could also do it like this
  // const person = await simpleStorage.people(0)
  // const favNumber = person.favoriteNumber
  // const pName = person.name

  assert.equal(name, expectedPersonName)
  assert.equal(favoriteNumber, expectedFavoriteNumber)
})
})
```

Kita juga melakukan testing mengenai gas price dari fungsi yang kita pakai menggunakan gas reporter. Kita akan menginstall gas reporter dengan perintah `yarn add hardhat-gas-reporter --dev`. kemudian setelah selesai kita akan mengimportnya pada hardhat.config.js kita seperti ini

```
require("hardhat-gas-reporter")
```

kita juga akan menambahkan export ke module.exports seperti ini

```
gasReporter: {  
    enabled: true,  
    currency: "USD",  
    outputFile: "gas-report.txt",  
    noColors: true,  
    coinmarketcap: COINMARKETCAP_API_KEY,  
},
```

Untuk mendapatkan COINMARKETCAP_API_KEY kita akan melakukan hal yang sama seperti etherscan. Masuk ke website [CoinMarketCap](#) dan registrasi akun baru, setelah selesai registrasi carilah API key dari akun kita dan kita dapat menambahkannya ke .env file kita dan memanggilnya pada hardhat.config.js menggunakan kode berikut:

```
const ETHERSCAN_API_KEY = process.env.ETHERSCAN_API_KEY || ""
```

terakhir untuk tambahan keamanan kita dapat menggunakan solidity coverage. Ini adalah module yang akan melakukan scan pada kode kita dan mencari kelemahan dari keamanan dari kode kita. Kita dapat menggunakan solidity coverage dengan menginstall nya dengan perintah `yarn add --dev solidity-coverage` dan menambahkan importnya pada hardhat.config.js kita seperti ini

```
require("solidity-coverage")
```

kita dapat melakukan coverage dengan perintah `yarn hardhat coverage` yang kemudian akan memberikan kita hasil coverage kode kita dalam file baru bernama “coverage.json”.

LESSON 7: HARDHAT FUND ME

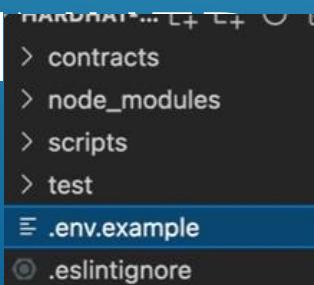
Reference: Patrick Collins

- ▶ Melakukan Clone dan mealkukan yarn hardhat fund me from github
<https://github.com/PatrickAlphaC/hardhat-fund-me-fcc>
- ▶ Setelah kita clone lalu lakukan perintah hal seperti dibawah ini:
 - ▶ “code .” dan masuk ke terminal lagi lalu
 - ▶ yarn
 - ▶ yarn hardhat deploy

HARDHAT SETUP - FUND ME

- ▶ Selanjutnya membuat direktori baru dengan perintah dibawah:
 - ▶ mkdir hardhat-fund-me-fcc
 - ▶ cd hardhat-fund-me-fcc
 - ▶ “code .” dan masuk ke terminal lagi lalu
 - ▶ yarn
 - ▶ yarn add –dev hardhat
 - ▶ yarn hardhat, lalu →
 - ▶ dan akan muncul ↓

HARDHAT SETU CONTI



A screenshot of a terminal window showing the Hardhat v2.9.3 setup process. The output is:

```
👷 Welcome to Hardhat v2.9.3 🎷

✓ What do you want to do? · Create an advanced sample project
✓ Hardhat project root: · /Users/patrick/hh-fcc/hardhat-fund-me-fcc
✓ Do you want to add a .gitignore? (Y/n) · y
? Do you want to install this sample project's dependencies with yarn? · n
```

- ▶ Selanjutnya menghapus “.npmignore” pada bagian kiri
- ▶ Lalu akan melakukan

HARDHAT S
CONTINUED



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

- Code Editor:** Displays a JavaScript file named `FundMe.test.js`. A yellow dashed circle highlights the line of code: `const endingFundMeBalance = await fundMe.provider.getBalance(fundMe.address);`
- VARIABLES:** Shows local variables: `endingDeployerBalance`, `endingFundMeBalance`, `startingDeployerBalance`, and `startingFundMeBalance`.
- WATCH:** No items are listed.
- PROBLEMS:** 2 issues found.
- TERMINAL:** Shows the command: `$ /Users/patrick/hh-fcc/hardhat-fund-me-fcc/node_modules/.bin/hardhat test`. The output includes: `Debugger attached.`
- CALL STACK:** Shows a single entry: `<anonymous> test/unit/...` with a status of `PAUSED ON BREAKPOINT`.
- LOADED SCRIPTS:** Shows a list of loaded scripts including `FundMe`, `constructor`, `fund`, and `callFn`.
- JavaScript Debug:** A sidebar showing coverage analysis for the `FundMe` contract.

BREAKPOINTS & DEBUGGING

- ▶ Membuat repository lalu kita add dengan “git add <filename>”, lalu kita commit dengan “git commit -m "Commit message" ”, lalu kita push menggunakan “git push origin master”

PUSHING TO GITHUB

LESSON 8: HTML/JAVASCRIPT FUND ME

Reference: Patrick Collins

In order for web3 to be approachable to the masses, we need to have user-friendly frontends and websites. There are a few challenges that full stack software engineers run into when approaching this problem in the blockchain space.

How do I connect Metamask to my UI? (Or Walletconnect, Phantom, etc.)

How do I execute a transaction with my smart contract from a website?

What are the tools the best of the best are doing?

So, in asking myself this problem and trying to figure out what to recommend to developers, I ended up looking at nearly ALL the most popular solutions. So in this article, we are going to:

Understand what is going on in the browser when we want to interact with or send a transaction to a blockchain.

We look at six of the most popular methods to connect to our web3 applications

Give code examples and show what all the biggest players in the space use, so you can use the same tools!

INTRODUCTION

If you'd like to see what some professional frontends look like right now, you can take a look at the Aave or Uniswap website.

Connected

getBalance

Withdraw

ETH Amount

0.1

Fund

CONNECTING HTML TO METAMASK

- ▶ With a recent update (Mobile v5.3.0), we added a feature that allows you to add one of several popular networks without having to input any details.
- ▶ You can find this menu using the following steps:
 - ▶ Tap the hamburger icon in the top-left of the screen.
 - ▶ Go to 'Settings', and then to 'Networks'.
 - ▶ Tap the 'Add Network' button at the bottom of the screen.
 - ▶ You should now see a list of networks you can add under the 'Popular' tab. Tap one and follow the prompts to add it to MetaMask. All done!
- ▶ This feature will also be added to Extension soon.

SENDING A TRANSACTION FROM A WEBSITE



```
JS index.js > ⚙ fund

18  async function fund(ethAmount) {
19    console.log(`Funding with ${ethAmount}...`)
20    if (typeof window.ethereum !== "undefined") {
21      // provider / connection to the blockchain
22      // signer / wallet / someone with some gas
23      // contract that we are interacting with
24      // ^ ABI & Address
25      const provider = new ethers.providers.Web3Provider(window.ethereum)
26
27    }
28  }
29
30  // fund function
31
32  // withdraw
33
```

SENDING A TRANSACTION FROM A WEBSITE

MetaMask - RPC Error:

[ethjs-query] while formatting ouputs from RPC '{"value":{"code":-32603,"data":{"code":-32000,"message":"Nonce too high. Expected nonce to be 2 but got 4. Note that transactions can't be queued when automining."}}}'

RESETTING AN ACCOUNT IN METAMASK

```
8    // listen for the tx to be mined
9    // listen for an event <- we haven't learned about yet!
0  } catch (error) {
1    console.log(error)
2  }
3 }

4 function listenForTransactionMine(transactionResponse, provider){
5
6
7
8 }
```

```
try {
  const transactionResponse = await contract.fund({
    value: ethers.utils.parseEther(ethAmount),
  })
  // hey, wait for this TX to finish
  await listenForTransactionMine(transactionResponse, provider)
} catch (error) {
  console.log(error)
}

function listenForTransactionMine(transactionResponse, provider) {
  console.log(`Mining ${transactionResponse.hash}...`)
  return new Promise()
  // create a listener for the blockchain
}
```

LISTENING FOR EVENTS AND COMPLETED TRANSACTIONS

```
function listenForTransactionMine(transactionResponse, provider) {
  console.log(`Mining ${transactionResponse.hash}...`)
  // listen for this transaction to finish
  provider.once(transactionResponse.hash, (transactionReceipt) => {
    console.log([
      `Completed with ${transactionReceipt.confirmations} confirmations`,
    ])
  })
}

// fund
```

NEXTJS SMART CONTRACT LOTTERY

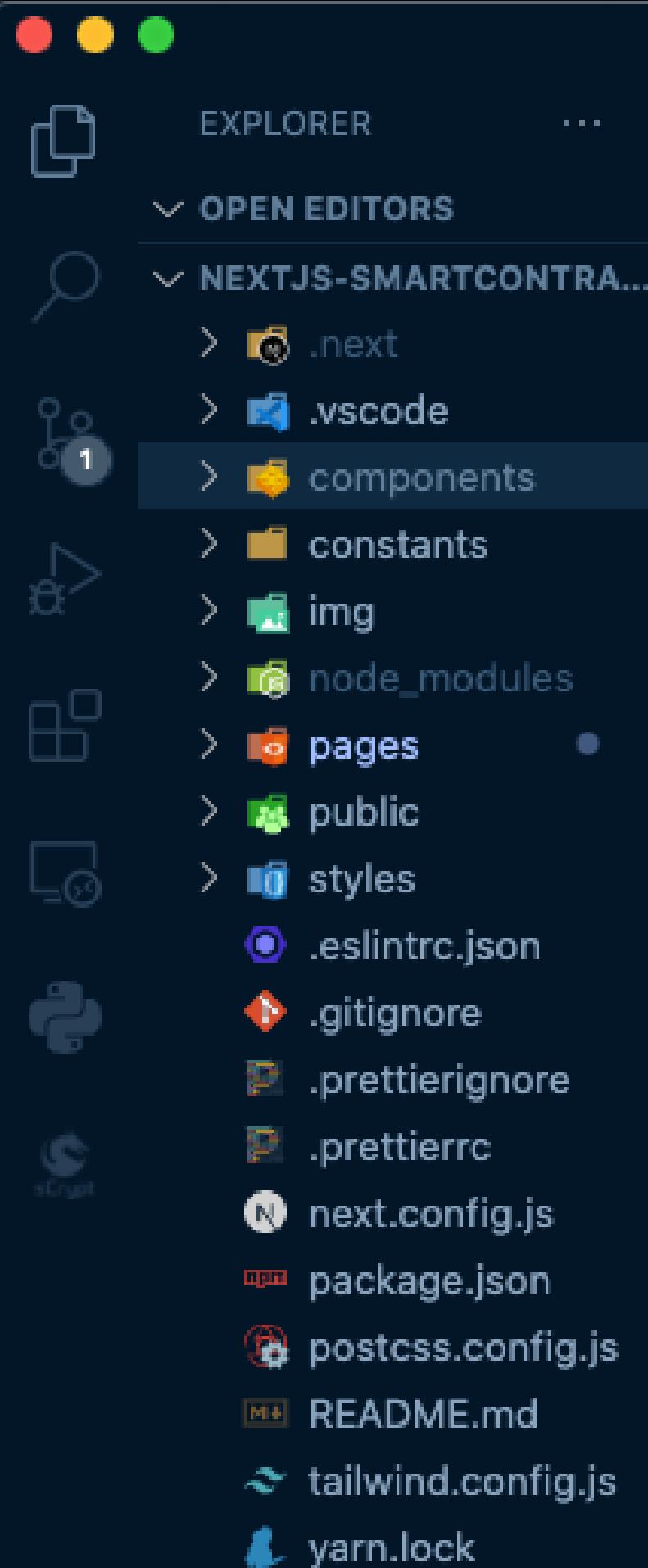
Blockchain

LIBRARY REACT YANG PERLU DISIAPKAN

- Web3React
- wagmi
- react-moralis
- useDapp
- Web3Modal
- useMetamask

Blockchain





NEXT JS SETUP

Persiapkan setup next JS dengan perintah tersebut di terminal.

```
yarn create next-app .
```

A screenshot of the Visual Studio Code interface. The top menu bar includes Code, File, Edit, Selection, View, Go, Run, Terminal, Window, and Help. The title bar shows "ManualHeader.js — nextjs-smartcontract-lott". The left sidebar has icons for Explorer, Open Editors, NextJS, components, Header.js, LotteryEntranc..., ManualHeader.js (which is selected), constants, img, node_modules, pages, public, styles, .eslintrc.json, .gitignore, .prettierignore, .prettierrc, next.config.js, package.json, postcss.config.js, README.md, tailwind.config.js, and yarn.lock. The main editor area displays the following code:

```
3 import { useEffect } from "react"
4 import { useMoralis } from "react-moralis"
5
6 // Top navbar
7 export default function ManualHeader() {
8     const { enableWeb3, isWeb3Enabled, isWeb3EnableLoading, account } = useMoralis()
9
10    useEffect(() => {
11        if (
12            !isWeb3Enabled &&
13            typeof window !== "undefined" &&
14            window.localStorage.getItem("connected")
15        ) {
16            enableWeb3()
17            // enableWeb3({provider: window.localStorage.getItem("connected")})
18        }
19    }, [isWeb3Enabled])
20    // no array, run on every render
21    // empty array, run once
22    // dependency array, run when the stuff in it changes
23
24    useEffect(() => {
25        Moralis.onAccountChanged((account) => {
26            console.log(`Account changed to ${account}`)
27            if (account == null) {
28                window.localStorage.removeItem("connected")
29                deactivateWeb3()
30            }
31            console.log("Null Account found")
32        })
33    }, [])
34
35    return (
36        <div>
```

COMPONENT

Membuat tampilan manual header jenis 1 dengan:

- JSX
- Moralis
- React Moralis

The screenshot shows the VS Code interface with the following details:

- EXPLORER**: Shows the project structure:
 - components > JS Header.js
 - NEXTJS... .next
 - .vscode
 - components
 - JS Header.js (selected)
 - JS LotteryEntranc...
 - JS ManualHeader.js
 - constants
 - img
 - node_modules
 - pages
 - public
- JS Header.js**: The active editor tab.
- Code Content:**

```
1 import { ConnectButton } from "web3uikit"
2
3 export default function Header() {
4     return (
5         <nav className="p-5 border-b-2 flex flex-row">
6             <h1 className="py-4 px-4 font-bold text-3xl"> Decentralized Lottery</h1>
7             <div className="ml-auto py-2 px-4">
8                 <ConnectButton moralisAuth={false}>
9                 </div>
10            </nav>
11        )
12    }
```

COMPONENT HEADER 2

Connection Button menggunakan web3uikit

Code File Edit Selection View Go Run Terminal Window Help

LotteryEntrance.js — nextjs-smartcontract-lottery-fcc

EXPLORER ... JS LotteryEntrance.js

OPEN EDITORS

components > JS LotteryEntrance.js > ...

```
1 import { contractAddresses, abi } from "../constants"
2 // dont export from moralis when using react
3 import { useMoralis, useWeb3Contract } from "react-moralis"
4 import { useEffect, useState } from "react"
5 import { useNotification } from "web3uikit"
6 import { ethers } from "ethers"

7
8 export default function LotteryEntrance() {
9     const { Moralis, isWeb3Enabled, chainId: chainIdHex } = useMoralis()
10    // These get re-rendered every time due to our connect button!
11    const chainId = parseInt(chainIdHex)
12    // console.log(`ChainId is ${chainId}`)
13    const raffleAddress = chainId in contractAddresses ? contractAddresses[chainId].address : null
14
15    // State hooks
16    // https://stackoverflow.com/questions/58252454/react-hooks-using-usestate
17    const [entranceFee, setEntranceFee] = useState("0")
18    const [numberOfPlayers, setNumberOfPlayers] = useState("0")
19    const [recentWinner, setRecentWinner] = useState("0")
20
21    const dispatch = useNotification()
22
23    const {
24        runContractFunction: enterRaffle,
25        data: enterTxResponse,
26        isLoading,
27        isFetching,
28    } = useWeb3Contract({
29        abi: abi,
30        contractAddress: raffleAddress,
31        functionName: "enterRaffle",
32        msgValue: entranceFee,
33        params: {},
34    })
```

> OUTLINE

> TIMELINE

CALLING FUNCTIONS IN NEXTJS

Menjalankan Contract Function:

- Moralis Provider
- useMoralis
- parseInt

Otomatis Memperbaharui Constant Value UI:

- ethers.utils.FormatTypes

TAILWIND & STYLING

Install Tailwind untuk
melakukan styling & jangan
lupa tambahkan extention
PostCSS, Tailwind.

The screenshot shows a VS Code interface with the following details:

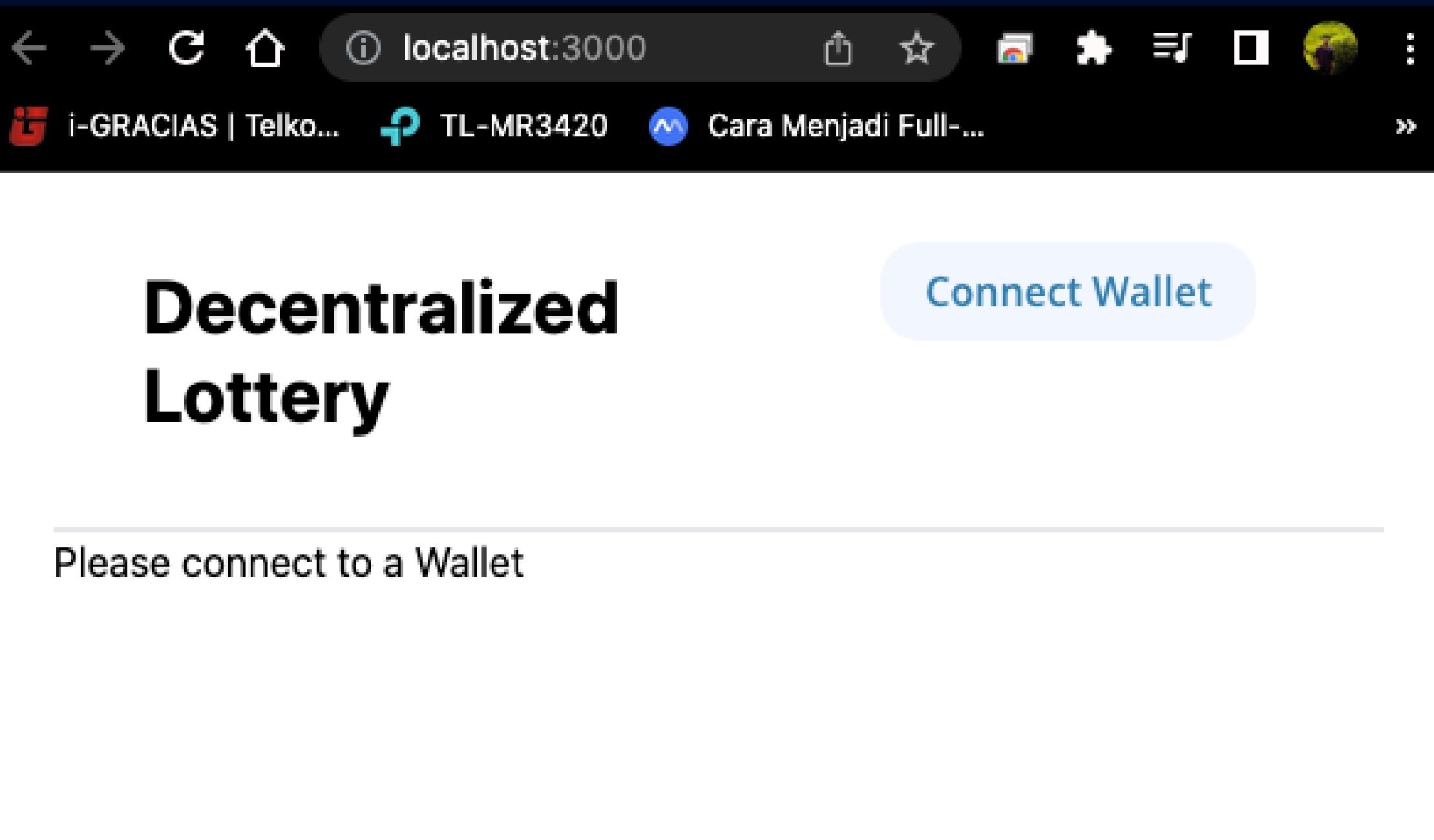
- EXPLORER:** Shows the project structure under "NEXTJS-SMARTCONTRA...".
- Home.module.css:** Contains CSS rules for ".container", ".main", ".footer", and ".title a".
- tailwind.config.js:** Contains the configuration for Tailwind, defining "module.exports" with "content", "theme", "extend", and "plugins".
- globals.css:** Contains Tailwind base, components, and utilities imports.

```
globals.css — nextjs-smartcontract-lottery-fcc
1 @tailwind base;
2 @tailwind components;
3 @tailwind utilities;

tailwind.config.js — tailwind.config.js > <unknown>
1 module.exports = {
2   ...
3   content: ["./pages/**/*.{js, ts, tsx}"],
4   theme: {
5     extend: {},
6   },
7   plugins: [],
8 }

tailwind.config.js — tailwind.config.js > <unknown>
1 .container {
2   padding: 0 2rem;
3 }
4
5 .main {
6   min-height: 100vh;
7   padding: 4rem 0;
8   flex: 1;
9   display: flex;
10  flex-direction: column;
11  justify-content: center;
12  align-items: center;
13 }
14
15 .footer {
16   display: flex;
17   flex: 1;
18   padding: 2rem 0;
19   border-top: 1px solid #eaeaea;
20   justify-content: center;
21   align-items: center;
22 }
23
24 .footer a {
25   display: flex;
26   justify-content: center;
27   align-items: center;
28   flex-grow: 1;
29 }
30
31 .title a {
32   color: #0070f3;
33   text-decoration: none;
34 }
```

Bottom status bar: Ln 3, Col 21 Spaces: 4 UTF-8 LF PostCSS ⚡ Go Live ✨ Prettier



RUNNING PROGRAM

Tampilan awal program di running

RUNNING PROGRAM

Pilihan menu koneksi
ke Wallet

localhost:3000

Decentralized Lottery

Please connect to a Wallet

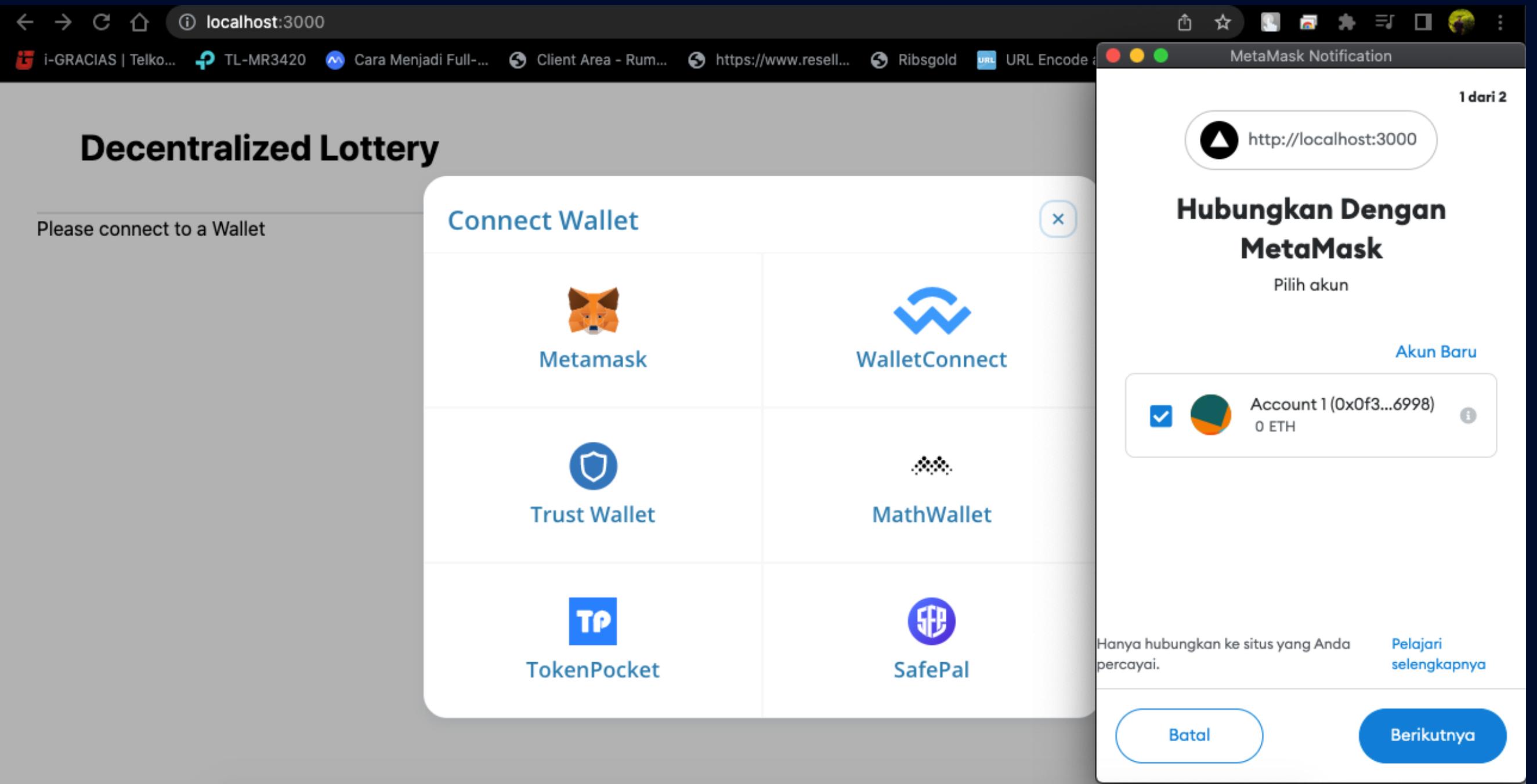
Connect Wallet

-  Metamask
-  WalletConnect
-  Trust Wallet
-  MathWallet
-  TokenPocket
-  SafePal

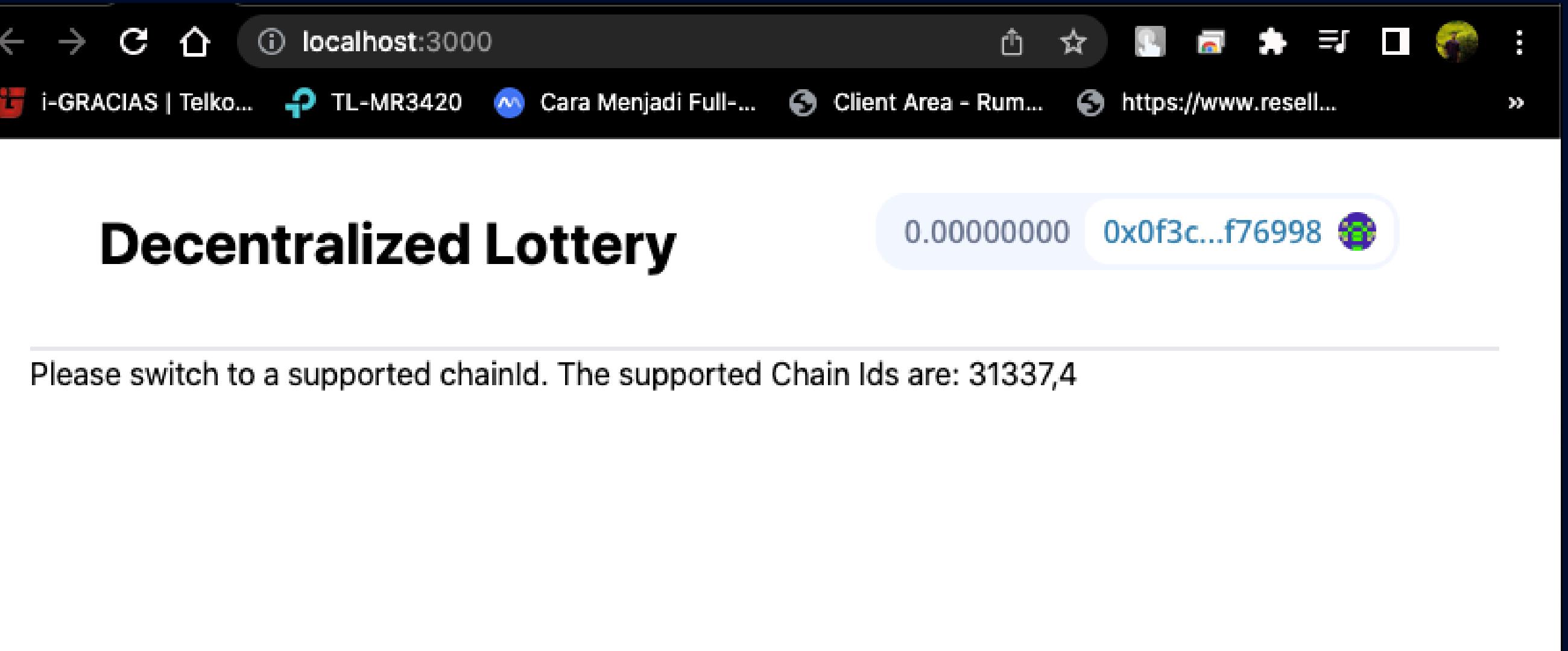
Connect Wallet

RUNNING PROGRAM

Setelah
mengkoneksikan pada
wallet Metamask



RUNNING PROGRAM



Tampilan setelah
terhubung pada wallet
Metamask

IPFS

Blockchain

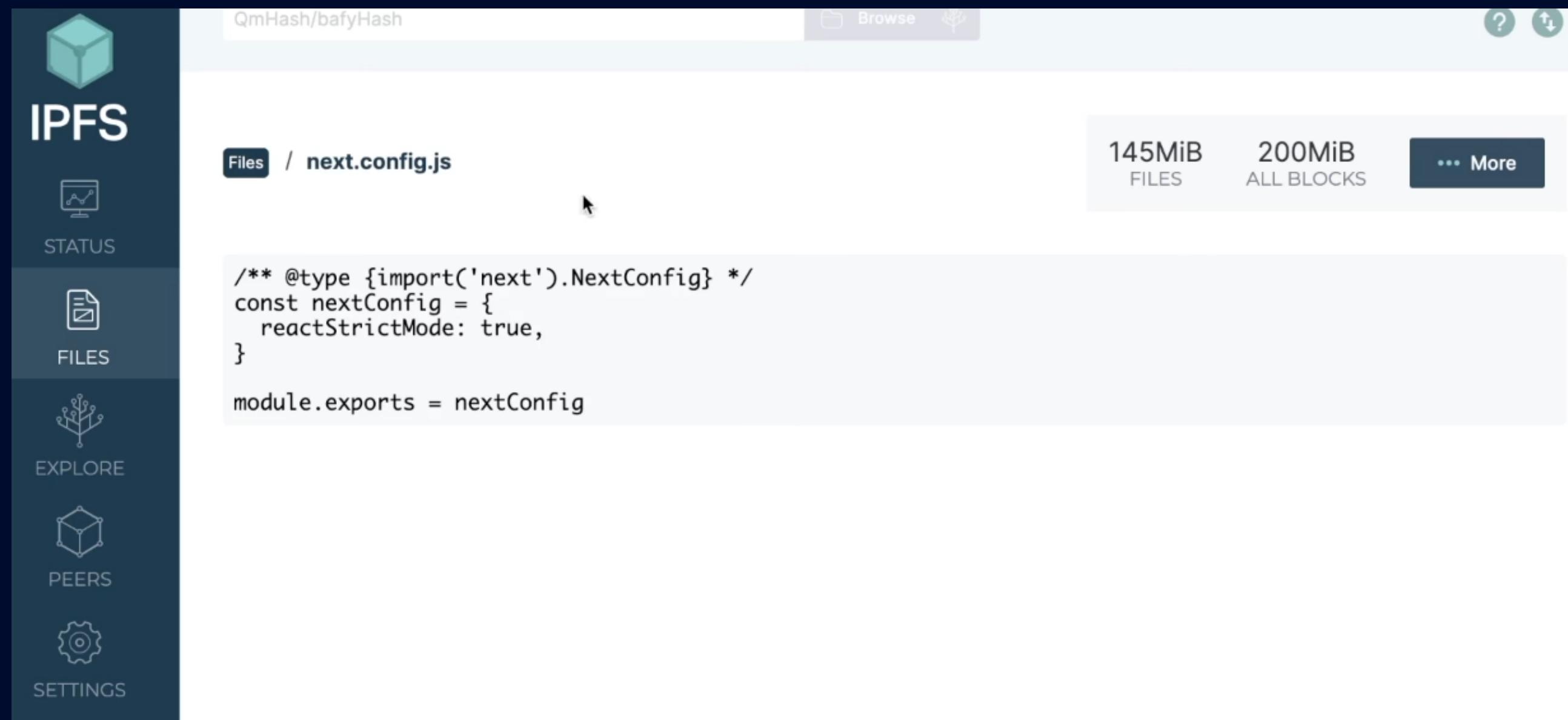
The screenshot shows the IPFS Desktop application window. On the left is a dark sidebar with the IPFS logo at the top, followed by menu items: STATUS (selected), FILES, EXPLORE, PEERS, and SETTINGS. At the bottom of the sidebar, it says "UI v2.15.0", "Revision fadd900", "See the code", and "Report a bug". The main content area has a header with "QmHash/bafyHash" and "Browse" buttons, along with a question mark and refresh icon. Below the header, the text "Connected to IPFS" is displayed, followed by "Hosting 18 KiB of data — Discovered 454 peers". It shows peer details: PEER ID (12D3KooWDjmW5JeeQWCm7MyfrdtQJTw44qznestLuPWURShZYPR), AGENT (go-ipfs v0.13.0 desktop), and UI (v2.15.0). A "► Advanced" link is also present. A modal dialog at the bottom asks for anonymous usage data with "OK" and "No thanks" buttons. The main area features two charts: "BANDWIDTH OVER TIME" showing fluctuating teal and orange lines on a grid, and "NETWORK TRAFFIC" showing a circular gauge with "27 KiB/s Incoming" and "0 KiB/s Outgoing".

IPFS

Install IPFS Desktop di
device anda

IPFS

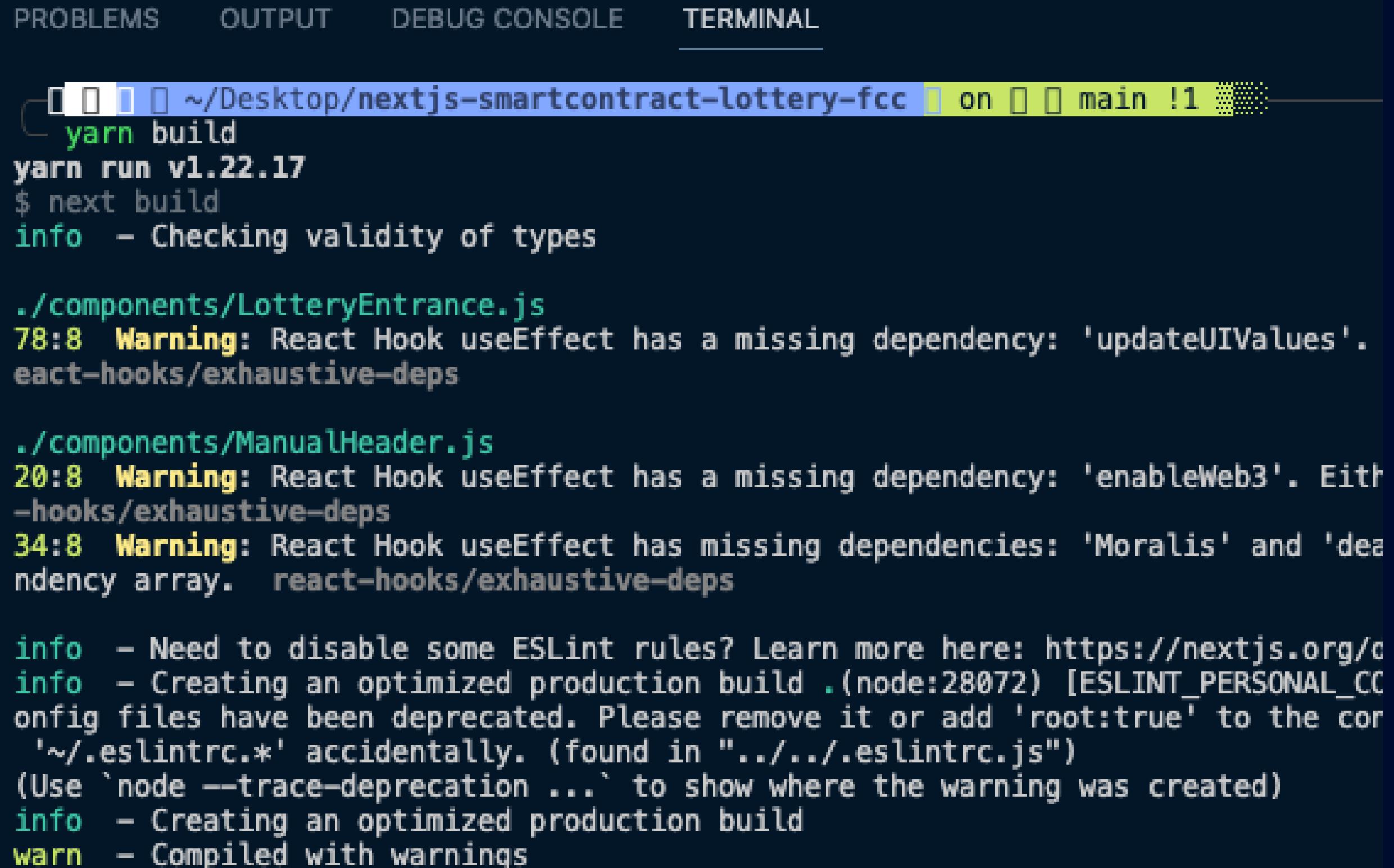
Open file tersebut lalu buka pada browser dan sediakan juga extention IPFS pada browser



IPFS BUILD DIREKTORI FILE

Setelah lakukan build adalah melakukan export pada terminal.

Coba running program tersebut dengan perintah "yarn run dev".



```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

C:\Users\~\Desktop\nextjs-smartcontract-lottery-fcc\on\main\!1
yarn build
yarn run v1.22.17
$ next build
info  - Checking validity of types

./components/LotteryEntrance.js
78:8 Warning: React Hook useEffect has a missing dependency: 'updateUIValues'.  
react-hooks/exhaustive-deps

./components/ManualHeader.js
20:8 Warning: React Hook useEffect has a missing dependency: 'enableWeb3'. Eith  
-hooks/exhaustive-deps
34:8 Warning: React Hook useEffect has missing dependencies: 'Moralis' and 'de  
ndency array. react-hooks/exhaustive-deps

info  - Need to disable some ESLint rules? Learn more here: https://nextjs.org/c  
info  - Creating an optimized production build .(node:28072) [ESLINT_PERSONAL_C  
onfig files have been deprecated. Please remove it or add 'root:true' to the cor  
'~/.eslintrc.*' accidentally. (found in ".../.eslintrc.js")
(Use `node --trace-deprecation ...` to show where the warning was created)
info  - Creating an optimized production build
warn  - Compiled with warnings
```

FINAL TEST IPFS

Decentralized Lottery

Hi from lottery entrance!

Enter Raffle

Entrance Fee: 0.1 ETH

Number Of Players: 5

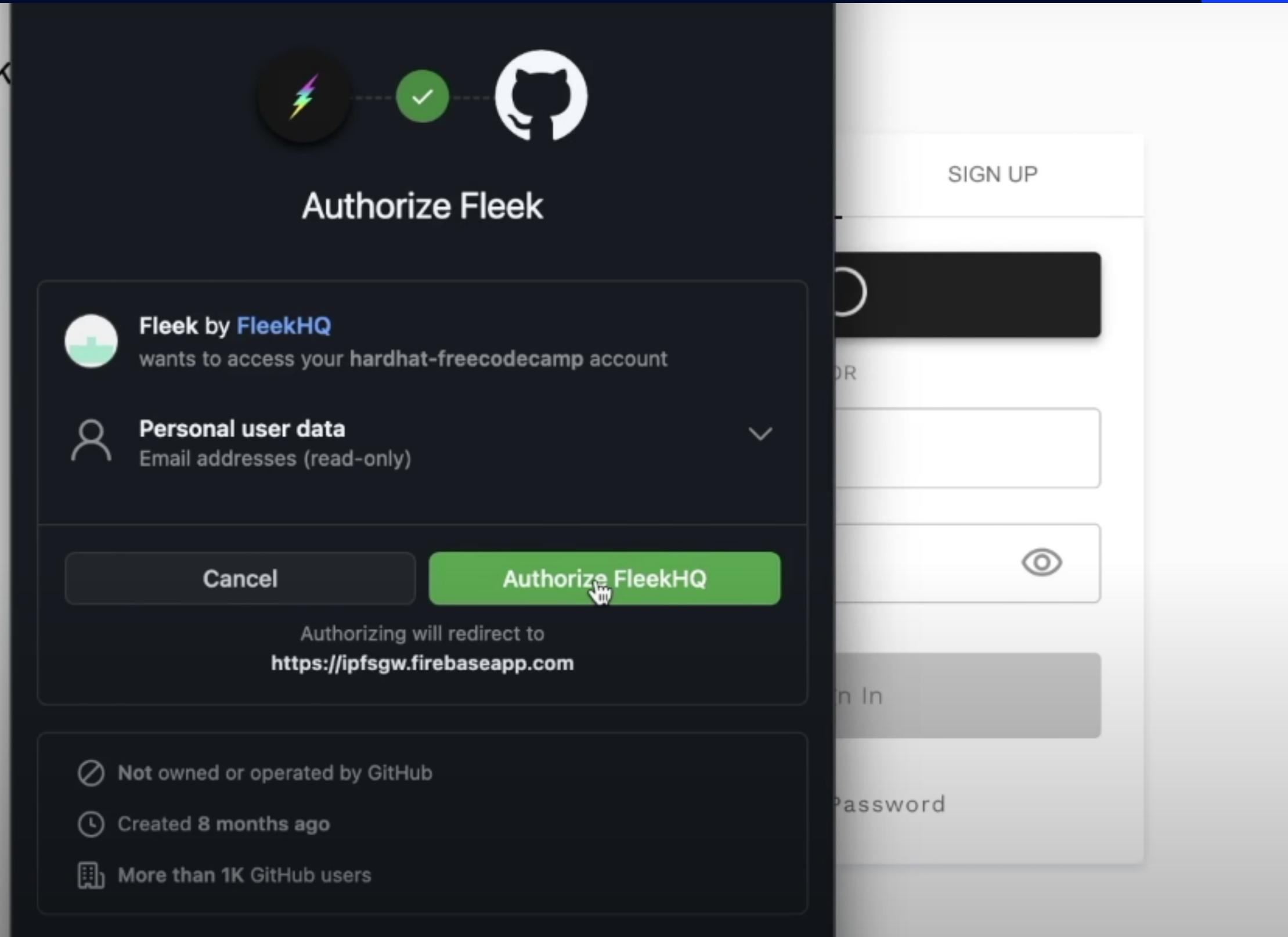
Recent Winner: 0xf39Fd6e51aad88F6F4ce6aB8827279cffFb92266

9999.49323010 0xf39f...b92266 



HOSTING ON IPFS & FILECOIN USING FLEEK

Future Technology
Blockchain



AUTHORIZE FLEEX

Hubungkan Fleex pada
github

SETTING FLEEK

Push Code di new repository baru yang sudah terhubung dengan flex dan Ubah build command seperti yang ada digambar tersebut

The screenshot shows the Fleek dashboard interface. On the left, there's a sidebar with options: Team (selected), Hosting (highlighted with a blue bar), Storage, Billing, Members, and Team Settings. The main area is titled "Hosting" and contains the following configuration fields:

- "If you're using a static site generator or build tool, we'll need to" (with a link to "Learn more in our docs →")
- "Framework": A dropdown menu set to "Other".
- "Docker Image Name (e.g. node:16)": The value is "fleek/next-js:node-16".
- "Build command": The value is "yarn install && yarn run build && yarn run export".
- "Publish directory": The value is "out".
- "Base Directory": This field is empty.

DEEPLY FLEEX

Lakukan Deploy pada
Fleex dan lihat status
File Coin ID.

The screenshot shows the Fleek web interface. On the left, there's a sidebar with a team dropdown set to "My team" and a navigation menu with "Hosting" selected. The main area shows a breadcrumb path "Hosting > holy-bird-9835". A sidebar on the right lists "General", "Build & Deploy" (which is expanded), "Continuous Deployment", "Advanced Build Settings", "Site Management", and "Domain Management". The "Build & Deploy" section contains fields for "Repository" (set to "https://api.github.com/repos/hardhat-freecodecamp/nextjs-smartcontract-lottery-fcc") with a "Change repository" button, "Base directory" (set to "Not set"), "Build command" (set to "yarn install && yarn run build && yarn next start"), and "Publish directory" (set to "out"). At the bottom, there's a link to "Learn more about common configuration directives in the docs →".

RUNNING CODE

CONNECT FLEEX

Decentralized Lottery

9999.39314657

0xf39f...b92266



Hi from lottery entrance!

Enter Raffle

Entrance Fee: 0.1 ETH

Number Of Players: 6

Recent Winner: 0xf39Fd6e51aad88F6F4ce6aB8827279cffFb92266

WHATS UPPP

THANK YOU

Fullstack Web3 Smart Contract

Future Blockchain Technology

BLOCKCHAIN

HARDHAT STARTER KIT



INSTALL FRAMEWORK

- npm install @chainlink/contracts --save
- yarn add @chainlink/contracts
- pipx install eth-brownie
- npm install truffle -g

```
brew install pipx
npm install @chainlink/contracts --save
npm WARN ERESOLVE overriding peer dependency
npm WARN While resolving: use-immer@0.6.0
npm WARN Found: react@18.2.0
npm WARN   node_modules/react
npm WARN     peer react@"^18.2.0" from react-dom@18.2.0
npm WARN   node_modules/react-dom
npm WARN     peer react-dom@">=17.0.0" from react-moralis@1.4.0
npm WARN   node_modules/react-moralis
npm WARN     1 more (the root project)
npm WARN     2 more (react-moralis, the root project)
npm WARN
npm WARN Could not resolve dependency:
npm WARN   peer react@"^16.8.0 || ^17.0.1" from use-immer@0.6.0
npm WARN   node_modules/react-moralis/node_modules/use-immer
npm WARN     use-immer@"^0.6.0" from react-moralis@1.4.0
npm WARN   node_modules/react-moralis
npm WARN
npm WARN Conflicting peer dependency: react@17.0.2
npm WARN   node_modules/react
npm WARN     peer react@"^16.8.0 || ^17.0.1" from use-immer@0.6.0
npm WARN   node_modules/react-moralis/node_modules/use-immer
npm WARN     use-immer@"^0.6.0" from react-moralis@1.4.0
npm WARN   node_modules/react-moralis
added 1 package, removed 32 packages, and audited 2266 packages in 32s
```



Search or jump to...

/ Pull requests Issues Marketplace Explore

Notifications +

smartcontractkit / hardhat-starter-kit Public template

Watch 12

Fork 202

Star 574

<> Code Issues 10 Pull requests 2 Discussions Actions Projects Wiki Security Insights

main

3 branches

3 tags

Go to file

Add file

Code

Use this template

bobanm Fixed documentation on how to run staging tests (#104)

5bce50f 17 days ago 159 commits

contracts

Changed constant variable names to UPPERCASE

26 days ago

deploy

Removed unnecessary usage of Link token interface from RandomN...

last month

scripts

massive testing overhaul

5 months ago

tasks

Add VRF V2, Fuzz Testing, NatSpec, Coverage

4 months ago

About

A repo for boilerplate code for testing, deploying, and shipping chainlink solidity code.

javascript

chainlink

hardhat

Readme

MIT license

574 stars

BUAT NEW PROJECT

Buat project baru dengan clone github hardhat starter kit:

git clone https://github.com/smartcontractkit/hardhat-starter-kit

cd hardhat-starter-kit

INSTAL DEPENDENCY

Install Dependency dengan perintah "yarn" pada terminal

The screenshot shows the VS Code interface with the terminal tab selected. The terminal window displays the output of a `yarn install` command. The output shows the process of resolving and fetching packages, followed by several warning messages about peer dependency issues between `@chainlink/test-helpers`, `@krebernisak/ovm-plugins`, and `nomiclabs/hardhat-truffle5` packages across different versions (e.g., 0.0.15, 0.0.36, 2.0.3). The terminal also indicates network connection trouble and retrying.

```
yarn install v1.22.17
[1/4] 🏛️ Resolving packages...
[2/4] 🚚 Fetching packages...
warning Pattern ["ethereumjs-abi@git+https://github.com/ethereumjs/ethereumjs-abi.git"] is trying
"/Users/darma/Library/Caches/Yarn/v6/npm-ethereumjs-abi-0.6.8-ee3994657fa7a427238e6ba92a84d0b529b"
as pattern ["ethereumjs-abi@0.6.8","ethereumjs-abi@^0.6.8","ethereumjs-abi@0.6.8"]. This could re
, skipping.
warning Pattern ["@eth-optimism/solc@^0.6.12-alpha.1"] is trying to unpack in the same destination
/v6/npm-@eth-optimism-solc-v0-6-12-0.6.12-alpha.1-041876f83b34c6afe2f19dfe9626568df6ed8590-integr
-v0.6.12" as pattern ["@eth-optimism/solc-v0.6.12@npm:@eth-optimism/solc@^0.6.12-alpha.1"]. This c
ehavior, skipping.
info There appears to be trouble with your network connection. Retrying...
info There appears to be trouble with your network connection. Retrying...
[3/4] ⚒️ Linking dependencies...
warning "@chainlink/test-helpers > @krebernisak/ovm-plugins@0.0.15" has incorrect peer dependency
.
warning "@chainlink/test-helpers > @krebernisak/ovm-plugins > @nomiclabs/hardhat-truffle5@2.0.3" I
bs/hardhat-web3@"2.0.0".
warning "@chainlink/test-helpers > @krebernisak/ovm-plugins > @nomiclabs/hardhat-truffle5@2.0.3" I
0.0-beta.36".
warning "@chainlink/test-helpers > typechain > ts-essentials@7.0.3" has unmet peer dependency "typ
warning "@chainlink/test-helpers > @krebernisak/ovm-plugins > @nomiclabs/hardhat-truffle5 > @nomi
unmet peer dependency "web3@^1.2.1".
warning "@chainlink/test-helpers > @krebernisak/ovm-plugins > @nomiclabs/hardhat-truffle5 > @nomi
unmet peer dependency "web3-core-helpers@^1.2.1".
warning "@chainlink/test-helpers > @krebernisak/ovm-plugins > @nomiclabs/hardhat-truffle5 > @nomi
unmet peer dependency "web3-core-promievent@^1.2.1".
warning "@chainlink/test-helpers > @krebernisak/ovm-plugins > @nomiclabs/hardhat-truffle5 > @nomi
```

The screenshot shows the VS Code interface with the title bar "hardhat-starter-kit". The left sidebar displays the file structure under "OPEN EDITORS", with "test" selected. The main area shows the terminal output:

```
~ / Desktop/hardhat-starter-kit on main
yarn hardhat test
yarn run v1.22.17
$ /Users/darma/Desktop/hardhat-starter-kit/node_modules/.bin/hardhat test
Downloading compiler 0.8.7
Downloading compiler 0.4.24
@chainlink/token/contracts/v0.4/token/linkStandardToken.sol:27:5: Warning: Use of the "var _allowance = allowed[_from][msg.sender];"
^____^

@chainlink/token/contracts/v0.4/LinkToken.sol:15:3: Warning: Defining constructors as functions deprecated. Use "constructor(...) { ... }" instead.
function LinkToken()
^ (Relevant source part starts here and spans across multiple lines).

@chainlink/token/contracts/v0.4/ERC677Token.sol:21:5: Warning: Invoking events without Transfer(msg.sender, _to, _value, _data);
^____^

@chainlink/token/contracts/v0.4/token/linkBasicToken.sol:25:5: Warning: Invoking events Transfer(msg.sender, _to, _value);
^____^

@chainlink/token/contracts/v0.4/token/linkStandardToken.sol:35:5: Warning: Invoking events Transfer(_from, _to, _value);
^____^
```

TEST HADHAT

lakukan Uji Coba dengan perintah "yarn hardhat test" pada terminal. Setelah itu akan mendapatkan data akun yang akan di uji coba.

The screenshot shows a terminal window with the following output:

```
/Users/darma/.zprofile:18: no such file or directory
[1/1] ~/Desktop/hardhat-starter-kit [on]
yarn hardhat node
yarn run v1.22.17
$ /Users/darma/Desktop/hardhat-starter-kit/node_modules/.bin/hardhat node
Nothing to compile
Local network detected! Deploying mocks...
deploying "LinkToken" (tx: 0x90ca71286c245a64ce27f032d93F642f64180aa3 with 1279499 gas)
deploying "MockV3Aggregator" (tx: 0x73367e44aa67CE288F8367e1Bb143E90bb3F0512 with 569671 gas)
deploying "VRFCoordinatorV2Mock" (tx: 0x4a1df221679d2D9a65F0992F2272dE9f3c7fa6e0 with 1797707 gas)
deploying "MockOracle" (tx: 0x7c5bf13ac29f675421704C703E8D87F634fB0Fc9 with 1131081 gas)
Mocks Deployed!

You are deploying to a local network, you'll need to
Please run `yarn hardhat console` to interact with your contracts

deploying "PriceConsumerV3" (tx: 0x1d0187d17e35781100a9becA4E685f962f0cF6C9 with 245612 gas)
Run Price Feed contract with command:
yarn hardhat read-price-feed --contract 0xDc64a1

deploying "APIConsumer" (tx: 0xd3acc18d9eb5d1b1739d9d3abcBD16989F875707 with 1404981 gas)
Run API Consumer contract with following command:
yarn hardhat request-data --contract 0x5FC8d3269
```

DEPLOY PRICE CONSUMERV3

Lakukan Deploying pada harga konsumen dengan perintah "yarn hardhat node" pada terminal anda

MENAMPAKILKAN HARGA

Menampilkan harga dengan cara masukkan perintah "yarn hardhat console --network localhost" pada terminal. Setelah itu ikuti perintah gambar tersebut untuk menampilkan nilai



A screenshot of a Mac OS X desktop environment showing a code editor window for a Hardhat project. The title bar reads "readPrice.js — hardhat-starter-kit". The code editor shows a file named "readPrice.js" with the following content:

```
scripts > JS readPrice.js > readPrice > price
6  async function readPrice() {
7    const priceConsumerV3 = await ethers.getContract("PriceConsumerV3")
8    const price = await priceConsumerV3.getLatestPrice()
9    console.log(price.toString())
10   }
11
12  readPrice()
13  .then(() => process.exit(0))
14  .catch(error) => {
15    console.error(error)
16    process.exit(1)
17  }
```

The terminal tab at the bottom shows the following command-line interaction:

```
yarn console --network localhost
yarn run v1.22.17
error Command "console" not found.
info Visit https://yarnpkg.com/en/docs/cli/run for documentation about this command.
yarn hardhat console --network localhost
yarn run v1.22.17
$ /Users/darma/Desktop/hardhat-starter-kit/node_modules/.bin/hardhat console --network localhost
Welcome to Node.js v16.15.1.
Type ".help" for more information.
> const priceConsumerV3 = await ethers.getContract("PriceConsumerV3")
undefined
> const price = await priceConsumerV3.getLatestPrice
undefined
> await priceConsumerV3.getLatestPrice()
BigNumber { _hex: '0x0ad78ebc5ac620000', _isBigNumber: true }
> (await priceConsumerV3.getLatestPrice()).toString()
'20000000000000000000'
>
```

At the bottom right of the terminal window, there is a small watermark that says "Blockchain".

The screenshot shows the VS Code interface with the following details:

- Explorer:** Shows the project structure under "HARDHAT-STARTER-KIT".
- Terminal:** Displays the output of the "yarn hardhat deploy" command.

```
:443:23)
    at DeploymentsManager.loadDeployments (/Users/darma/Desktop/hardhat-starter-kit/node_modules/hardhat-deployer.ts:619:17)
error Command failed with exit code 1.
info Visit https://yarnpkg.com/en/docs/cli/run for documentation about this command.
yarn hardhat deploy
yarn run v1.22.17
$ /Users/darma/Desktop/hardhat-starter-kit/node_modules/.bin/hardhat deploy
Nothing to compile
Local network detected! Deploying mocks...
deploying "LinkToken" (tx: 0x90ca71286c245a64ceaaafde24331fe80e7daf4cf71e7ad32fcf620ae66a943c8)...: deployed
7f032d93F642f64180aa3 with 1279499 gas
deploying "MockV3Aggregator" (tx: 0x73367e44aa675ec6c93d61558f83d0f310bb095b07043a5cef3310614d95f406)...: deployed
CE288F8367e1Bb143E90bb3F0512 with 569671 gas
deploying "VRFCoordinatorV2Mock" (tx: 0x4a1df22fde7a09f1539013ccece1869430a238b0a6213bda3ceab8d866121a26)...: deployed
679d2D9a65F0992F2272dE9f3c7fa6e0 with 1797707 gas
deploying "MockOracle" (tx: 0x7c5bf13ac29f67542f81da4712d1a199f079ec7e4e32f35cecab7fc52885da82)...: deployed
704C703E8D87F634fB0Fc9 with 1131081 gas
Mocks Deployed!

You are deploying to a local network, you'll need a local network running to interact
Please run `yarn hardhat console` to interact with the deployed smart contracts!

deploying "PriceConsumerV3" (tx: 0x1d0187d17e357248aa44108bc810533839e20a73625698ddbba1b3707c428a2a)...: deployed
81100a9becA4E685f962f0cF6C9 with 245612 gas
Run Price Feed contract with command:
yarn hardhat read-price-feed --contract 0xDc64a140Aa3E981100a9becA4E685f962f0cF6C9 --network localhost

deploying "APIConsumer" (tx: 0xd3acc18d9eb5d1b177e8338dc15e772e49ff29a965968183e07e19bc98285301)...: deployed
39d9d3abcBD16989F875707 with 1404981 gas
Run API Consumer contract with following command:
yarn hardhat request-data --contract 0x5FC8d32690cc91D4c39d9d3abcBD16989F875707 --network localhost

deploying "RandomNumberConsumerV2" (tx: 0xf88f155ba41e6a4d95c9e296ed49147888b017dd96004d4a17ab78e0b1b82d9b).A0a67DB372996a5FaB50D91eAA73d2eBe6 with 564345 gas
Then run RandomNumberConsumer contract with the following command
yarn hardhat request-random-number --contract 0x2279B7A0a67DB372996a5FaB50D91eAA73d2eBe6 --network localhost

deploying "KeepersCounter" (tx: 0x86836f42077c9d4e1700a35fb27a757e8a6c1d724c4c96fa434146e061677f4)...: deployed
079BF849Dc5567aDC3F2FdC318 with 448166 gas
Head to https://keepers.chain.link/ to register your contract for upkeeps. Then run the following command to es:
yarn hardhat read-keepers-counter --contract 0x8A791620dd6260079BF849Dc5567aDC3F2FdC318 --network localhost
```

DEPLOYING HARDHAT

HARDHAT

ERC20

Get Started





```
iTerm2 Shell Edit View Session Scripts Profiles Toolbelt Window  
darma@Darma-MacBook-Pro:~/Desktop/hardhat-erc20  
~/desktop  
mkdir hardhat-erc20  
~/desktop  
cd /Users/darma/Desktop/hardhat-erc20  
~/Desktop/hardhat-erc20  
yarn add --dev hardhat  
yarn add v1.22.17  
warning package-lock.json found. Your project contains lock files generated by  
s advised not to mix package managers in order to avoid resolution inconsisten  
d lock files. To clear this warning, remove package-lock.json.  
[1/4] 🔎 Resolving packages...  
warning ethereum-waffle > @ethereum-waffle/provider > ganache-core > web3-provi  
.4.2: Deprecated in favor of '@metamask/eth-sig-util'  
[2/4] 🚀 Fetching packages...  
[3/4] 🔁 Linking dependencies...  
warning " > bootstrap@5.1.3" has unmet peer dependency "@popperjs/core@^2.10.2"  
warning "react-moralis > use-immer@0.6.0" has incorrect peer dependency "react@  
warning "ethereum-waffle > @ethereum-waffle/compiler > typechain > ts-essential  
endency "typescript@>=3.7.0".  
[4/4] 🏗️ Building fresh packages...  
success Saved lockfile.  
success Saved 3 new dependencies.  
info Direct dependencies  
└── hardhat@2.10.0  
info All dependencies  
└── hardhat@2.10.0  
   └── mocha@10.0.0  
      └── workerpool@2.1
```

CREATE NEW PROJECT

01.

Create Folder

mkdir hardhat-erc20

02.

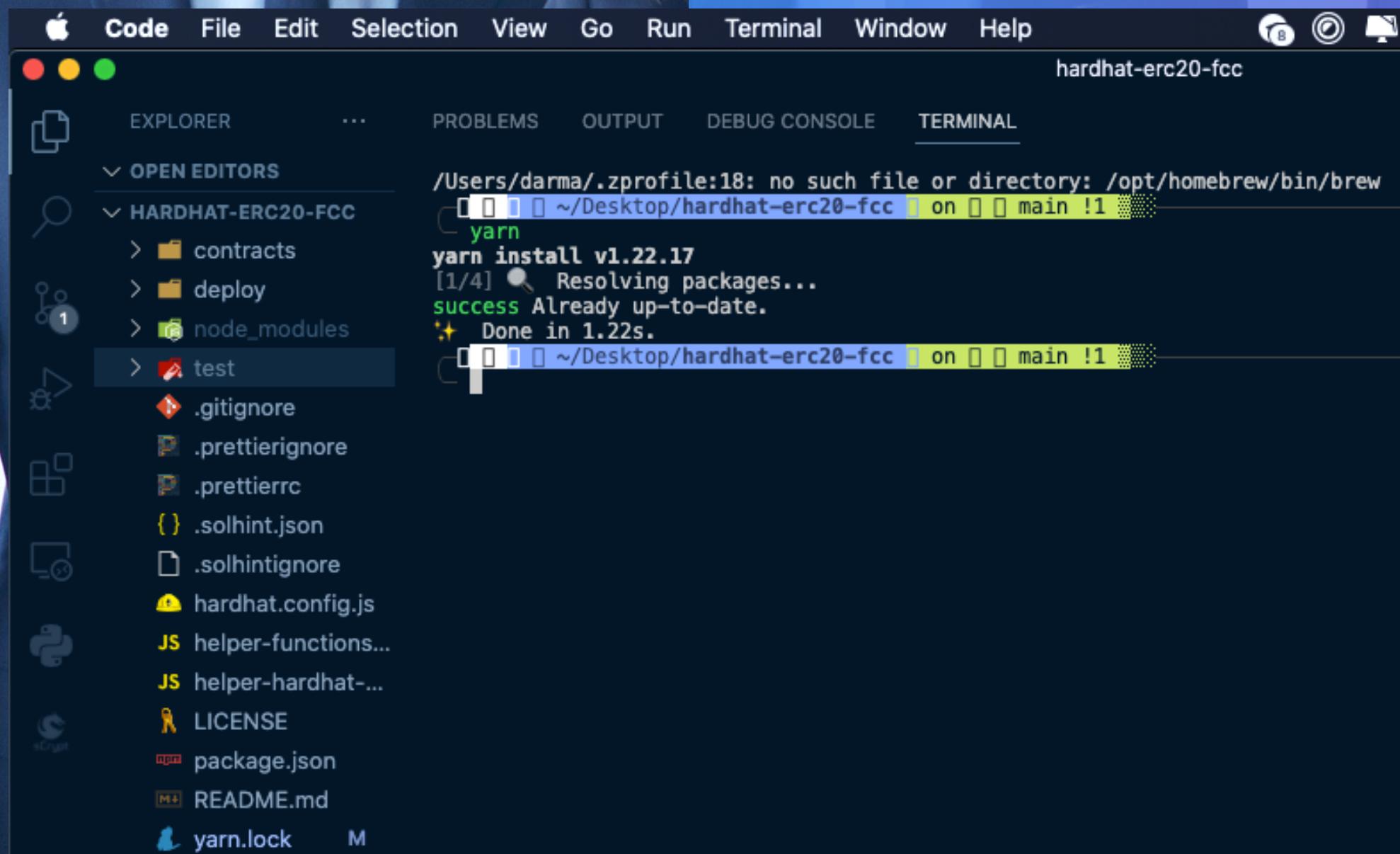
Framework hardhat

yarn add --dev hardhat

Install Open Zeppelin

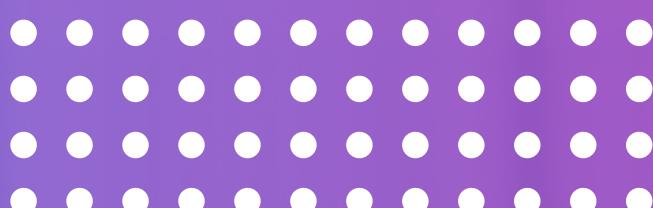
Instal dengan cara "npm install @openzeppelin/contract" pada terminal anda

```
~/.nvm/versions/node/v14.15.4/bin/npx hardhat run scripts/deploy.js --network mainnet
[npm] npm install @openzeppelin/contracts
[npm] notice New minor version of npm available! 8.11.0 → 8.13.2
[npm] notice Changelog: https://github.com/npm/cli/releases/tag/v8.13.2
[npm] notice Run npm install -g npm@8.13.2 to update!
[npm] notice
```



```
hardhat-erc20-fcc
Code File Edit Selection View Go Run Terminal Window Help
EXPLORER PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
/Users/darma/.zprofile:18: no such file or directory: /opt/homebrew/bin/brew
yarn install v1.22.17
[1/4] Resolving packages...
success Already up-to-date.
Done in 1.22s.
```

Install Dependency



OpenZeppelin-Contract



```
OurToken.sol — hardhat-erc20-fcc
contracts > OurToken.sol
1 // contracts/OurToken.sol
2 // SPDX-License-Identifier: MIT
3 pragma solidity ^0.8.7;
4
5 import "@openzeppelin/contracts/token/ERC20/ERC20.sol";
6
7 contract OurToken is ERC20 {
8     constructor(uint256 initialSupply) ERC20("OurToken",
9         _mint(msg.sender, initialSupply);
10    }
11 }
12

ManualToken.sol 8
contracts > ManualToken.sol
1 // SPDX-License-Identifier: MIT
2 pragma solidity ^0.8.7;
3
4 interface tokenRecipient {
5     function receiveApproval(
6         address _from,
7         uint256 _value,
8         address _token,
9         bytes calldata _extraData
10    ) external;
11 }
12
13 contract TokenERC20 {
14     // Public variables of the token
15     string public name;
16     string public symbol;
17     uint8 public decimals = 18;
18     // 18 decimals is the strongly suggested default,
19     uint256 public totalSupply;
20
21     // This creates an array with all balances
22     mapping(address => uint256) public balanceOf;
23     mapping(address => mapping(address => uint256)) p
24
25     // This generates a public event on the blockchain
26     event Transfer(address indexed from, address inde
27
28     // This generates a public event on the blockchain
29 }
```

PROBLEMS 8 OUTPUT DEBUG CONSOLE TERMINAL

/Users/darma/.zprofile:18: no such file or directory: /opt/homebrew/bin/brew

Ln 1, Col 1 Spaces: 2 UTF-8 LF Solidity ⚡ Go Live 🛡️ ✅ Pret

Create code with openzeppelin

Blockchain



Deploying Token

The screenshot shows a code editor interface with a dark theme. The main area displays a JavaScript file named `01-deploy-token.js`. The code is used to deploy a contract named `OurToken` with an initial supply of `INITIAL_SUPPLY`. It also includes logic to verify the deployment on a live network if the environment variable `ETHERSCAN_API_KEY` is set. The code editor has tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, and TERMINAL. The TERMINAL tab shows the command `yarn hardhat deploy` being run, followed by the output of the deployment process. The sidebar on the left shows the project structure, including files like `ManualToken.sol`, `OurToken.sol`, and configuration files like `hardhat.config.js` and `helper-functions.js`.

```
const { getNamedAccounts, deployments, network } = require("hardhat")
const {
  developmentChains,
  INITIAL_SUPPLY,
} = require("../helper-hardhat-config")
const { verify } = require("../helper-functions")

module.exports = async ({ getNamedAccounts, deployments }) => {
  const { deploy, log } = deployments
  const { deployer } = await getNamedAccounts()
  const ourToken = await deploy("OurToken", {
    from: deployer,
    args: [INITIAL_SUPPLY],
    log: true,
    // we need to wait if on a live network so we can verify properly
    waitConfirmations: network.config.blockConfirmations || 1,
  })
  log(`ourToken deployed at ${ourToken.address}`)

  if (
    !developmentChains.includes(network.name) &&
    process.env.ETHERSCAN_API_KEY
  ) {
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
yarn hardhat deploy
yarn run v1.22.17
$ /Users/darma/Desktop/hardhat-erc20-fcc/node_modules/.bin/hardhat deploy
Compiling 6 files with 0.8.7
Solidity compilation finished successfully
deploying "OurToken" (tx: 0x167069b2c1ff4ced4891bcc9964738e61f35af2c80ac024bd4a5aee3b3e6eb0)...: deployed at 0x5FbDB2315678afecb367f032d93F642f64180aa3 with 1225998 gas
ourToken deployed at 0x5FbDB2315678afecb367f032d93F642f64180aa3
✨ Done in 8.36s.
```

Ln 1, Col 1 Spaces: 2 UTF-8 LF {} JavaScript Go Live Prettier



Deploying Token

```
~/Desktop/hardhat-erc20-fcc on main !1 ?1
yarn hardhat deploy
yarn run v1.22.17
$ /Users/darma/Desktop/hardhat-erc20-fcc/node_modules/.bin/hardhat deploy
Compiling 6 files with 0.8.7
Solidity compilation finished successfully
deploying "OurToken" (tx: 0x167069b2c1ff48ced4891bcc9964738e61f35af2c80ac024bd4a5aee3b3e6eb0)...
f032d93F642f64180aa3 with 1225998 gas
ourToken deployed at 0x5FbDB2315678afecb367f032d93F642f64180aa3
+ Done in 8.36s.
```

LESSON 13 HARDHAT DEFI

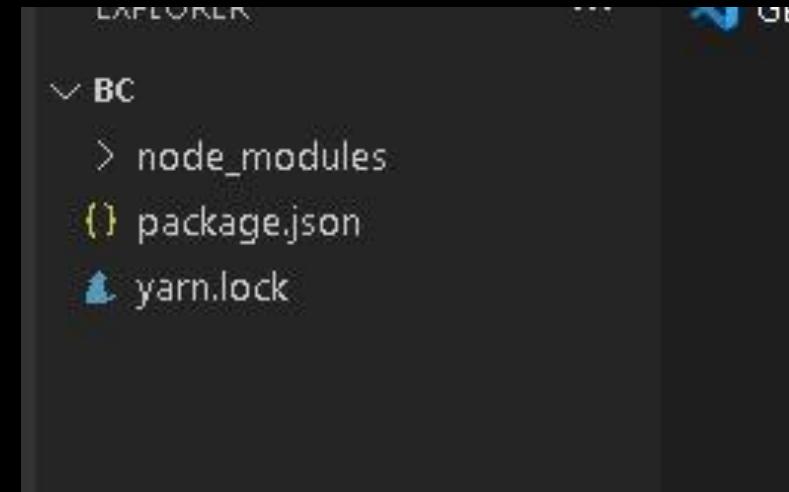
WHAT IS DEFI?

- defi merupakan salah satu smartcontract terbaik. defi merupakan menuju industri yang berefek besar, dan sangat cepat karena berapa banyak decentralized yang bagus daripada keuangan pada centralized, karena akan menjauhi protokol centralized dan kepercayaan dalam memberikan akses pada market, smartcontrant akan membuat uang anda aman.

WHAT IS AAVE?

- ▶ Aave adalah adalah protocol simpan peminjaman yang memungkinkan kita untuk meminjam dan meminjamkan mata uang kripto, jadi dapat menyimpan token sebagai collateral, ini mirip seperti simpan uang di bank lalu yields awal pada pengguna lain
- ▶ Aave team tidak pernah menyentuk uang pengguna. Tidak ada yang pernah menyentuh uang, ini semua adalah smart contract, ini semua merupakan programmatic code

PROGMATIC BORROWING & LENDING



- ▶ Untuk programnya menggunakan javascript, sebelum melakukan programmatic harus install dulu yarn untuk install hardhat javascript

```
C:\WINDOWS\system32\cmd.exe - "C:\Program Files\nodejs\nod
Microsoft Windows [Version 10.0.19044.1766]
(c) Microsoft Corporation. All rights reserved.

D:\Tugas\BC>yarn add --dev hardhat
yarn add v1.22.19
Info No lockfile found.
[1/4] Resolving packages...
  @ethersproject/logger@^5.6.0
```

- ▶ Lakukan running hardhat dengan cara yarn hardhat lalu pilih empty hardhat.config.js

```
PS D:\Tugas\BC> yarn hardhat
```

```
? What do you want to do? ...
Create a JavaScript project
Create a TypeScript project
> Create an empty hardhat.config.js
Quit
```

RUNNING HARDHAT

Setelah membuat empty hardhat.config.js selanjutnya menambahkan package

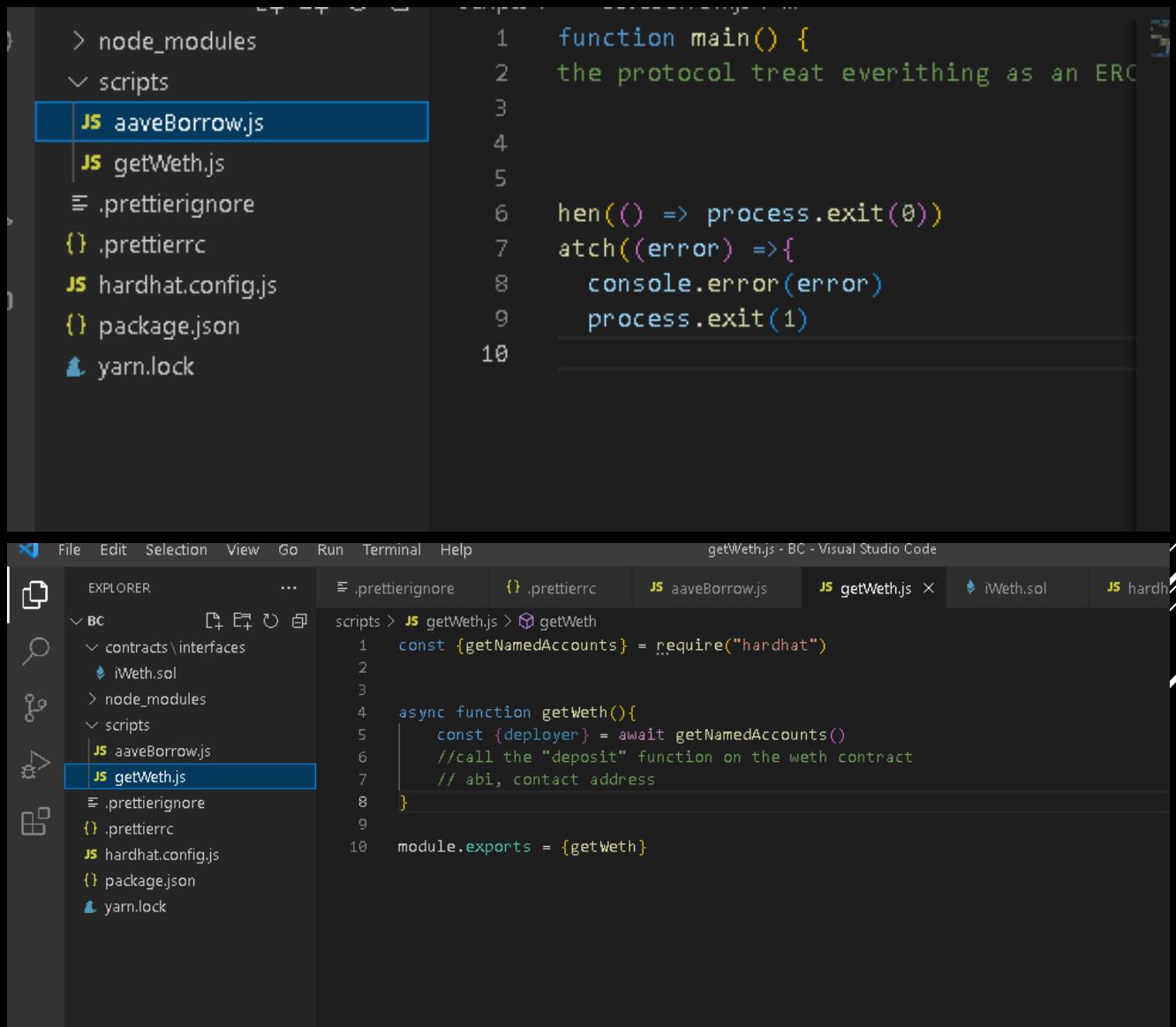
```
PS D:\Tugas\BC> yarn add --dev @nomiclabs/hardhat-ethers@npm:hardhat-deploy-ethers ethers @nomiclabs/hardhat-etherscan @nomiclabs/hardhat-waffle chai ethereum-waffle hardhat hardhat-contract-sizer hardhat-deploy hardhat-gas-reporter prettier prettier-plugin-solidity solhint solidity-coverage dotenv
yarn add v1.22.19
Warning package.json: No license field
Warning No license field
[1/4] Resolving packages...
```

COPY FILE PRETTIER DI SECURITY HARDHAT

The screenshot shows a GitHub repository page for `PatrickAlphaC/hardhat-security-fcc`. The repository is public and has 1 branch and 0 tags. The `Code` tab is selected. The commit history shows a merge pull request from `0xJepsen/patch-1` and several initial commits related to setting up prettier and solhint. The file structure on the left includes `.prettierignore`, `.prettierrc`, `contracts`, `images`, `.gitignore`, `.npmignore`, and configuration files for `.prettierignore`, `.prettierrc`, `.solhint.json`, and `.solhintignore`.

Commit	Message	Date
Merge pull request #5 from 0xJepsen/patch-1	nah	3 days ago
contracts	init commit	20 days ago
images	fix	4 months ago
.gitignore	init commit	20 days ago
.npmignore	init commit	4 months ago
.prettierignore	init commit	4 months ago
.prettierrc	init commit	4 months ago
.solhint.json	init commit	4 months ago
.solhintignore	init commit	4 months ago

MEMBUAT SCRIPT



The screenshot shows a Visual Studio Code interface with a dark theme. The left sidebar displays a file tree with the following structure:

- > node_modules
- scripts
 - JS aaveBorrow.js**
 - JS getWeth.js
 - .prettiignore
 - .prettierrc
 - hardhat.config.js
 - package.json
 - yarn.lock

The main editor area shows the content of the 'getWeth.js' file:

```
1 function main() {
2     the protocol treat everithing as an ERC
3
4
5
6     hen(() => process.exit(0))
7     atch((error) =>{
8         console.error(error)
9         process.exit(1)
10}
```

The bottom status bar shows the current file names: .prettiignore, .prettierrc, JS aaveBorrow.js, JS getWeth.js, iWeth.sol, and hardh.

The screenshot shows a dark-themed code editor interface with several tabs and files visible.

EXPLORER tab is open, showing the project structure:

- contracts > interfaces > **iWeth.sol**
- > node_modules
- < contracts\interfaces
- < .prettierignore
- < .prettierrc
- iWeth.sol**
- aaveBorrow.js**
- getWeth.js**
- < package.json
- < hardhat.config.js
- < .yaml.lock

The **iWeth.sol** file is selected and its content is displayed:

```
1 pragma solidity ^0.4.19;
2
3 interface IWeth {
4     function allowance(address owner, address spender) external view returns (uint256 remaining);
5
6     function approve(address spender, uint256 value) external returns (bool success);
7
8     function balanceOf(address owner) external view returns (uint256 balance);
9
10    function decimals() external view returns (uint8 decimalPlaces);
11
12    function name() external view returns (string memory tokenName);
13
14    function symbol() external view returns (string memory tokenSymbol);
15
16    function totalSupply() external view returns (uint256 totalTokensIssued);
17
18    function transfer(address to, uint256 value) external returns (bool success);
19
20    function transferFrom(
21        address from,
22        address to,
23        uint256 value
24    ) external returns (bool success);
```

The **hardhat.config.js** file is also partially visible at the bottom:

```
/** @type import('hardhat/config').HardhatUserConfig */
module.exports = {
  solidity: "0.8.9",
}, {
  solidity: "0.4.19"
};
```

COMPILE

The screenshot shows a terminal window with a dark theme. On the left, there's a circular icon containing the word "COMPILE". The terminal displays the following command and its output:

```
PS D:\Tugas\BC> yarn hardhat compile
yarn run v1.22.19
warning package.json: No license field
$ D:\Tugas\BC\node_modules\.bin\hardhat compile
✖ Help us improve Hardhat with anonymous crash reports & basic usage data? (Y/n) • y
(node:13876) ExperimentalWarning: stream/web is an experimental feature. This feature could change at any time
(Use `node --trace-warnings ...` to show where the warning was created)
Downloading compiler 0.4.19
Compiled 1 Solidity file successfully
Done in 26.41s.
PS D:\Tugas\BC>
```

The terminal also shows the contents of the `hardhat.config.js` file in the background:

```
JS hardhat.config.js > ...
1  * @type {import('hardhat/config').HardhatUserConfig}
2  module.exports = {
3    solidity: '0.8.9', solidity: '0.4.19'
4  };
5
```



```
scripts > JS getWeth.js > getWeth
1  const {getNamedAccounts, ethers} = require("hardhat")
2
3  const AMOUNT = ethers.utils.parseEther("0.02")
4
5  async function getWeth(){
6      const [deployer] = await getNamedAccounts()
7      //call the "deposit" function on the weth contract
8      // abi, contact address
9      // 0xC02aaa39b223FE8D0A0e5C4F27eAD9083C756Cc2
10
11     const iWeth = await ethers.getContractAt("IWETH", "0xC02aaa39b223FE8D0A0e5C4F27eAD9083C756Cc2", deployer)
12     const tx = await iWeth.deposit({value: AMOUNT})
13     await tx.wait(1)
14     const wethBalance = await iWeth.balanceOf(deployer)
15     console.log(`Got ${wethBalance.toString()} WETH`)
16
17 }
18
19 module.exports = {getWeth}
```

UPDATE GETWETH DENGAN MENAMBAHKAN CONTRACTS

KESIMPULAN

- ▶ Pada Lesson 13 ini membahas tentang DeFi yang merupakan salah satu smartcontract yang fungsinya mirip dengan bank umum. DeFi ini memiliki banyak token yang dapat dipake untuk simpan dan pinjam mata uang kripto. Salah satu token popular adalah Aave yang pembuatannya memerlukan hardhat yang basis nya javascript.

LESSON 14 HARDHAT NFTS



WHAT IS NFT?

- ▶ NFT merupakan toket yang memiliki standard ERC 721 yang dibuat dari Ethereum platform. NFT atau non Fungible Tokeen merupakan token standar mirip dengan ERC 20. ERC 20 contohnya adalah link, marker, Aave Maker.

- ▶ ERC-721 merupakan standar dari NFT. ERC-721 merupakan basic dari semua standar yang ada.
- ▶ Perbedaan ERX-721 dengan ERC 20 adalah ERC20 lebih simple dalam mapping address, sedangkan ERX720 memiliki token ID yang unik .
- ▶ Token ID adalah unique owner. Pada token ID memiliki token URI.

```
// Mapping from token ID to owner address  
mapping (uint256 => address) private _owners;
```

ERC-721

```
contract ERC20 is Context, IERC20 {  
    mapping (address => uint256) private _balances;
```

- ▶ Ketika assetnya Unique dan dapat di visualisasi dan dapat perlihatkan, jadi perlu mendefinisikan atribut dari sebuah objek.
- ▶ Pada karya seni kita perlu mendefinisikan. Contoh pada karakter game perlu memerlukan statn pada NFT. Ini merupakan meta data dan token URI muncul.
- ▶ Ethereum memiliki gas yang cukup mahal, terus bagaimana artis meningumpang gambar-gambar? Dan ini merupakan bagian dari chain?
- ▶ Jawabannya adalah NFT, artis dapat menyebarkan karyanya pada Ethereum chain

NFT



EXPLORER

HARDHAT-NFT-FOC

- > contracts
- deploy
- JS 00-deploy-mocks.js**
- JS 01-deploy-basic-nft.js**
- JS 02-deploy-dynamic-svg-nft.js**
- JS 03-deploy-random-ipfs-nft.js**
- JS 04-mint.js**
- > images
- > test
- > utils
- .env.example**
- .gitignore**
- .npmignore**
- .prettierignore**
- ...

01-deploy-basic-nft.js

```
deploy > JS 01-deploy-basic-nft.js > <unknown> > exports > basicNft
1  const { network } = require("hardhat")
2  const { developmentChains } = require("../helper-hardhat-config")
3  const { verify } = require("../utils/verify")
4
5  module.exports = async ({ getNamedAccounts, deployments }) => {
6      const { deploy, log } = deployments
7      const { deployer } = await getNamedAccounts()
8
9      log("-----")
10     arguments = []
11     const basicNft = await deploy("BasicNft", [
12         {
13             from: deployer,
14             args: arguments,
15             log: true,
16             waitConfirmations: network.config.blockConfirmations || 1,
17         }
18     ])
19 }
```

CODE OVERVIEW

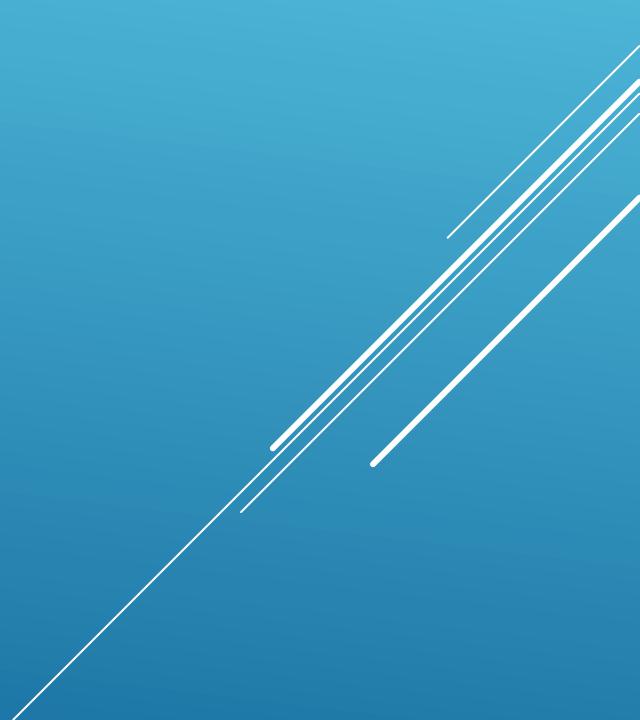
LESSON 15 FULL STACK NFT MARKETPLACE



CONTRACTS

```
◆ NftMarketplace.sol ×

hardhat-nft-marketplace-fcc > contracts > ◆ NftMarketplace.sol
1 // SPDX-License-Identifier: MIT
2 pragma solidity ^0.8.7;
3
4 import "@openzeppelin/contracts/token/ERC721/IERC721.sol";
5 import "@openzeppelin/contracts/security/ReentrancyGuard.sol";
6
7 // Check out https://github.com/Fantom-foundation/Artion-Contracts/blob/5c90d2bc04
8 // For a full decentralized nft marketplace
9
10 error PriceNotMet(address nftAddress, uint256 tokenId, uint256 price);
11 error ItemNotForSale(address nftAddress, uint256 tokenId);
12 error NotListed(address nftAddress, uint256 tokenId);
13 error AlreadyListed(address nftAddress, uint256 tokenId);
14 error NoProceeds();
15 error NotOwner();
16 error NotApprovedForMarketplace();
17 error PriceMustBeAboveZero();
18
19 contract NftMarketplace is ReentrancyGuard {
20     struct Listing {
21         uint256 price;
22         address seller;
23     }
```

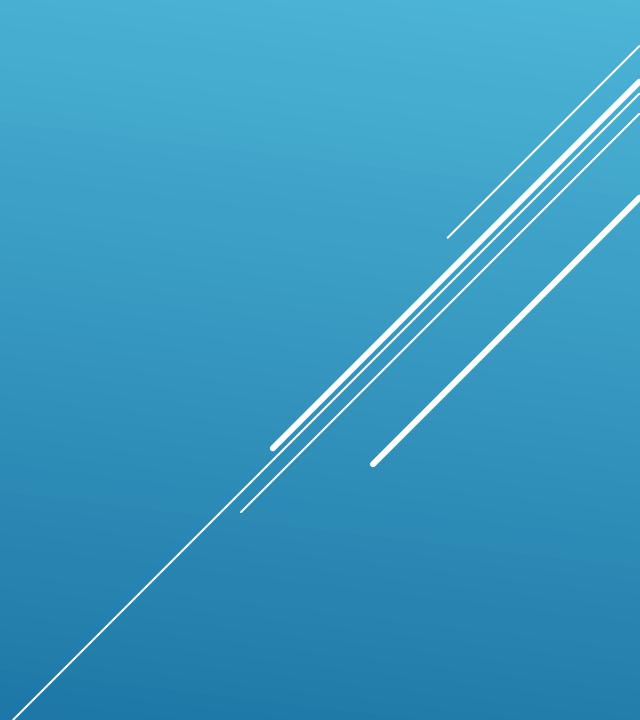


```
... NftMarketplace.sol JS 01-deploy-nft-marketplace.js ...
D hardhat-nft-marketplace-fcc > deploy > JS 01-deploy-nft-marketplace.js > ...
e-fcc
1 const { network } = require("hardhat")
2 const { developmentChains, VERIFICATION_BLOCK_CONFIRMATIONS } = require("../helper-hardhat-config")
3 const { verify } = require("../utils/verify")
4
5 module.exports = async ({ getNamedAccounts, deployments }) => {
6   const { deploy, log } = deployments
7   const { deployer } = await getNamedAccounts()
8   const waitBlockConfirmations = developmentChains.includes(network.name)
9     ? 1
10     : VERIFICATION_BLOCK_CONFIRMATIONS
11
12 log("-----")
13 const arguments = []
14 const nftMarketplace = await deploy("NftMarketplace", {
15   from: deployer,
16   args: arguments,
17   log: true,
18   waitConfirmations: waitBlockConfirmations,
19 })
20
21 // Verify the deployment
22 if (!developmentChains.includes(network.name) && process.env.ETHERSCAN_API_KEY) {
23   log("Verifying...")
24   await verify(nftMarketplace.address, arguments)
}
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE powershell
```

NFT MARKETPLACE DEPLOY

```
hardhat-nft-marketplace-fcc > deploy > JS 02-deploy-basic-nfts.js ...  
1 const { network } = require("hardhat")  
2 const { developmentChains, VERIFICATION_BLOCK_CONFIRMATIONS } = require("../helper-hardhat-config")  
3 const { verify } = require("../utils/verify")  
4  
5 module.exports = async ({ getNamedAccounts, deployments }) => {  
6   const { deploy, log } = deployments  
7   const { deployer } = await getNamedAccounts()  
8   const waitBlockConfirmations = developmentChains.includes(network.name)  
9     ? 1  
10    : VERIFICATION_BLOCK_CONFIRMATIONS  
11  
12   log("-----")  
13   const args = []  
14   const basicNft = await deploy("BasicNft", {  
15     from: deployer,  
16     args: args,  
17     log: true,  
18     waitConfirmations: waitBlockConfirmations,  
19   })  
20  
21   const basicNftTwo = await deploy("BasicNftTwo", {  
22     from: deployer,  
23     args: args,  
24     log: true,
```

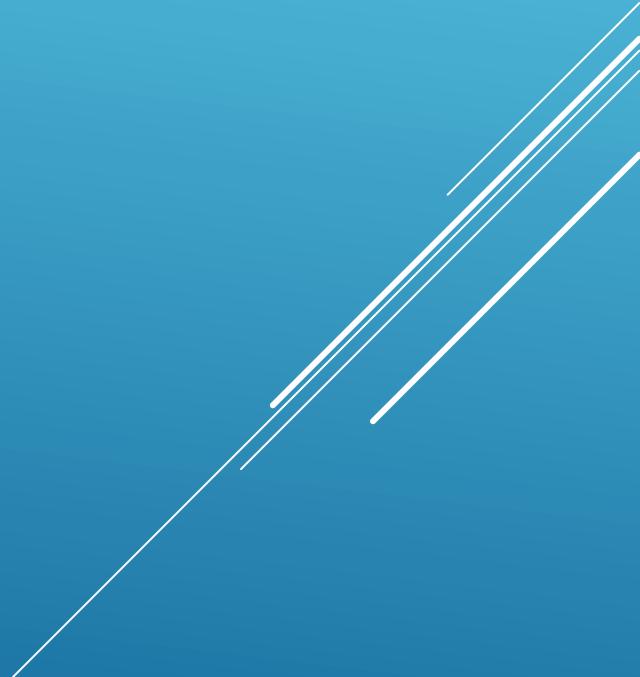
NFT BASIC DEPLOY



```
◆ NftMarketplace.sol   JS 03-update-front-end.js ×
hardhat-nft-marketplace-fcc > deploy > JS 03-update-front-end.js > ...
1  const {
2    frontEndContractsFile,
3    frontEndContractsFile2,
4    frontEndAbiLocation,
5    frontEndAbiLocation2,
6  } = require("../helper-hardhat-config")
7  require("dotenv").config()
8  const fs = require("fs")
9  const { network } = require("hardhat")
10
11 module.exports = async () => {
12   if (process.env.UPDATE_FRONT_END) {
13     console.log("Writing to front end...")
14     await updateContractAddresses()
15     await updateAbi()
16     console.log("Front end written!")
17   }
18 }
19
20 async function updateAbi() {
21   const nftMarketplace = await ethers.getContract("NftMarketplace")
22   fs.writeFileSync(
23     `${frontEndAbiLocation}NftMarketplace.json`,
24     nftMarketplace.interface.format(ethers.utils.FormatTypes.json)
25   )
}
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

UPDATE FRONT END



```
❖ NftMarketplace.sol      JS hardhat.config.js ×

hardhat-nft-marketplace-fcc > JS hardhat.config.js > ...
1  require("@nomiclabs/hardhat-waffle")
2  require("@nomiclabs/hardhat-etherscan")
3  require("hardhat-deploy")
4  require("solidity-coverage")
5  require("hardhat-gas-reporter")
6  require("hardhat-contract-sizer")
7  require("dotenv").config()
8
9  /**
10   * @type import('hardhat/config').HardhatUserConfig
11   */
12
13 const MAINNET_RPC_URL =
14   process.env.MAINNET_RPC_URL ||
15   process.env.ALCHEMY_MAINNET_RPC_URL ||
16   "https://eth-mainnet.alchemyapi.io/v2/your-api-key"
17 const RINKEBY_RPC_URL =
18   process.env.RINKEBY_RPC_URL || "https://eth-rinkeby.alchemyapi.io/v2/your-api-key"
19 const KOVAN_RPC_URL =
20   process.env.KOVAN_RPC_URL || "https://eth-kovan.alchemyapi.io/v2/your-api-key"
21 const POLYGON_MAINNET_RPC_URL =
22   process.env.POLYGON_MAINNET_RPC_URL || "https://polygon-mainnet.alchemyapi.io/v2/your-api-key"
23 const PRIVATE_KEY = process.env.PRIVATE_KEY || "0x"
```

HARDHAT CONFIG

LESSON 16: HARDHAT UPGRADES

Delegate Call

FILE EXPLORERS

Workspaces Workspace

contracts scripts tests artifacts .deps README.txt DelegateCallExample.sol

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.13;

// NOTE: Deploy this contract first
contract B {
    // NOTE: storage layout must be the same as contract A
    uint public num;
    address public sender;
    uint public value;

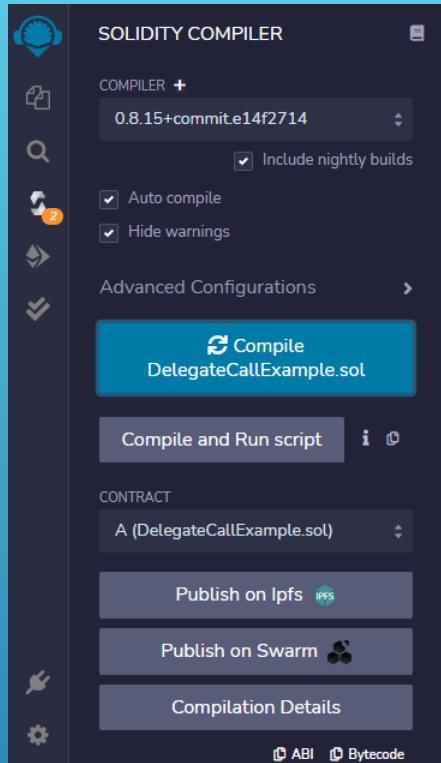
    function setVars(uint _num) public payable {
        num = _num;
        sender = msg.sender;
        value = msg.value;
    }
}

contract A {
    uint public num;
    address public sender;
    uint public value;

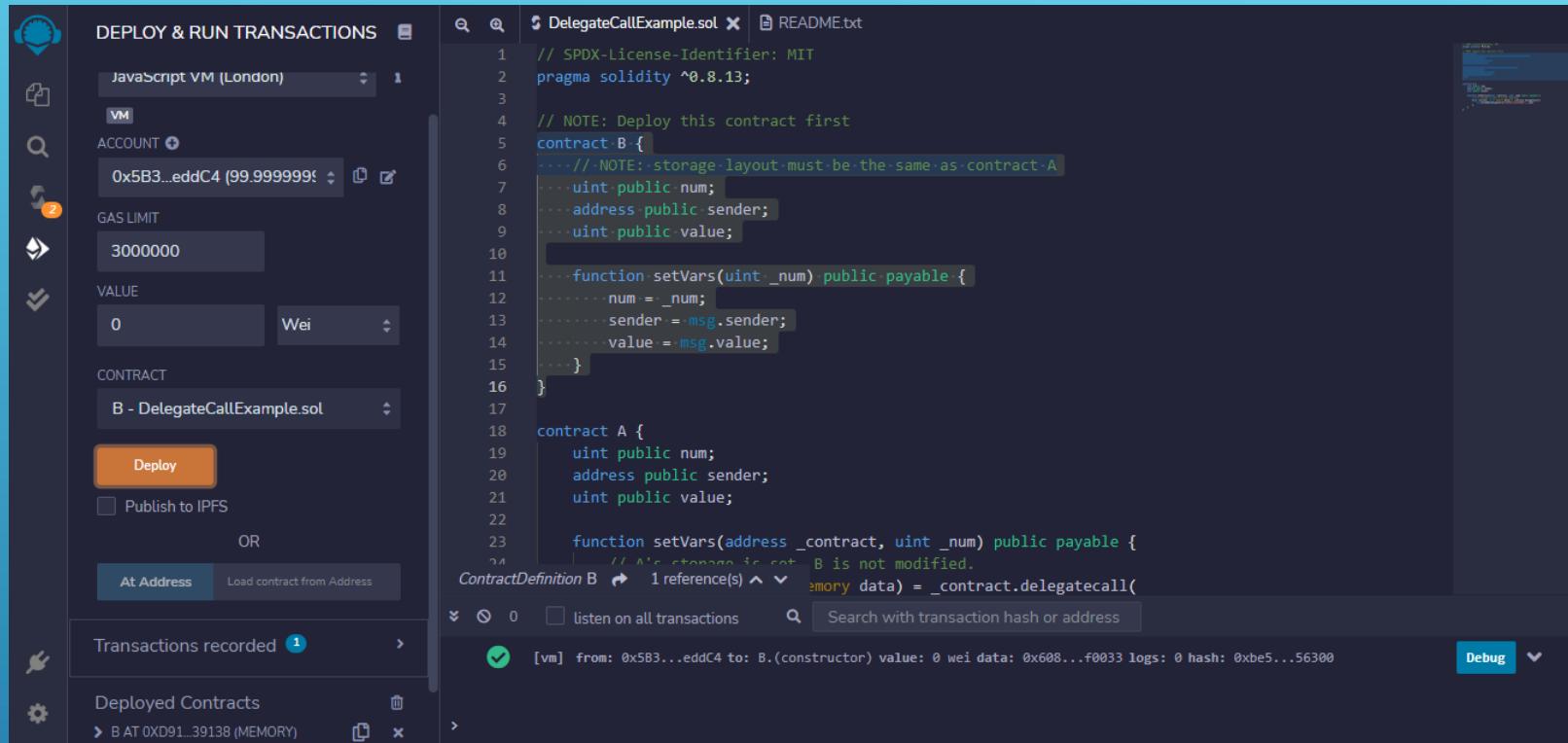
    function setVars(address _contract, uint _num) public payable {
        // A's storage is set, B is not modified.
        (bool success, bytes memory data) = _contract.delegatecall(
            abi.encodeWithSignature("setVars(uint)", _num)
        );
    }
}
```

B

DELEGATECALLEXAMPLE.SOL
DENGAN ISI SEPERTI DI LINK
[HTTPS://SOLIDITY-BY-
EXAMPLE.ORG/DELEGATECALL](https://solidity-by-example.org/delegatecall)



COMPILE FILE
DELEGATECALLEXAMPLE.SOL
DENGAN VERSI 0.8.13 KEATAS



The screenshot shows the Truffle UI interface for deploying a Solidity contract. On the left, the 'DEPLOY & RUN TRANSACTIONS' sidebar is visible, featuring a JavaScript VM (London) selected, an account of 0x5B3...eddC4 with a balance of 99.99999999999999 Wei, a gas limit of 3000000, and a value of 0 Wei. The 'CONTRACT' dropdown is set to 'B - DelegateCallExample.sol'. A prominent orange 'Deploy' button is centered below the dropdown. Below it, there's an option to 'Publish to IPFS' and two deployment methods: 'At Address' (selected) and 'Load contract from Address'. The main area displays the Solidity code for 'DelegateCallExample.sol' and 'README.txt'. The code defines two contracts, A and B. Contract A has public variables num, sender, and value. Contract B has a constructor and a function 'setVars' that delegates calls to Contract A. A note in the code states: 'NOTE: Deploy this contract first' and 'NOTE: storage.layout.must.be.the.same.as.contract.A'. The right side of the interface shows a transaction history with one entry: '[vmm] from: 0x5B3...eddC4 to: B.(constructor) value: 0 wei data: 0x608...f0033 logs: 0 hash: 0xbe5...56300'. A 'Debug' button is located at the bottom right of the transaction list.

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.13;

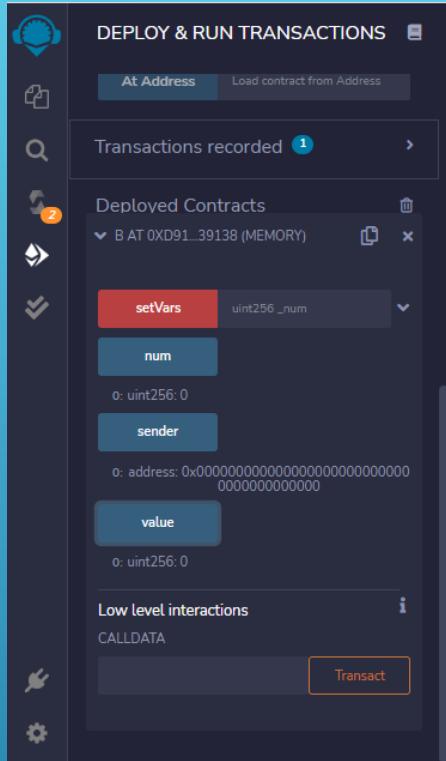
// NOTE: Deploy this contract first
contract B {
    // NOTE: storage.layout.must.be.the.same.as.contract.A
    uint public num;
    address public sender;
    uint public value;

    function setVars(uint _num) public payable {
        num = _num;
        sender = msg.sender;
        value = msg.value;
    }
}

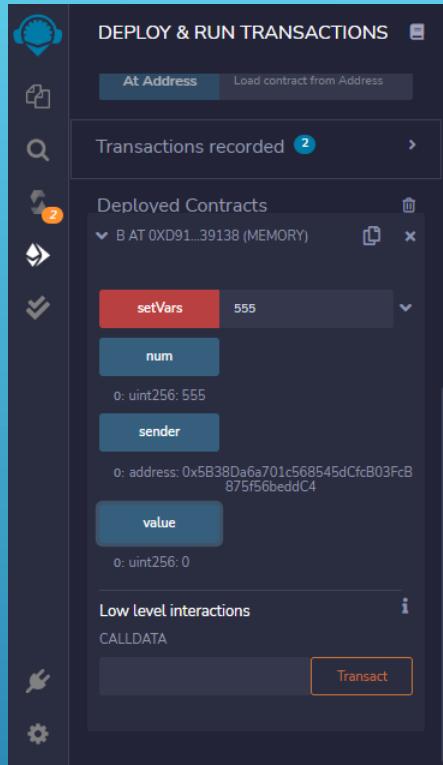
contract A {
    uint public num;
    address public sender;
    uint public value;

    function setVars(address _contract, uint _num) public payable {
        // 'num' is not modified.
        // 'sender' is not modified.
        // 'value' is not modified.
        // 'storage.layout.must.be.the.same.as.contract.A'
        _contract.delegatecall(
            abi.encodeWithSignature("setVars(uint)", _num)
        );
    }
}
```

LALU KITA DEPLOY CONTRACT B, MAKA AKAN MUNCUL TAMPAKAN SEPERTI INI



KLIK NUM, SENDER, DAN VALUE MAKA
ISINYA AKAN KOSONG SEPERTI DIBAWAH
INI PADA CONTRACT B.



LALU KITA SETVARS 555 MAKA IA AKAN
MENGISI NUM 555 DAN AKAN MENGIRIM
SENDER.

DEPLOY & RUN TRANSACTIONS

ENVIRONMENT
JavaScript VM (London)

ACCOUNT
0x5B3...eddC4 (99.999999)

GAS LIMIT
3000000

VALUE
0 Wei

CONTRACT
A - DelegateCallExample.sol

Deploy

Publish to IPFS

OR

At Address Load contract from Address

Transactions recorded 3

DelegateCallExample.sol

```
function setVars(uint _num) public payable {
    num = _num;
    sender = msg.sender;
    value = msg.value;
}

contract A {
    uint public num;
    address public sender;
    uint public value;

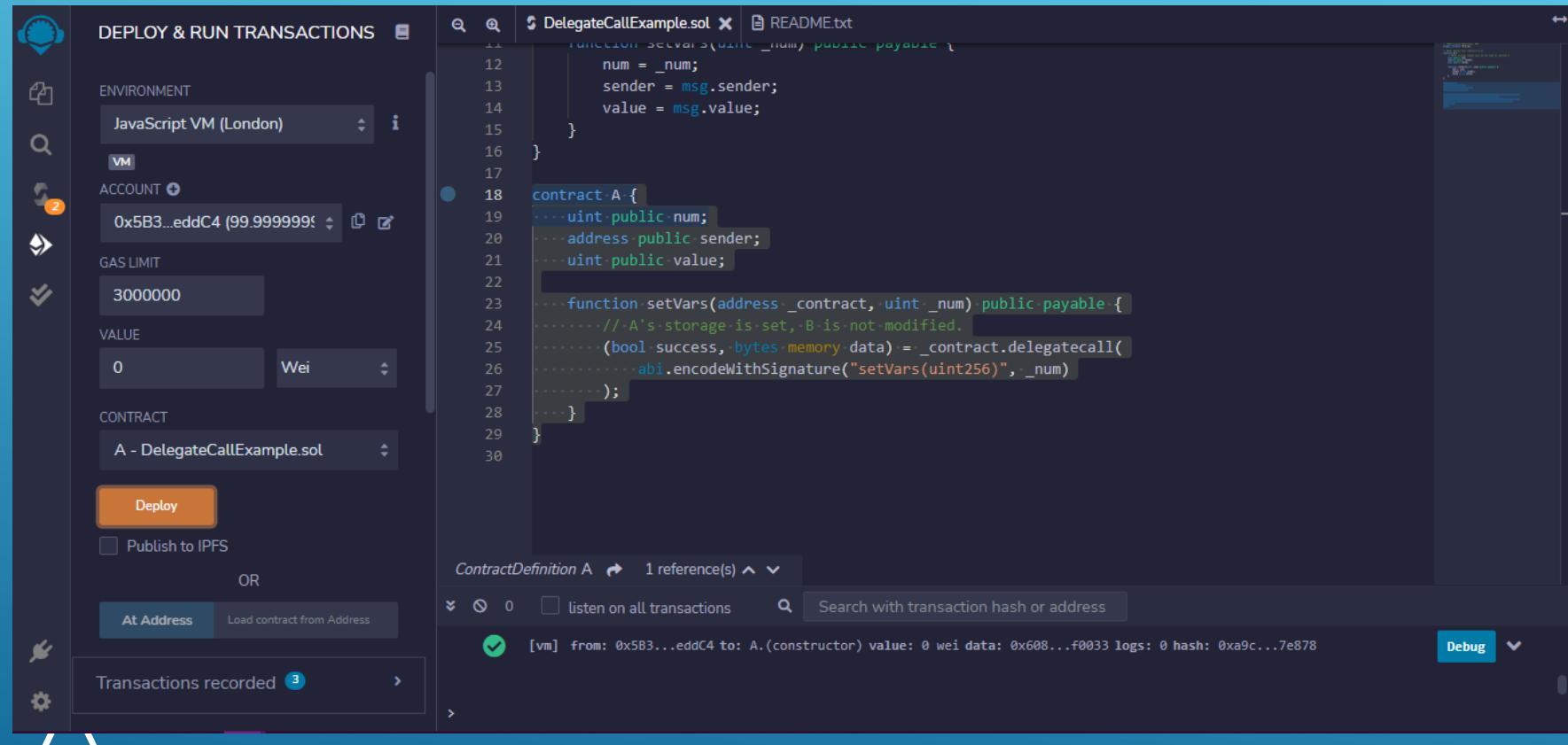
    function setVars(address _contract, uint _num) public payable{
        // A's storage is set, B is not modified.
        (bool success, bytes memory data) = _contract.delegatecall(
            abi.encodeWithSignature("setVars(uint256)", _num)
        );
    }
}
```

ContractDefinition A 1 reference(s)

0 listen on all transactions Search with transaction hash or address

[vm] from: 0x5B3...eddC4 to: A.(constructor) value: 0 wei data: 0x608...f0033 logs: 0 hash: 0xa9c...7e878

Debug



DEPLOY & RUN TRANSACTIONS

value
0: uint256: 0

Low level interactions

CALDATA

Transact

A AT 0XF8E...9FBE8 (MEMORY)

setVars address _contract, uint256

num

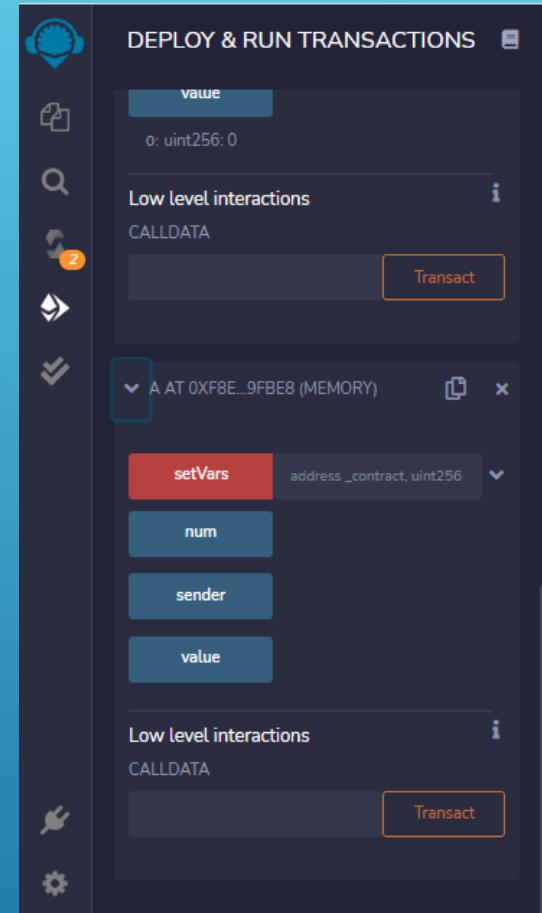
sender

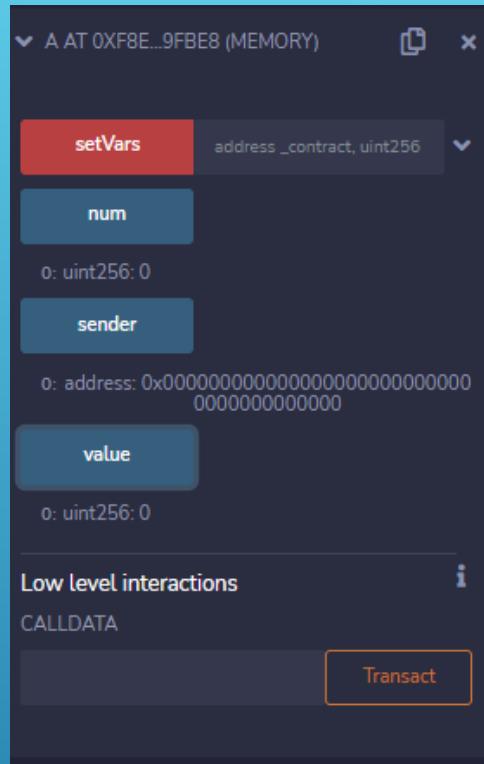
value

Low level interactions

CALDATA

Transact





KLIK NUM, SENDER, DAN VALUE MAKA
ISINYA AKAN KOSONG SEPERTI DIBAWAH
INI, PADA CONTRACT A.

LALU KITA PINJAM FUNGSI DARI CONTRACT B RUN SEKALI, LALU HAPUS KEMBALI FUNCTION TERSEBUT, DAN INI ADALAH PERUMPAMAAN SISTEM DELEGATE BERKERJA

```
function setVars(uint _num) public payable {
    num = _num;
    sender = msg.sender;
    value = msg.value;
}

contract A {
    uint public num;
    address public sender;
    uint public value;

    function setVars(address _contract, uint _num) public payable {
        // A's storage is set, B is not modified.
        (bool success, bytes memory data) = _contract.delegatecall(
            abi.encodeWithSignature("setVars(uint256)", _num)
        );
    }

    function setVars(uint _num) public payable {
        num = _num;
        sender = msg.sender;
        value = msg.value;
    }
}
```

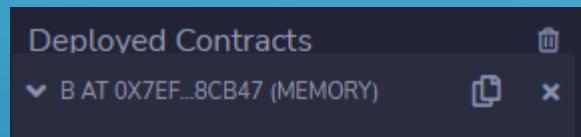
```
function setVars(uint _num) public payable {
    num = _num;
    sender = msg.sender;
    value = msg.value;
}

contract A {
    uint public num;
    address public sender;
    uint public value;

    function setVars(address _contract, uint _num) public payable {
        // A's storage is set, B is not modified.
        (bool success, bytes memory data) = _contract.delegatecall(
            abi.encodeWithSignature("setVars(uint256)", _num)
        );
    }

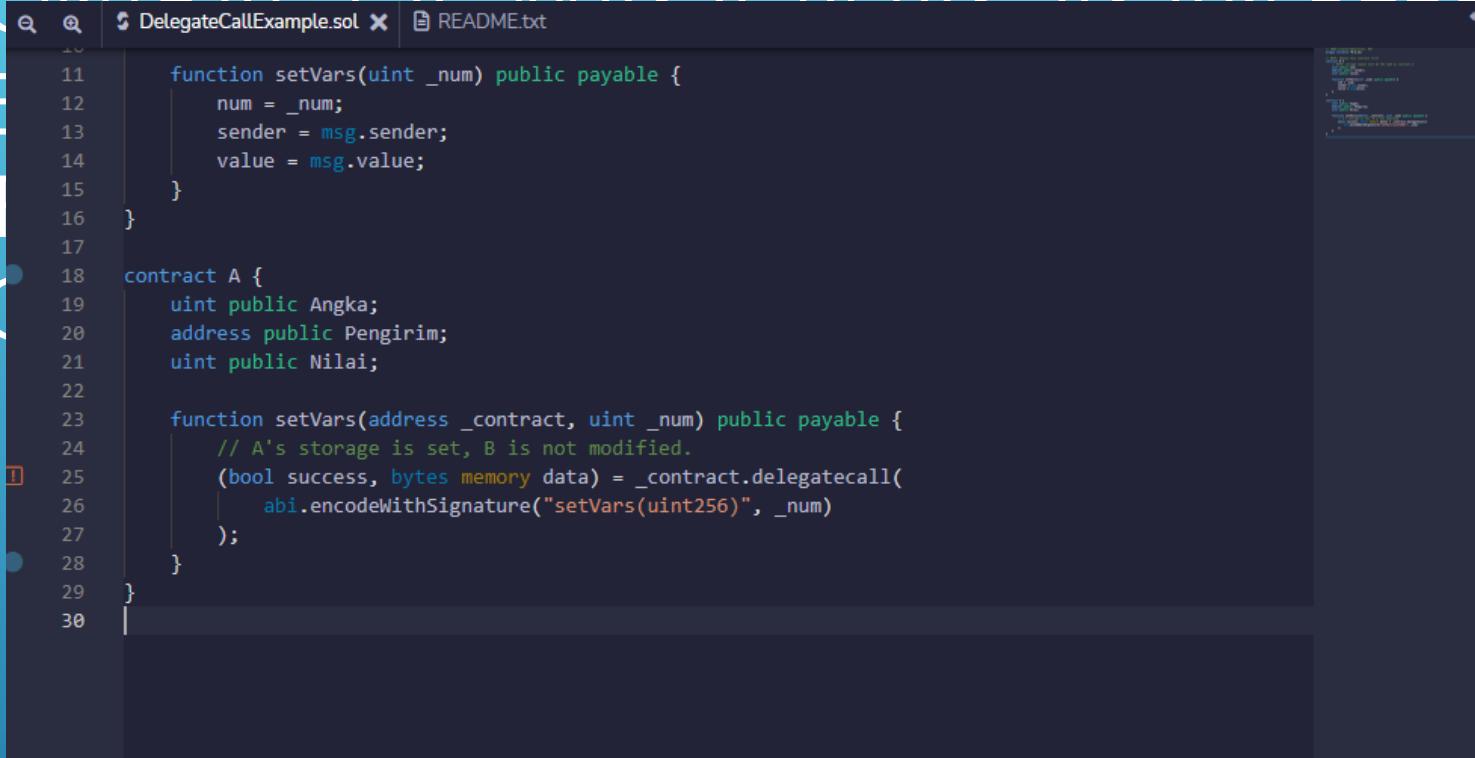
    function setVars(uint _num) public payable {
        num = _num;
        sender = msg.sender;
        value = msg.value;
    }
}
```

KITA COPY ADDRESS PADA CONTRACT B LALU ISI SETVARS DENGAN ADDRESS DAN CALL CONTRACT A, MAKA BERJALAN DENGAN BAIK



A screenshot of the 'A AT 0X358...D5EE3 (MEMORY)' contract interaction interface. The 'setVars' tab is active, showing a field with the value '97943daF4ED8CB47, 666'. Below it are three other tabs: 'num' (value: 0: uint256: 666), 'sender' (value: 0: address: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4), and 'value' (value: 0: uint256: 0). At the bottom, there's a 'Low level interactions' section with a 'CALldata' button and a 'Transact' button.

JIKA KITA MENGUBAH NAMA VARIABLE PADA
CONTRACT A Maka TIDAK AKAN BERDENGARUH
DEPON DI CONTRACT B

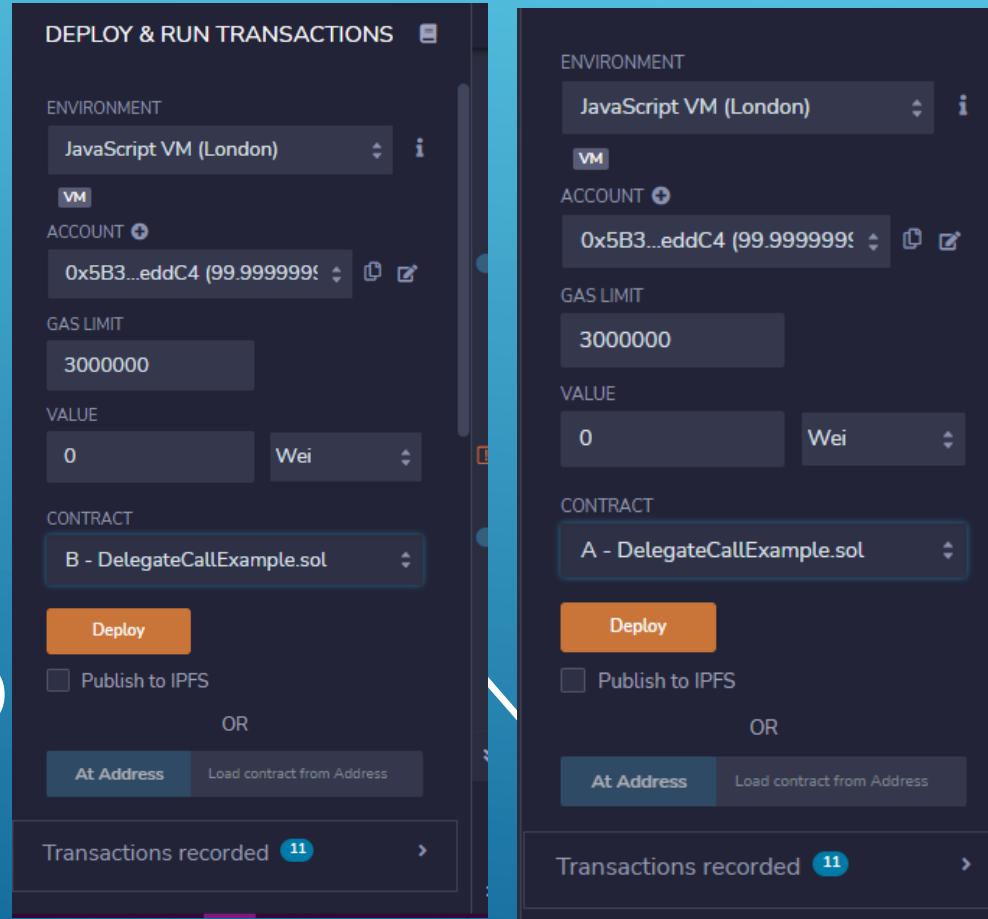


The image shows a screenshot of a code editor with a dark theme. The file is named 'DelegateCallExample.sol'. The code defines a contract 'A' with the following structure:

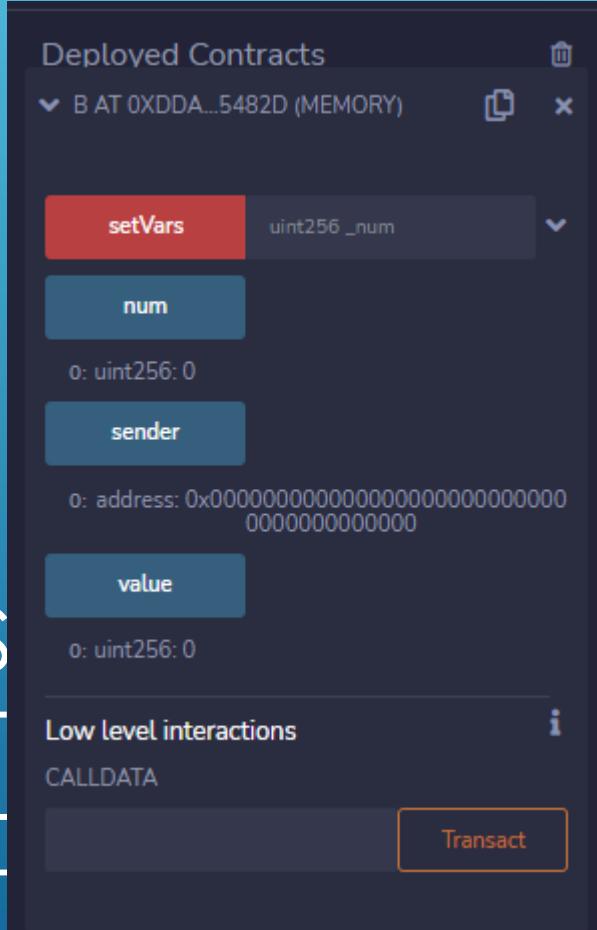
```
11 function setVars(uint _num) public payable {
12     num = _num;
13     sender = msg.sender;
14     value = msg.value;
15 }
16 }
17
18 contract A {
19     uint public Angka;
20     address public Pengirim;
21     uint public Nilai;
22
23     function setVars(address _contract, uint _num) public payable {
24         // A's storage is set, B is not modified.
25         (bool success, bytes memory data) = _contract.delegatecall(
26             abi.encodeWithSignature("setVars(uint256)", _num)
27         );
28     }
29 }
30 }
```

The code uses the `delegatecall` function to call the `setVars` function of another contract, `_contract`, passing it the value of `_num`. The variables `Angka`, `Pengirim`, and `Nilai` are declared as public in the `A` contract.

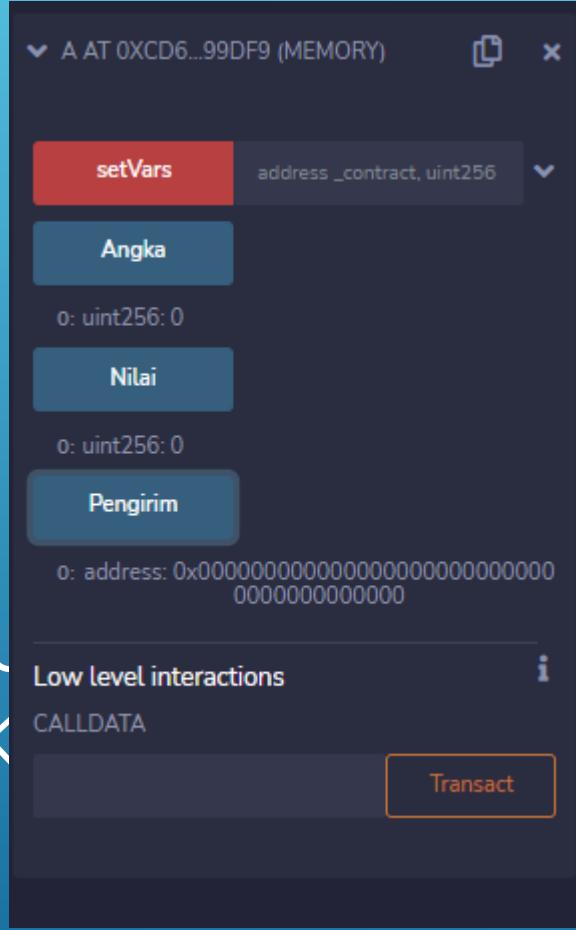
LALU KITA D A DAN B

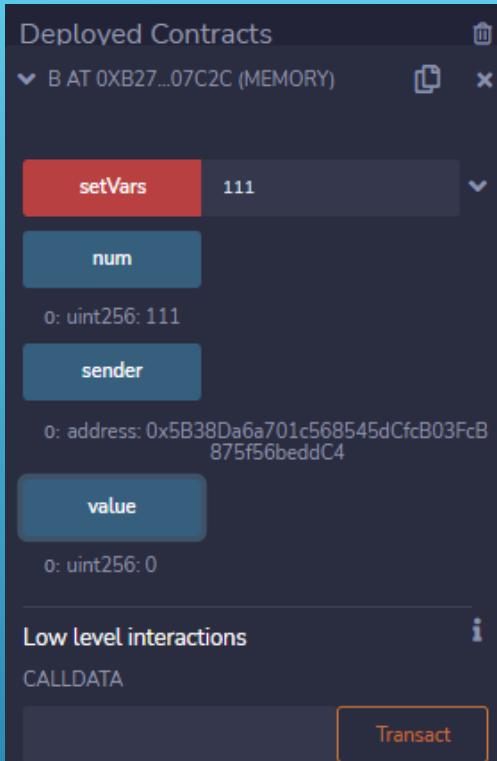


KLIK NUM,
CONTRACT
DAN NILAI



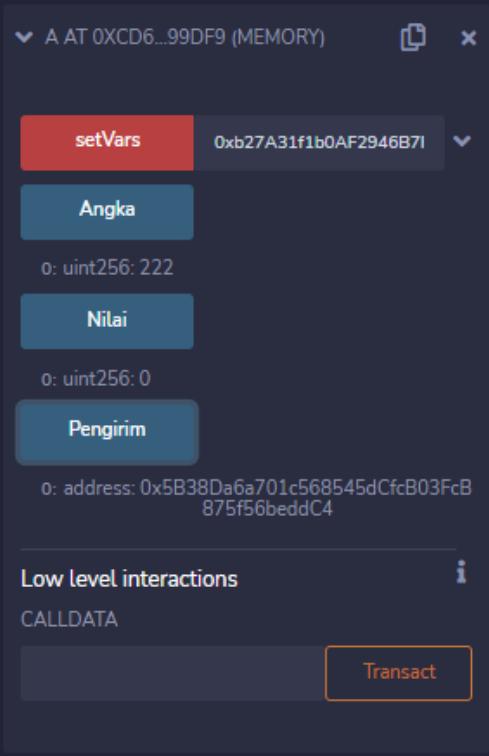
ALL
ANGKA.





KITA SETVARS LAGI PADA CONTRACT B

COPY ALAMAT CONTRACT A
SETVARS PADA DIINGINKAN, M
PERSIS SAMA D
DENGAN NAM

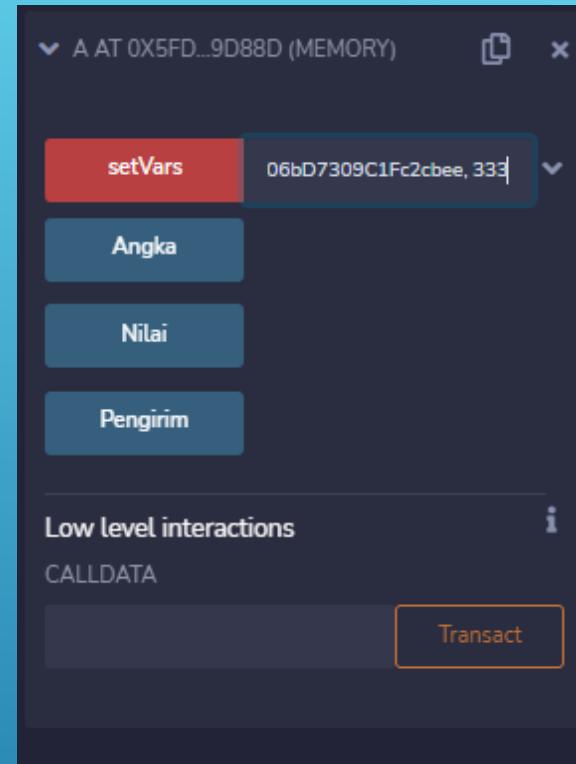


CONTRACT B DAN MASUKAN DI
A BESERTA NILAI YANG
RACT A AKAN BEKERJA
NTRACT B, WALAUPUN
YANG BERBEDA

```
contract A {
    bool public Angka;
    address public Pengirim;
    uint public Nilai;

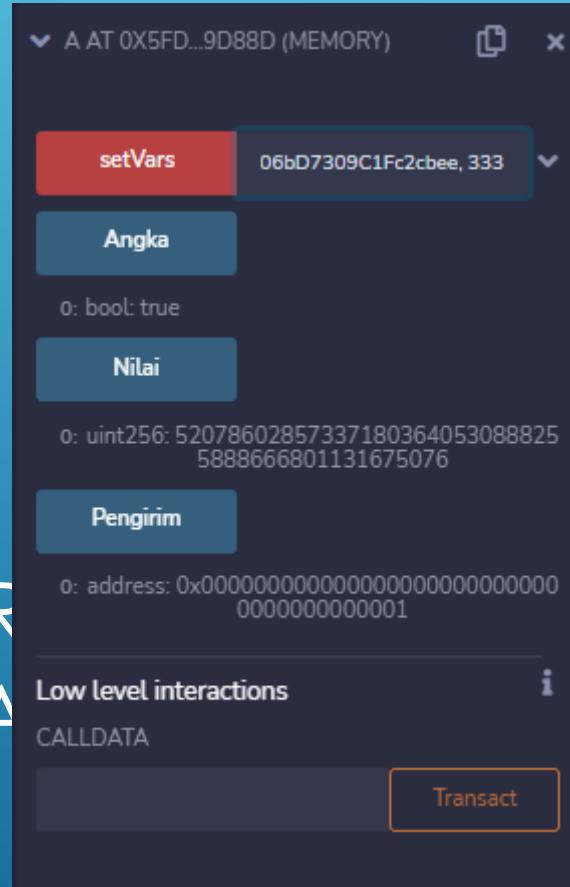
    function setVars(address _contract, uint _num) public payable {
        // A's storage is set, B is not modified.
        (bool success, bytes memory data) = _contract.delegatecall(
            abi.encodeWithSignature("setVars(uint256)", _num)
        );
    }
}
```

SEKARANG COBA KITA UBAH TIPE VARIABLE ANGKA PADA CONTRACT A JADI BOOLEAN

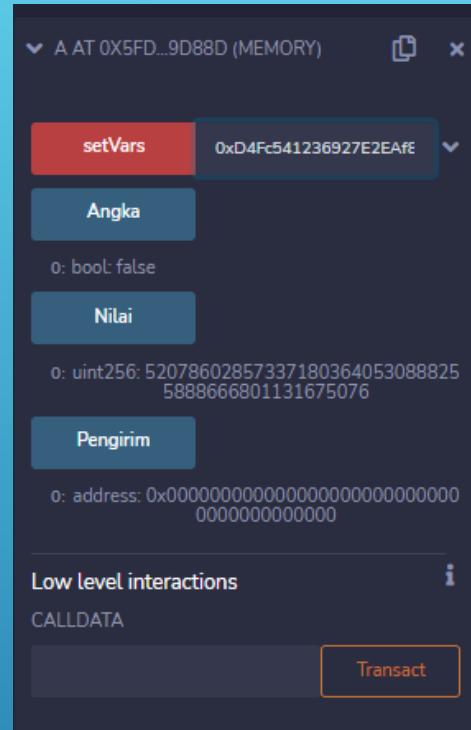


KITA DEPLOY KEMBALI CONTRACT A DAN
ISI KEMBALI ADDRESS CONTRACT B SERTA
SETVARS YANG KITA INGINKAN

KLIK ANGKA, PENGIRIM
CONTRACT A, MAKAN
KARENA BOOLEAN



DI
AN TRUE



JIKA KITA ISI 0 PADA SETVARSNYA, MAKA AKAN BERNILAI FALSE KARENA PADA CONTRACT B VALUENYA ADALAH INTEGER/ANGKA JADI SELAIN 0 AKAN BERNILAI TRUE, SEDANGKAN 0 FALSE

LESSON 17 : HARDHAT DAOS

Voting Contract

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell <https://aka.ms/pscore6>

PS E:\BlockChain> **mkdir** hardhat-dao-fcc

Directory: E:\BlockChain

Mode	LastWriteTime	Length	Name
d----	7/6/2022 9:34 PM		hardhat-dao-fcc

PS E:\BlockChain> **cd** hardhat-dao-fcc\

PS E:\BlockChain\hardhat-dao-fcc> **code** .

PERTAMA BUAT FLODER HARDHAT-DAO-FCC DAN MASUK KE DALAMANYA

```
PS E:\BlockChain\hardhat-dao-fcc> git --version  
git version 2.37.0.windows.1  
PS E:\BlockChain\hardhat-dao-fcc> node --version  
v16.9.1  
PS E:\BlockChain\hardhat-dao-fcc> yarn --version  
1.22.19
```

CERI VERSI YANG KITA PUNYA DARI
MULAI GIT, NODE, DAN YARN

KITA TAMBAHKAN HARDHAT KE YARN

```
PS E:\BlockChain\hardhat-dao-fcc> yarn add --dev hardhat
yarn add v1.22.19
info No lockfile found.
[1/4] Resolving packages...
[2/4] Fetching packages...
[3/4] Linking dependencies...
[4/4] Building fresh packages...
success Saved lockfile.
success Saved 207 new dependencies.
```

BUAT B BERNA

```
hardhat-dao-fcc > contracts > ♦ Box.sol
 1 // contracts/Box.sol
 2 // SPDX-License-Identifier: MIT
 3 pragma solidity ^0.8.0;
 4
 5 import "@openzeppelin/contracts/access/Ownable.sol";
 6
 7 contract Box is Ownable {
 8     uint256 private value;
 9
10     // Emitted when the stored value changes
11     event ValueChanged(uint256 newValue);
12
13     // Stores a new value in the contract
14     function store(uint256 newValue) public onlyOwner {
15         value = newValue;
16         emit ValueChanged(newValue);
17     }
18
19     // Reads the last stored value
20     function retrieve() public view returns (uint256) {
```

```
PS E:\BlockChain\hardhat-dao-fcc> yarn add --dev @openzeppelin/contracts
yarn add v1.22.19
warning package.json: No license field
warning No license field
[1/4] Resolving packages...
[2/4] Fetching packages...
[3/4] Linking dependencies...
[4/4] Building fresh packages...

warning No license field
success Saved 1 new dependency.
info Direct dependencies
└─ @openzeppelin/contracts@4.7.0
info All dependencies
└─ @openzeppelin/contracts@4.7.0
```

KITA TAMBAHKAN OPEN ZEPPELIN CONTRACTS

```
PS E:\BlockChain\hardhat-dao-fcc> yarn hardhat compile
yarn run v1.22.19
warning package.json: No license field
$ E:\BlockChain\hardhat-dao-fcc\node_modules\.bin\hardhat compile
 888   888           888 888           888
 888   888           888 888           888
 888   888           888 888           888
 8888888888 8888b. 888d888 .d88888 88888b. 8888b. 888888
 888   888     "88b 888P" d88" 888 888 "88b     "88b 888
 888   888 .d888888 888   888 888 888 .d888888 888
 888   888 888 888 Y88b 888 888 888 888 888 Y88b.
 888   888 "Y888888 888   "Y888888 888 "Y888888 "Y888

Welcome to Hardhat v2.9.9

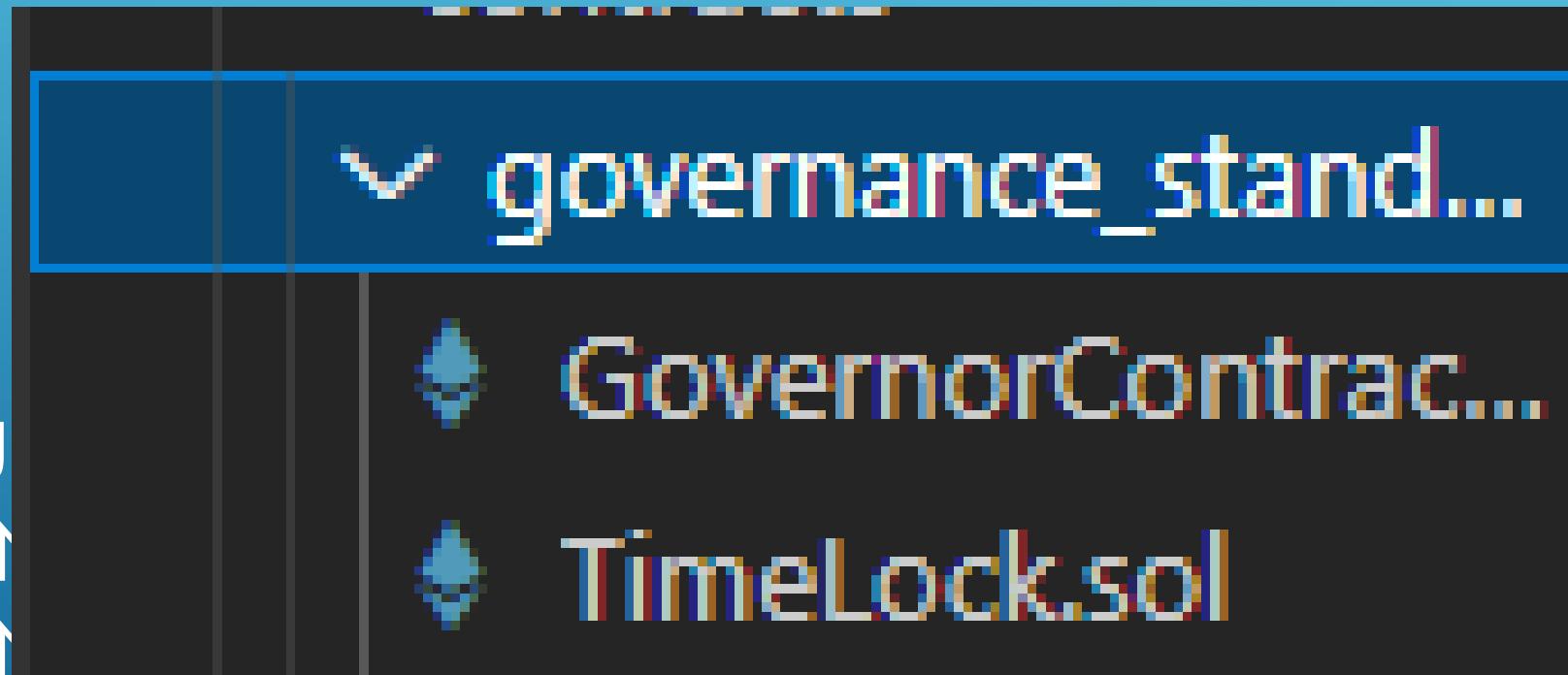
✓ What do you want to do? · Create a basic sample project
✓ Hardhat project root: · E:\BlockChain\hardhat-dao-fcc
✓ Do you want to add a .gitignore? (Y/n) · y
✓ Help us improve Hardhat with anonymous crash reports & basic usage data? (Y/n) · t
rue

You need to install these dependencies to run the sample project:
  yarn add --dev "hardhat@^2.9.9" "@nomiclabs/hardhat-waffle@^2.0.0" "ethereum-waffl
e@^3.0.0" "chai@^4.2.0" "@nomiclabs/hardhat-ethers@^2.0.0" "ethers@^5.0.0"

Project created
See the README.md file for some example tasks you can run.
Done in 65.59s.
```

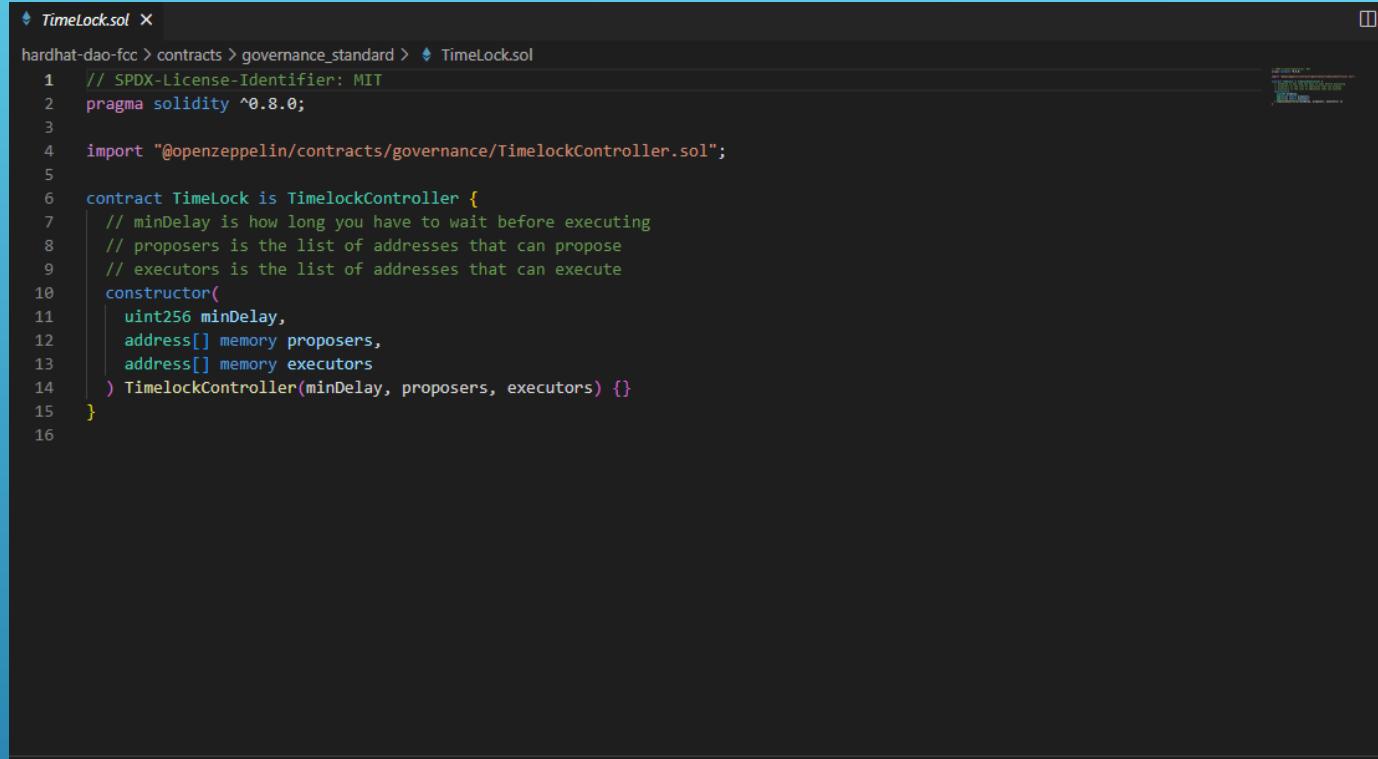
LALU KITA COMPILE HARDHAT
TERSEBUT

LALU.BUF
GOVERNOR.SOK
GOVERNOR.sol
TIMELOCK.SOK



```
◆ GovernorContract.sol ✘
hardhat-dao-fcc > contracts > governance_standard > ◆ GovernorContract.sol
1 // SPDX-License-Identifier: MIT
2 pragma solidity ^0.8.9;
3
4 import "@openzeppelin/contracts/governance/Governor.sol";
5 import "@openzeppelin/contracts/governance/extensions/GovernorCountingSimple.sol";
6 import "@openzeppelin/contracts/governance/extensions/GovernorVotes.sol";
7 import "@openzeppelin/contracts/governance/extensions/GovernorVotesQuorumFraction.sol";
8 import "@openzeppelin/contracts/governance/extensions/GovernorTimelockControl.sol";
9
10 contract GovernorContract is
11     Governor,
12     GovernorCountingSimple,
13     GovernorVotes,
14     GovernorVotesQuorumFraction,
15     GovernorTimelockControl
16 {
17     uint256 public s_votingDelay;
18     uint256 public s_votingPeriod;
19
20     constructor(
21         ERC20Votes _token,
22         TimelockController _timelock,
23         uint256 _quorumPercentage,
24         uint256 _votingPeriod,
25         uint256 _votingDelay
26     )
27         Governor("GovernorContract")
28         GovernorVotes(_token)
29         GovernorVotesQuorumFraction(_quorumPercentage)
```

GOVERNANCECONTRACT.SOL BERISI
HAMPIR SEMUA LOGIKA UNTUK SISTEM
VOTING INI



```
TimeLock.sol ✘
hardhat-dao-fcc > contracts > governance_standard > TimeLock.sol
1 // SPDX-License-Identifier: MIT
2 pragma solidity ^0.8.0;
3
4 import "@openzeppelin/contracts/governance/TimelockController.sol";
5
6 contract TimeLock is TimelockController {
7     // minDelay is how long you have to wait before executing
8     // proposers is the list of addresses that can propose
9     // executors is the list of addresses that can execute
10    constructor(
11        uint256 minDelay,
12        address[] memory proposers,
13        address[] memory executors
14    ) TimelockController(minDelay, proposers, executors) {}
15}
16
```

TIME LOCK BERISI TENTANG ADDITIONAL CONTRACT SI OWNERNYA

```
PS E:\BlockChain\hardhat-dao-fcc> yarn add --dev @nomiclabs/hardhat-ethers@npm:hardhat-deploy-ethers ethers
yarn add v1.22.19
warning package.json: No license field
warning No license field
[1/4] Resolving packages...
[2/4] Fetching packages...
[3/4] Linking dependencies...
[4/4] Building fresh packages...

success Saved lockfile.
warning No license field
success Saved 10 new dependencies.
info Direct dependencies
└─ @nomiclabs/hardhat-ethers@0.3.0-beta.13
    └─ ethers@5.6.9
info All dependencies
└─ @ethersproject/contracts@5.6.2
└─ @ethersproject/json-wallets@5.6.1
└─ @ethersproject/providers@5.6.8
└─ @ethersproject/solidity@5.6.1
└─ @ethersproject/units@5.6.1
└─ @ethersproject/wallet@5.6.2
└─ @nomiclabs/hardhat-ethers@0.3.0-beta.13
    └─ aes-js@3.0.0
    └─ bech32@1.1.4
    └─ ethers@5.6.9
Done in 13.18s.
```

```
Done in 13.18s.          > yarn add --dev hardhat-deploy
yarn add v1.22.19
warning package.json: No license field
warning No license field
[1/4] Resolving packages...
[2/4] Fetching packages...
[3/4] Linking dependencies...
warning "hardhat-deploy > zksync-web3@0.7.9" has incorrect peer dependency "ethers@~5.5.0".
[4/4] Building fresh packages...
success Saved lockfile.
warning No license field
success Saved 15 new dependencies.
info Direct dependencies
└─ hardhat-deploy@0.11.11
info All dependencies
└─ @types/qs@6.9.7
└─ asynckit@0.4.0
└─ axios@0.21.4
└─ combined-stream@1.0.8
└─ delayed-stream@1.0.0
└─ encode-utf8@1.0.3
└─ fmix@0.1.0
└─ form-data@4.0.0
└─ fs-extra@10.1.0
└─ hardhat-deploy@0.11.11
└─ match-all@1.2.6
└─ mime-db@1.52.0
└─ mime-types@2.1.35
└─ murmur-128@0.2.1
└─ zksync-web3@0.7.9
Done in 8.08s.
PS E:\BlockChain\hardhat-dao-fcc>
```

KITA INSTALL YARN ADD --DEV TYPESCRIPT TYPECHAIN

TS
@
UN

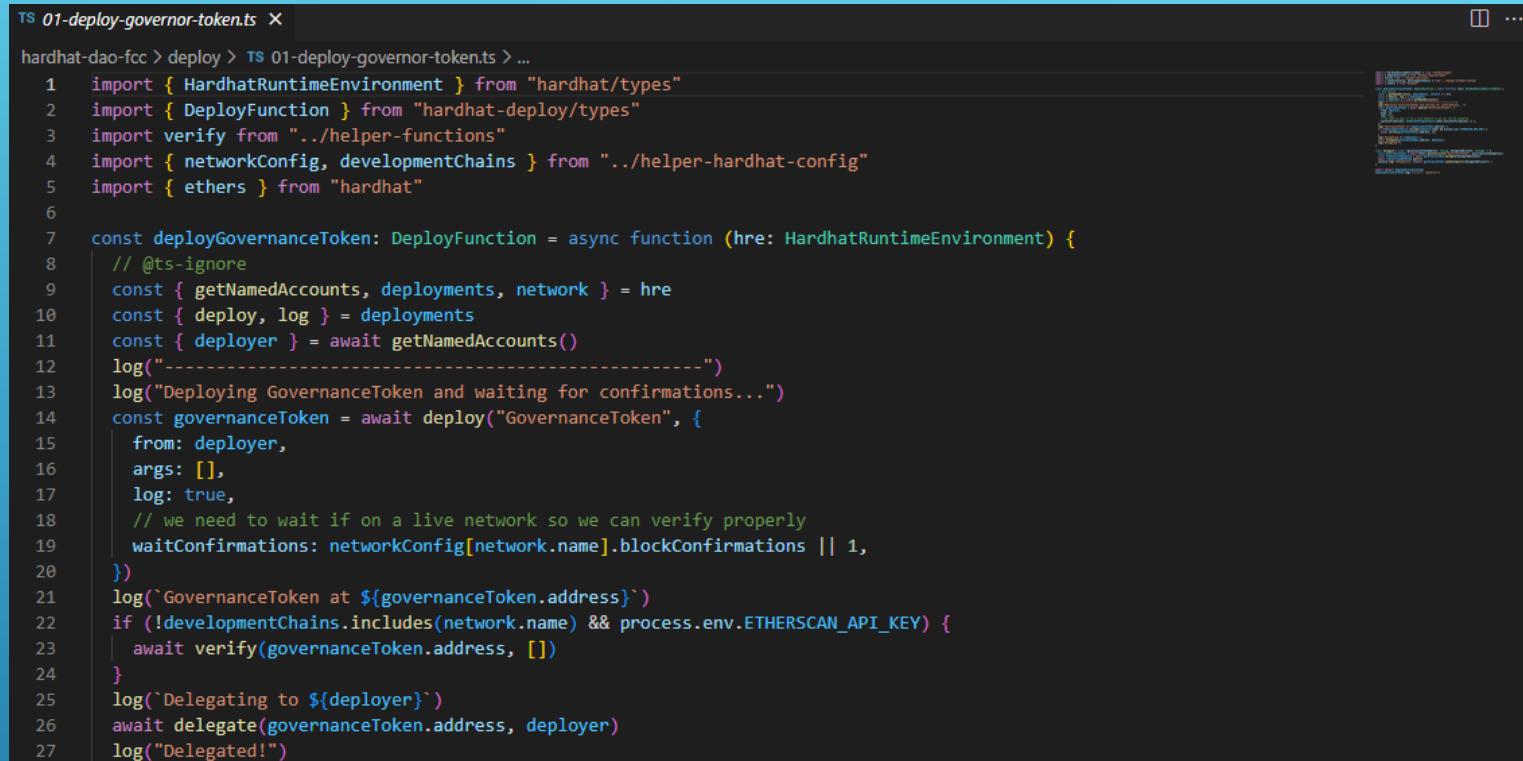
```
PS E:\BlockChain\hardhat-dao-fcc> yarn add --dev typescript typechain ts-node @typechain/ethers-v5 @typechain/hardhat @types/chai @types/node
yarn add v1.22.19
warning package.json: No license field
warning No license field
[1/4] Resolving packages...
[2/4] Fetching packages...
warning Pattern ["@types/node@^18.0.3"] is trying to unpack in the same destination "C:\\Users\\Fath\\AppData\\Local\\Yarn\\Cache\\v6\\npm-@types-node-18.0.3-463fc47f13ec0688a33aec75d078a0541a447199-integrity\\node_modules\\@types\\node" as pattern ["@types/node@","@types/node@","@types/node@*","@types/node@*","@types/node@*"]. This could result in non-deterministic behavior, skipping.
[3/4] Linking dependencies...
warning "hardhat-deploy > zkSyncWeb3@0.7.9" has incorrect peer dependency "ethers@~5.5.0".
warning " > @typechain/ethers-v5@10.1.0" has unmet peer dependency "@ethersproject/abi@^5.0.0".
warning " > @typechain/ethers-v5@10.1.0" has unmet peer dependency "@ethersproject/bytes@^5.0.0".
warning " > @typechain/ethers-v5@10.1.0" has unmet peer dependency "@ethersproject/providers@^5.0.0".
warning " > @typechain/hardhat@6.1.2" has unmet peer dependency "@ethersproject/abi@^5.4.7".
warning " > @typechain/hardhat@6.1.2" has unmet peer dependency "@ethersproject/providers@^5.4.7".
[4/4] Building fresh packages...
success Saved lockfile.
warning No license field
success Saved 38 new dependencies.
info Direct dependencies
└─ @typechain/ethers-v5@10.1.0
└─ @typechain/hardhat@6.1.2
```

TYPES/NODE

B
FILE

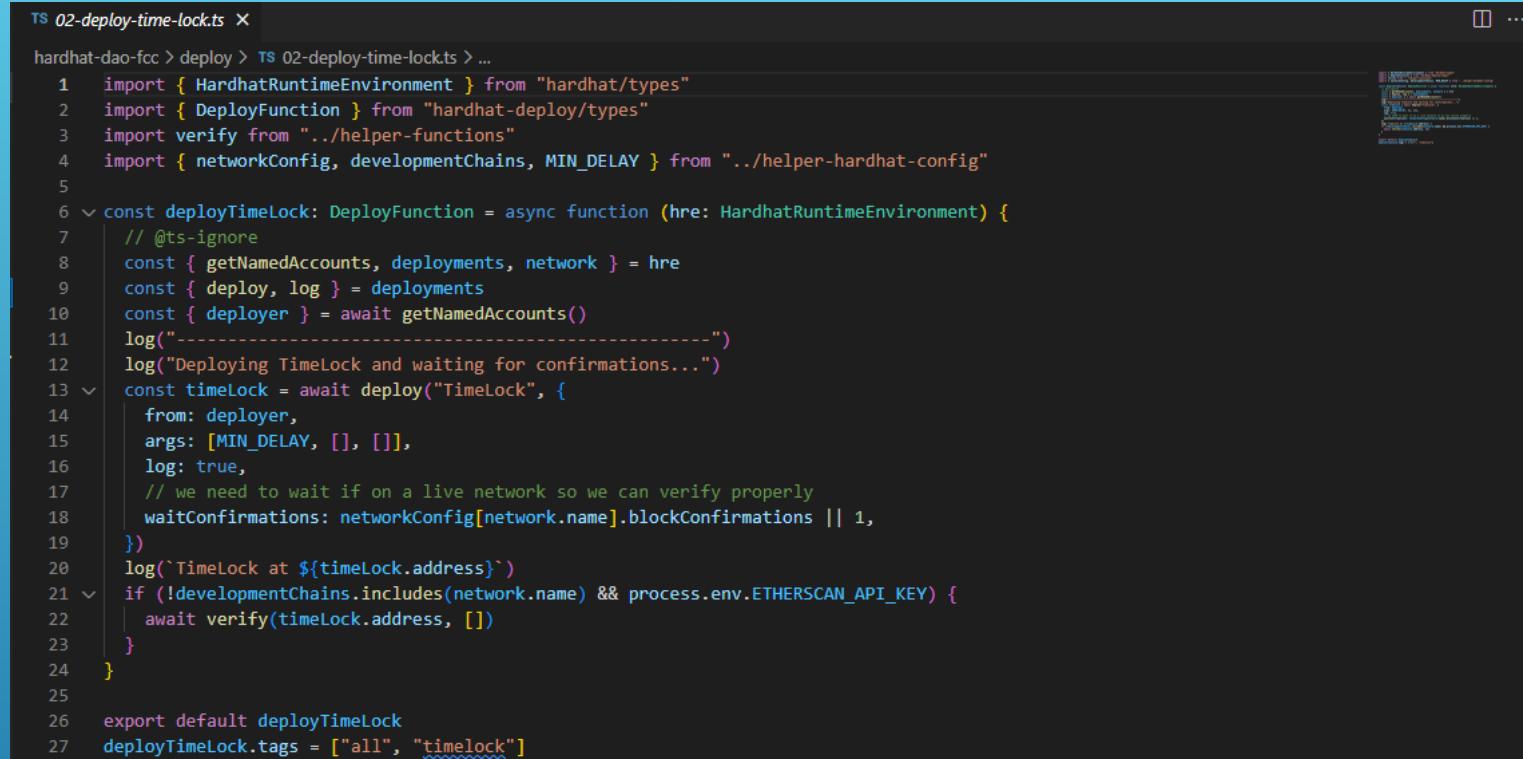
	✓ deploy
	TS 01-deploy-governor-token.ts
	TS 02-deploy-time-lock.ts
	TS 03-deploy-governor-contract.ts
	TS 04-setup-governance-contracts.ts
	TS 05-deploy-box.ts

01-deploy-governor-token.ts
02-deploy-time-lock.ts
03-deploy-governor-
contract.ts
04-setup-governance-
contracts.ts
05-deploy-box.ts



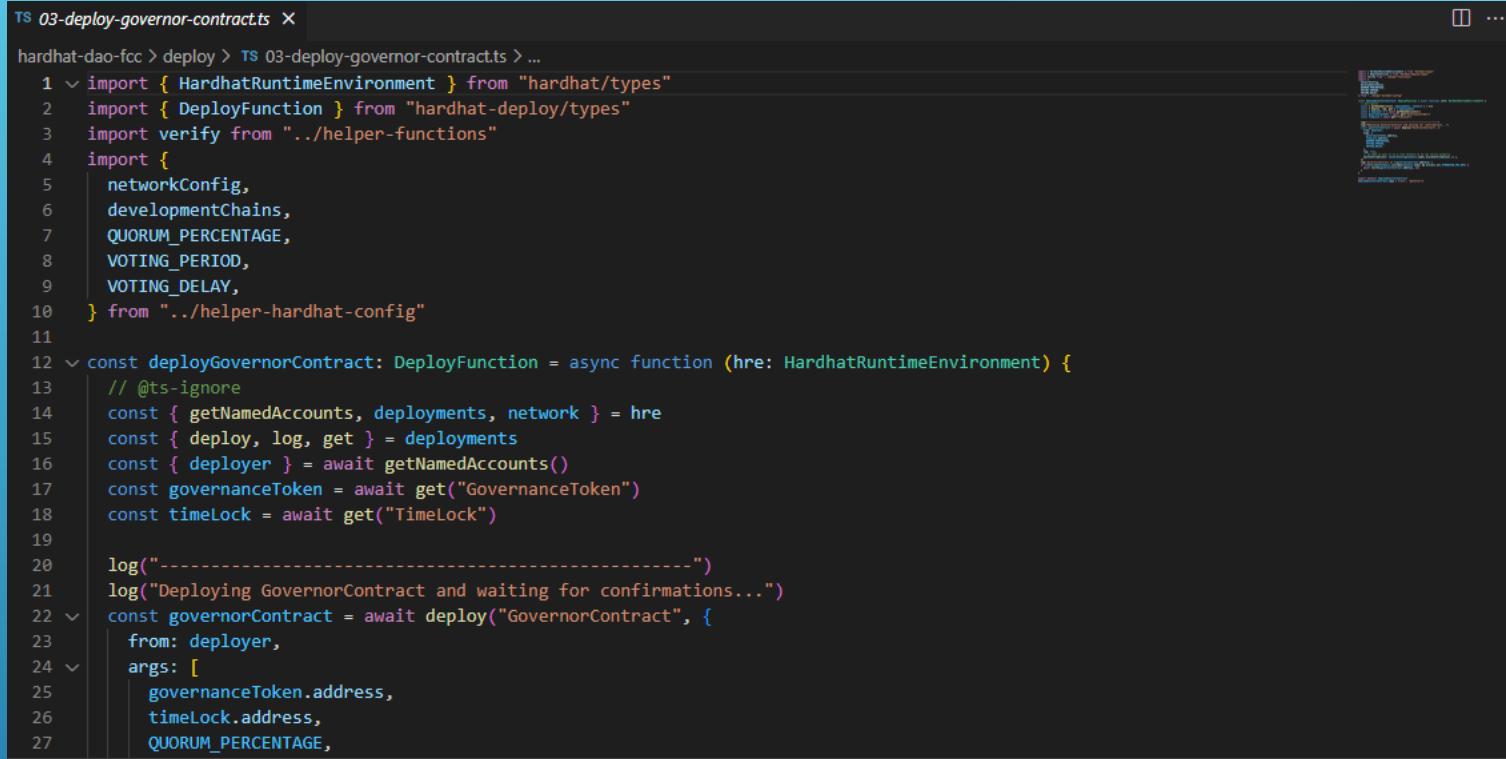
```
ts 01-deploy-governor-token.ts ×
hardhat-dao-fcc > deploy > TS 01-deploy-governor-token.ts > ...
1 import { HardhatRuntimeEnvironment } from "hardhat/types"
2 import { DeployFunction } from "hardhat-deploy/types"
3 import verify from "../helper-functions"
4 import { networkConfig, developmentChains } from "../helper-hardhat-config"
5 import { ethers } from "hardhat"
6
7 const deployGovernanceToken: DeployFunction = async function (hre: HardhatRuntimeEnvironment) {
8   // @ts-ignore
9   const { getNamedAccounts, deployments, network } = hre
10  const { deploy, log } = deployments
11  const { deployer } = await getNamedAccounts()
12  log(`-----`)
13  log(`Deploying GovernanceToken and waiting for confirmations...`)
14  const governanceToken = await deploy("GovernanceToken", {
15    from: deployer,
16    args: [],
17    log: true,
18    // we need to wait if on a live network so we can verify properly
19    waitConfirmations: networkConfig[network.name].blockConfirmations || 1,
20  })
21  log(`GovernanceToken at ${governanceToken.address}`)
22  if (!developmentChains.includes(network.name) && process.env.ETHERSCAN_API_KEY) {
23    await verify(governanceToken.address, [])
24  }
25  log(`Delegating to ${deployer}`)
26  await delegate(governanceToken.address, deployer)
27  log(`Delegated!`)
```

FILE UTI-DEPLOY-GOVERNOR-TOKEN.TS UNTUK MENDEPLOY TOKEN, MEMBERI TIAP AKUN PRIVATE KEY



```
ts 02-deploy-time-lock.ts X
hardhat-dao-fcc > deploy > ts 02-deploy-time-lock.ts > ...
1 import { HardhatRuntimeEnvironment } from "hardhat/types"
2 import { DeployFunction } from "hardhat-deploy/types"
3 import verify from "../helper-functions"
4 import { networkConfig, developmentChains, MIN_DELAY } from "../helper-hardhat-config"
5
6 const deployTimeLock: DeployFunction = async function (hre: HardhatRuntimeEnvironment) {
7   // @ts-ignore
8   const { getNamedAccounts, deployments, network } = hre
9   const { deploy, log } = deployments
10  const { deployer } = await getNamedAccounts()
11  log(`-----`)
12  log("Deploying TimeLock and waiting for confirmations...")
13  const timeLock = await deploy("TimeLock", {
14    from: deployer,
15    args: [MIN_DELAY, [], []],
16    log: true,
17    // we need to wait if on a live network so we can verify properly
18    waitConfirmations: networkConfig[network.name].blockConfirmations || 1,
19  })
20  log(`TimeLock at ${timeLock.address}`)
21  if (!developmentChains.includes(network.name) && process.env.ETHERSCAN_API_KEY) {
22    await verify(timeLock.address, [])
23  }
24}
25
26 export default deployTimeLock
27 deployTimeLock.tags = ["all", "timelock"]
```

FILE 02-DEPLOY-TIME-LOCK.TS UNTUK MENDEPLOY TIME LOCKS, ADDITIONAL CONTRACT OWNER



```
TS 03-deploy-governor-contract.ts X
hardhat-dao-fcc > deploy > TS 03-deploy-governor-contract.ts > ...
1 import { HardhatRuntimeEnvironment } from "hardhat/types"
2 import { DeployFunction } from "hardhat-deploy/types"
3 import verify from "../helper-functions"
4 import {
5   networkConfig,
6   developmentChains,
7   QUORUM_PERCENTAGE,
8   VOTING_PERIOD,
9   VOTING_DELAY,
10 } from "../helper-hardhat-config"
11
12 const deployGovernorContract: DeployFunction = async function (hre: HardhatRuntimeEnvironment) {
13   // @ts-ignore
14   const { getNamedAccounts, deployments, network } = hre
15   const { deploy, log, get } = deployments
16   const { deployer } = await getNamedAccounts()
17   const governanceToken = await get("GovernanceToken")
18   const timeLock = await get("TimeLock")
19
20   log("-----")
21   log("Deploying GovernorContract and waiting for confirmations...")
22   const governorContract = await deploy("GovernorContract", {
23     from: deployer,
24     args: [
25       governanceToken.address,
26       timeLock.address,
27       QUORUM_PERCENTAGE,
```

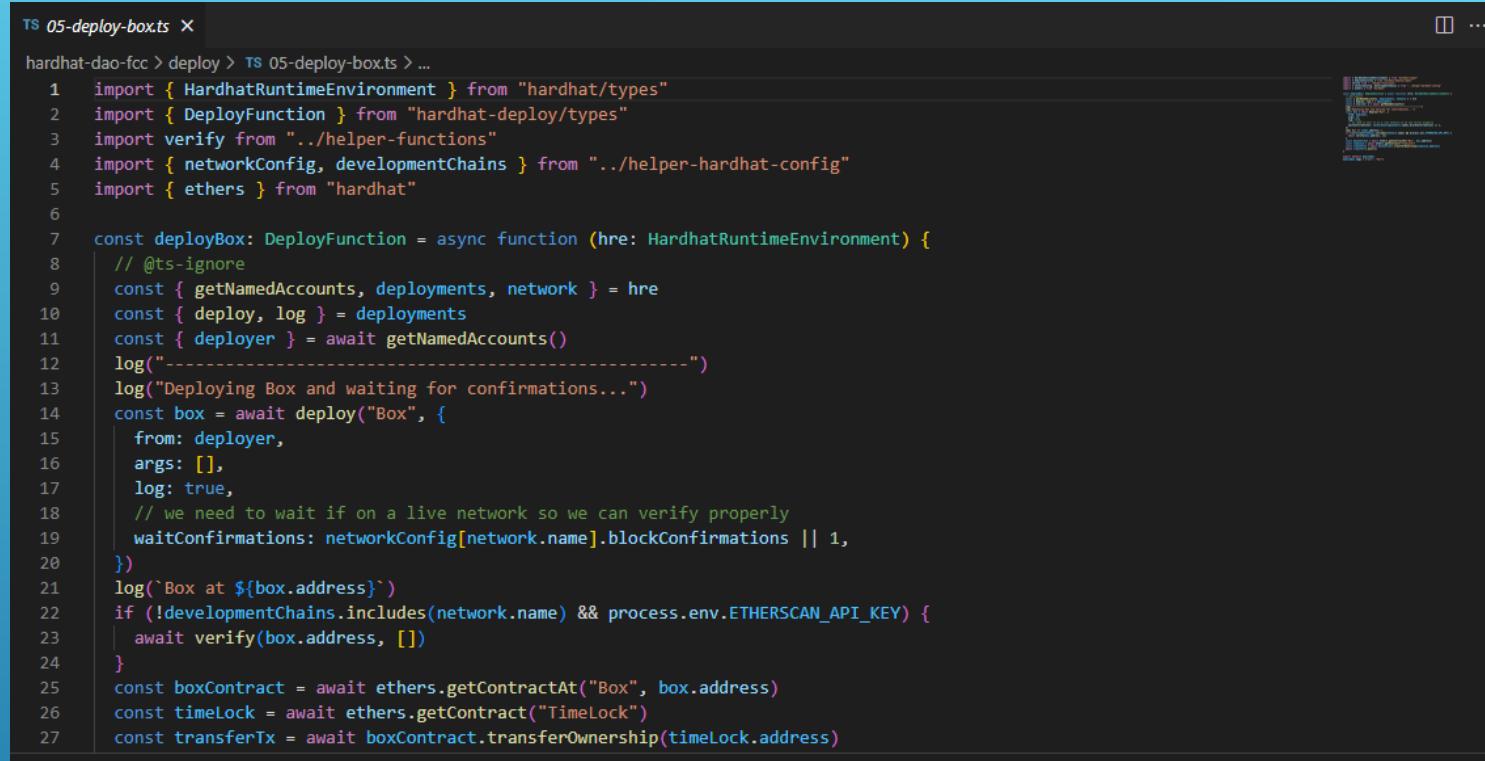
FILE 03-DEPLOY-GOVERNOR-
CONTRACT.TS UNTUK MENDEPLOY
SEMUA LOGIC DALAM VOTING.

The screenshot shows a code editor window with a dark theme. The file being edited is '04-setup-governance-contracts.ts'. The code is written in TypeScript and interacts with Hardhat and Ethers.js to set up governance contracts. It defines a function 'setupContracts' that deploys three contracts: GovernanceToken, TimeLock, and GovernorContract. It then grants roles to the TimeLock contract (PROPOSER_ROLE, EXECUTOR_ROLE, and TIMELOCK_ADMIN_ROLE) and revokes the ADMIN_ROLE from the deployer.

```
ts 04-setup-governance-contracts.ts X
hardhat-dao-fcc > deploy > TS 04-setup-governance-contracts.ts > ...

1 import { HardhatRuntimeEnvironment } from "hardhat/types"
2 import { DeployFunction } from "hardhat-deploy/types"
3 import verify from "../helper-functions"
4 import { networkConfig, developmentChains, ADDRESS_ZERO } from "../helper-hardhat-config"
5 import { ethers } from "hardhat"
6
7 const setupContracts: DeployFunction = async function (hre: HardhatRuntimeEnvironment) {
8     // @ts-ignore
9     const { getNamedAccounts, deployments, network } = hre
10    const { log } = deployments
11    const { deployer } = await getNamedAccounts()
12    const governanceToken = await ethers.getContract("GovernanceToken", deployer)
13    const timeLock = await ethers.getContract("TimeLock", deployer)
14    const governor = await ethers.getContract("GovernorContract", deployer)
15
16    log("-----")
17    log("Setting up contracts for roles...")
18    // would be great to use multicall here...
19    const proposerRole = await timeLock.PROPOSER_ROLE()
20    const executorRole = await timeLock.EXECUTOR_ROLE()
21    const adminRole = await timeLock.TIMELOCK_ADMIN_ROLE()
22
23    const proposerTx = await timeLock.grantRole(proposerRole, governor.address)
24    await proposerTx.wait(1)
25    const executorTx = await timeLock.grantRole(executorRole, ADDRESS_ZERO)
26    await executorTx.wait(1)
27    const revokeTx = await timeLock.revokeRole(adminRole, deployer)
```

FILE 04-SETUP-GOVERNANCE-CONTRACTS.TS UNTUK MENGATUR CARA KERJA VOTING YANG NANTI AKAN BERJALAN



```
ts 05-deploy-box.ts X
hardhat-dao-fcc > deploy > TS 05-deploy-box.ts > ...
1 import { HardhatRuntimeEnvironment } from "hardhat/types"
2 import { DeployFunction } from "hardhat-deploy/types"
3 import verify from "../helper-functions"
4 import { networkConfig, developmentChains } from "../helper-hardhat-config"
5 import { ethers } from "hardhat"
6
7 const deployBox: DeployFunction = async function (hre: HardhatRuntimeEnvironment) {
8   // @ts-ignore
9   const { getNamedAccounts, deployments, network } = hre
10  const { deploy, log } = deployments
11  const { deployer } = await getNamedAccounts()
12  log("-----")
13  log("Deploying Box and waiting for confirmations...")
14  const box = await deploy("Box", {
15    from: deployer,
16    args: [],
17    log: true,
18    // we need to wait if on a live network so we can verify properly
19    waitConfirmations: networkConfig[network.name].blockConfirmations || 1,
20  })
21  log(`Box at ${box.address}`)
22  if (!developmentChains.includes(network.name) && process.env.ETHERSCAN_API_KEY) {
23    await verify(box.address, [])
24  }
25  const boxContract = await ethers.getContractAt("Box", box.address)
26  const timeLock = await ethers.getContract("TimeLock")
27  const transferTx = await boxContract.transferOwnership(timeLock.address)
```

FILE 05-DEPLOY-BOX.TS UNTUK MENDEPLOY SMARTCONTRACT YANG SUDAH KITA BUAT DIAWAL

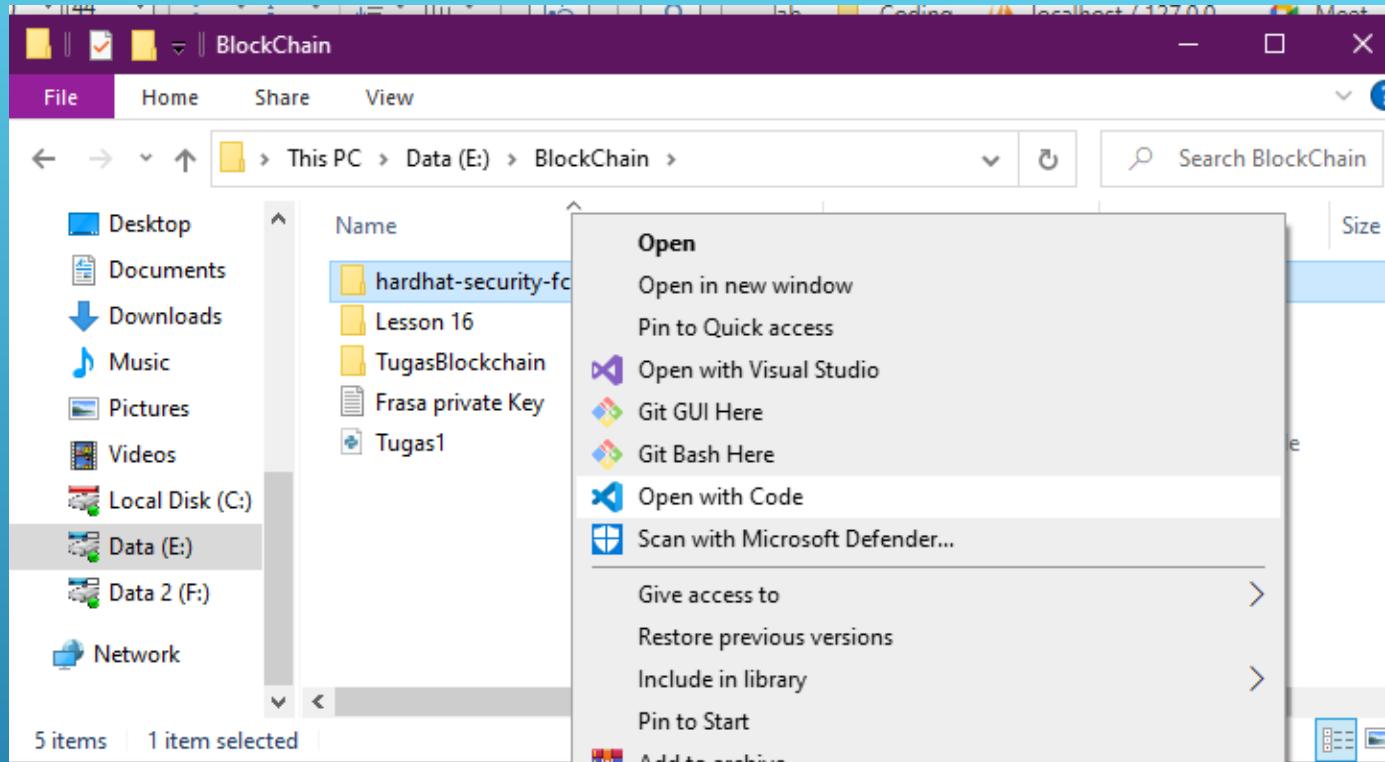
```
PS E:\BlockChain\hardhat-dao-fcc> yarn hardhat deploy
yarn run v1.22.17
warning package.json: No license field
$ E:\BlockChain\hardhat-dao-fcc\node_modules\.bin\hardhat deploy
Nothing to compile
No need to generate any newer typings.
Deploying Governance Token...
deploying "GovernanceToken" (tx: 0x6755c235273eae4735db03594be3c08fbe59f167b3ae4be2d1112a9973388561)...: deployed at 0x5FbDB2315678afecb367f032d93F642f64180aa3 with 3376648 gas
Deployed governance token to address 0x5FbDB2315678afecb367f032d93F642f64180aa3
Checkpoints 1
Delegated!
Deploying Timelock...
deploying "TimeLock" (tx: 0x8230faa079de6656f3ae1b4199c107d9ddd7e0623823701fcac09dc05d634370)...: deployed at 0x9fE46736679d2D9a65F0992F2272dE9f3c7fa6e0 with 2684060 gas
Deploying governor
deploying "GovernorContract" (tx: 0x6bd36bd7451cb0e8099364cb9165b7d659695efba040718452de48599a7e48d1)...: deployed at 0xCf7Ed3AccA5a467e9e704C703E8D87F634fB0Fc9 with 4690056 gas
Setting up roles...
Deploying Box...
deploying "Box" (tx: 0xfb0cce71d18a6e32d4e789bd30caeab46cebe24ba7d50924af8a870c2429e)...: deployed at 0xa513E6E4b8f2a923D98304ec87F64353C4D5C853 with 445808 gas
YOU DUN IT!!!
```

LALU KITA DEPLOY DENGAN YARN
HARDHAT DEPLOY, MAKA AKAN KELUAR
SEPERTI INI

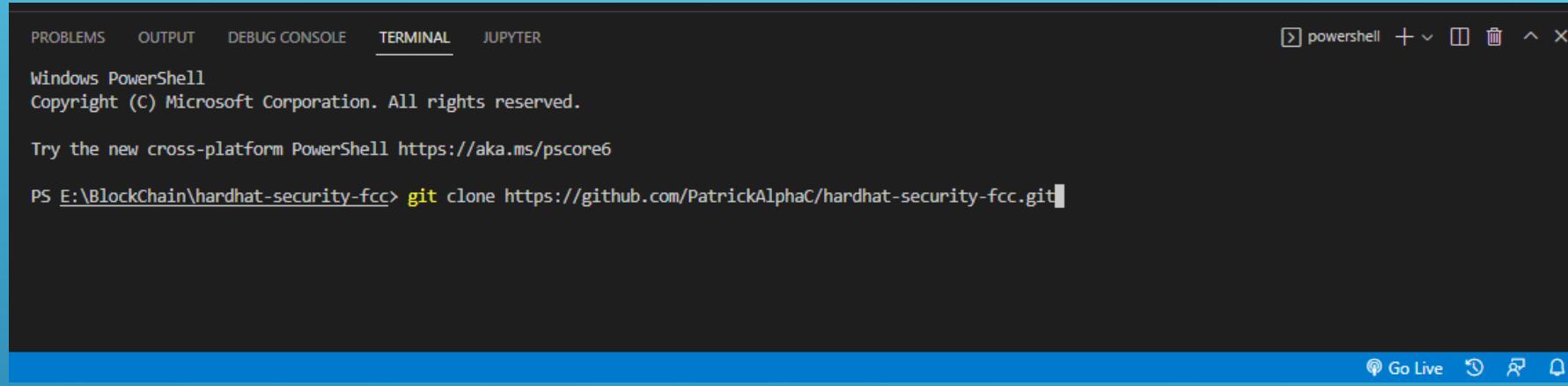
LESSON 18: SECURITY & AUDIT

Slither





BUAT FOLDER HARDHAT-SECURITY-FCC
DAN MASUK KE TERMINAL DALAM FOLDER
HARDHAT-SECURITY-FCC



A screenshot of the Visual Studio Code interface. The top navigation bar includes 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', TERMINAL, and 'JUPYTER'. The terminal tab is active, showing a Windows PowerShell session. The output in the terminal is as follows:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS E:\BlockChain\hardhat-security-fcc> git clone https://github.com/PatrickAlphaC/hardhat-security-fcc.git
```

The status bar at the bottom shows icons for 'Go Live', a timer, a search icon, and a notifications icon.

MASUK KE VS CODE DAN CLONE CODE DI
GIT
[HTTPS://GITHUB.COM/PATRICKALPHAC/HARDHAT-SECURITY-FCC.GIT](https://github.com/PatrickAlphaC/hardhat-security-fcc.git) SEPERTI YANG
ADA DI VIDEO

```
Cloning into '.'...
remote: Enumerating objects: 79, done.
remote: Counting objects: 100% (79/79), done.
remote: Compressing objects: 100% (61/61), done.
remote: Total 79 (delta 34), reused 48 (delta 14), pack-reused 0
Receiving objects: 100% (79/79), 168.40 KiB | 3.37 MiB/s, done
Resolving deltas: 100% (34/34), done.
```

JIKA CLONE BERHASIL MA
MUNCUL SEPERTI INI DAN TILLE-TILLE TERSEDOT

The screenshot shows a Visual Studio Code interface. On the left, the Explorer sidebar displays a project structure with files like package.json, .gitignore, .npmignore, .prettierrc, .solhint.json, .solhintignore, hardhat.config.js, helper-functions.js, helper-hardhat-config.js, LICENSE, README.md, and yarn.lock. The package.json file is open in the main editor area, showing its contents. On the right, a terminal window is open, displaying the same cloning log as the one on the left. The status bar at the bottom of the code editor shows file paths, line numbers, and other development settings.

```
File Edit Selection View Go Run Terminal Help package.json - hardhat-security-fcc - Visual Studio Code
package.json
package.json > ...
1 {
  "devDependencies": {
    "@nomiclabs/hardhat-ethers": "npm:hardhat-deploy-ethers",
    "ethers": "^5.5.4",
    "hardhat": "^2.9.0",
    "hardhat-deploy": "^0.10.5"
  },
  "dependencies": {
    "@chainlink/contracts": "^0.4.0",
    "@nomiclabs/hardhat-etherscan": "^3.0.3",
    "@nomiclabs/hardhat-waffle": "^2.0.3",
    "@openzeppelin/contracts": "^4.5.0",
    "@openzeppelin/hardhat-upgrades": "1.15.0",
    "dotenv": "~16.0.0",
    "hardhat-contract-sizer": "2.5.0",
    "hardhat-gas-reporter": "1.0.8",
    "prettier": "2.5.1",
    "prettier-plugin-solidity": "1.0.0-beta.19",
    "solidity-coverage": "0.7.20"
  },
  "scripts": {
    "slither": "slither . --solc-remaps '@openzeppelin-node_modules/@openzeppelin @chainlink-node_modules/@chainlink' --exc
b.com/PatrickAlphaC/hardhat-security-fcc.git .
Cloning into '.'...
remote: Enumerating objects: 79, done.
remote: Counting objects: 100% (79/79), done.
remote: Compressing objects: 100% (61/61), done.
remote: Total 79 (delta 34), reused 48 (delta 14), pack-reused 0
Receiving objects: 100% (79/79), 168.40 KiB | 3.37 MiB/s, done
Resolving deltas: 100% (34/34), done.
```

```
PS E:\BlockChain\hardhat-security-fcc> python --version  
Python 3.8.5
```

INSTALL PYTHON DAN JIKA SUDAH INSTALL
CEK VERSION PYTHON TERSEBUT (INI
SUDAH PERNAH INSTALL SEBELUMNYA)

```
PS E:\BlockChain\hardhat-security-fcc> pip --version
pip 22.0.3 from c:\users\fath\appdata\local\programs\python\py
thon38-32\lib\site-packages\pip (python 3.8)
```

INSTALL PIP DAN CEK VERSION PIP
TERSEBUT (INI SUDAH PERNAH INSTALL
SEBELUMNYA)

```
PS E:\BlockChain\hardhat-security-fcc> pip3 install solc-select
  Downloading solc_select-0.2.1-py3-none-any.whl (16 kB)
Installing collected packages: solc-select
Successfully installed solc-select-0.2.1
```

LALU PADA PIP KITA INSTALL SOLC-
SELECT

```
PS E:\BlockChain\hardhat-security-fcc> pip3 install slither-analyzer
Collecting slither-analyzer
  Downloading slither_analyzer-0.8.3-py3-none-any.whl (547 kB)
    ━━━━━━━━━━━━━━━━ 547.9/547.9 kB 6.9 MB/s eta 0:00:00
Requirement already satisfied: pysha3>=1.0.2 in c:\users\fath\appdata\local\programs\python\python38-32\lib\site-packages (from slither-analyzer) (1.0.2)
Collecting crytic-compile>=0.2.3
  Downloading crytic_compile-0.2.3-py3-none-any.whl (87 kB)
    ━━━━━━━━━━━━━━ 87.2/87.2 kB 5.1 MB/s eta 0:00:00
Collecting prettytable>=0.7.2
  Downloading prettytable-3.3.0-py3-none-any.whl (26 kB)
Collecting wcwidth
  Downloading wcwidth-0.2.5-py2.py3-none-any.whl (30 kB)
Installing collected packages: wcwidth, prettytable, crytic-compile, slither-analyzer
Successfully installed crytic-compile-0.2.3 prettytable-3.3.0
slither-analyzer-0.8.3 wcwidth-0.2.5
```

LALU KITA INSTALL JUGA SLITHER-ANALYZER

CEK JIKA SLITHER SUDAH TERINSTALL DENGAN BAIK MENGGUNAKAN COMMAND SLITHER –HELP, JIKA BERJALAN DENGAN BAIK DAN TIDAK ADA ERROR MAKA SLITHER TERINSTALL DENGAN BAIK

```
PS E:\BlockChain\hardhat-security-fcc> slither --help
usage: slither target [flag]

target can be:
  - file.sol // a Solidity file
  - project_directory // a project directory. See https://github.com/crytic/crytic-compile#crytic-compile for the supported platforms
    - 0x... // a contract on mainnet
    - NETWORK:0x... // a contract on a different network. Supported networks: mainnet,ropsten,kovan,rinkeby,goerli,tobalab,a,bsc,testnet.bsc,arbi,testnet.arbi,poly,avax,testnet.avax,ftm

For usage information, see
https://github.com/crytic/slither/wiki/Usage

optional arguments:
  -h, --help            show this help message and exit
  --version           displays the current version

Compile options:
  --compile-force-framework COMPILE_FORCE_FRAMEWORK
    Force the compile to a given framework (solc,truffle,embark,dapp,etherlime,etherscan,vyper,waffle,brownie,solc-json,buidler,hardhat,foundry,standard,archive)
  --compile-remove-metadata
    Remove the metadata from the bytetimes
  --compile-custom-build COMPILE_CUSTOM_BUILD
    Replace platform specific build command
  --ignore-compile
    Do not run compile of any platform

Solc options:
  --solc SOLC          solc path
  --solc-remaps SOLC_REMAPS
```

```
Additional options:
  --json JSON          Export the results as a JSON file ("--json -" to export to stdout)
  --sarif SARIF        Export the results as a SARIF JSON file ("--sarif -" to export to stdout)
  --json-types JSON_TYPES
    Comma-separated list of result types to output to JSON, defaults to detectors,printers. Available types: compilations,console,detectors,printers,list-detectors,list-printers
  --zip ZIP            Export the results as a zipped JSON file
  --zip-type ZIP_TYPE
    Zip compression type. One of lzma,stored,deflated,bzip2. Default lzma
  --markdown-root MARKDOWN_ROOT
    URL for markdown generation
  --filter-paths FILTER_PATHS
    Comma-separated list of paths for which results will be excluded
  --triage-mode
    Run triage mode (save results in slither.db.json)
  --config-file CONFIG_FILE
    Provide a config file (default: slither.config.json)
  --solc-ast
    Provide the contract as a json AST
  --generate-patches
    Generate patches (json output only)
```

PS E:\BlockChain\hardhat-security-fcc> **npm install --global yarn**

found 0 vulnerabilities

npm notice

npm notice New minor version of npm available! 8.7.0 → 8.13.2

npm notice Changelog: <https://github.com/npm/cli/releases/tag/v8.13.2>

npm notice Run **npm install -g npm@8.13.2** to update!

npm notice

NPM INSTALL --GLOBAL YARN

```
PS E:\BlockChain\hardhat-security-fcc> yarn --version  
1.22.19
```

LALU KITA CEK VERSION DARI YARN
TERSEBUT, AGAR MENANDAKAN YARN
SUDAH TERINSTALL

```
PS E:\BlockChain\hardhat-security-fcc> yarn  
yarn install v1.22.19  
[1/4] Resolving packages...  
[2/4] Fetching packages...  
[3/4] Linking dependencies...  
warning " > @nomiclabs/hardhat-waffle@2.0.3" has incorrect peer dependency "@nomiclabs/hardhat-ethers@^2.0.0".  
warning " > @nomiclabs/hardhat-waffle@2.0.3" has unmet peer dependency "ethereum-waffle@^3.2.0".  
warning " > @openzeppelin/hardhat-upgrades@1.19.0" has incorrect peer dependency "@nomiclabs/hardhat-ethers@^2.0.0".  
warning " > @openzeppelin/hardhat-upgrades@1.19.0" has incorrect peer dependency "@nomiclabs/hardhat-etherscan@^3.1.0".  
warning " > hardhat-deploy@0.10.5" has unmet peer dependency "@ethersproject/hardware-wallets@^5.0.14".  
[4/4] Building fresh packages...  
Done in 142.14s.
```

LALU KITA HIDUPKAN YARN TERSEBUT
DENGAN COMMAND YARN

LAI DE FILE TER

```
{ package.json x
  package.json > {} scripts > slither
    "hardhat-deploy": "^0.10.5"
  },
  "dependencies": {
    "@chainlink/contracts": "^0.4.0",
    "@nomiclabs/hardhat-etherscan": "^3.0.3",
    "@nomiclabs/hardhat-waffle": "^2.0.3",
    "@openzeppelin/contracts": "^4.5.0",
    "@openzeppelin/hardhat-upgrades": "^1.15.0",
    "dotenv": "^16.0.0",
    "hardhat-contract-sizer": "^2.5.0",
    "hardhat-gas-reporter": "^1.0.8",
    "prettier": "^2.5.1",
    "prettier-plugin-solidity": "^1.0.0-beta.19",
    "solidity-coverage": "^0.7.20"
  },
  ▷ Debug
  "scripts": {
    "slither": "slither --solc-remaps '@openzeppelin=node_modules/@openzeppelin@chainlink=node_modules/@chainlink' --exec",
    "toolbox": "docker run -it --rm -v $PWD:/src trailofbits/eth-security-toolbox",
    "lint": "solhint 'contracts/*.sol'",
    "lint:fix": "solhint 'contracts/**/*.sol' --fix",
    "format": "prettier --write ."
  }
}
29 }
```

```
PS E:\BlockChain\hardhat-security-fcc> slither . -  
ps '@openzeppelin=node_modules/@openzeppelin @chai  
_modules/@chainlink' --exclude naming-convention,e  
nction,low-level-calls  
'npx hardhat compile --force' running  
Downloading compiler 0.8.7  
Compiled 12 Solidity files successfully
```

JIKA BERHASIL AKAN MUNCUL SEPERTI INI,
NANTINYA SLITHER AKAN MENGANALISA
SEMUA CONTRACT YANG ADA DI FOLDER
INI.

Reentrancy in EtherStore.withdraw() (contracts/Reentrancy.sol#15-21):
External calls:
- (success) = msg.sender.call{value: balances[msg.sender]} (contracts/Reentrancy.sol#18)
State variables written after the call(s):
- balances[msg.sender] = 0 (contracts/Reentrancy.sol#20)
Reference: <https://github.com/crytic/slither/wiki/documentation#reentrancy-vulnerabilities>

MetamorphicContract.owner (contracts/MetamorphicContract.sol#6) is never initialized. It is used in:
- MetamorphicContract.kill() (contracts/MetamorphicContract.sol#8-11)
Reference: <https://github.com/crytic/slither/wiki/documentation#uninitialized-state-variables>

Address.verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#201-221) usually
- INLINE ASM (node_modules/@openzeppelin/contracts/utils/Address.sol#213-216)
Reference: <https://github.com/crytic/slither/wiki/documentation#assembly-usage>

VaultFuzzTest.echidna_test_find_password() (contracts/test/fuzzing/VaultFuzzTest.sol#9-11) compares to a boolean:
- s_locked == true (contracts/test/fuzzing/VaultFuzzTest.sol#10)
Reference: <https://github.com/crytic/slither/wiki/documentation#boolean-equality>

Different versions of Solidity are used:
- Version used: ['0.8.7', '^0.8.0', '^0.8.7']
- ^0.8.0 (node_modules/@openzeppelin/contracts/utils/Initializable.sol#4)
- ^0.8.0 (node_modules/@openzeppelin/contracts/ERC20/ERC20.sol#4)
- ^0.8.0 (node_modules/@openzeppelin/contracts/ERC20/IERC20.sol#4)
- ^0.8.0 (node_modules/@openzeppelin/contracts/ERC20/extensions/IERC20Metadata.sol#4)
- ^0.8.1 (node_modules/@openzeppelin/contracts/Address.sol#4)
- ^0.8.0 (node_modules/@openzeppelin/contracts/Context.sol#4)

- ^0.8.0 (node_modules/@openzeppelin/contracts/Context.sol#4)
- ^0.8.0 (contracts/BadRNG.sol#2)
- ^0.8.0 (contracts/LiquidityPoolAsOracle.sol#4)
- 0.8.7 (contracts/MetamorphicContract.sol#4)
- ^0.8.0 (contracts/Reentrancy.sol#2)
- ^0.8.0 (contracts/Vault.sol#8)
- ^0.8.7 (contracts/test/fuzzing/VaultFuzzTest.sol#4)
Reference: <https://github.com/crytic/slither/wiki/documentation#different-pragma-directives-are-used>

Pragma version^0.8.0 (node_modules/@openzeppelin/croxy/utils/Initializable.sol#4) allows old version
Pragma version^0.8.0 (node_modules/@openzeppelin/coin/ERC20/ERC20.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/coin/ERC20/IERC20.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/coin/ERC20/extensions/IERC20Metadata.sol#4) allows old versions
Pragma version^0.8.1 (node_modules/@openzeppelin/contracts/Address.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/Context.sol#4) allows old versions
Pragma version^0.8.0 (contracts/BadRNG.sol#2) allows old versions
Pragma version^0.8.0 (contracts/LiquidityPoolAsOracle.sol#4) allows old versions
Pragma version^0.8.0 (contracts/Reentrancy.sol#2) allows old versions

MUNCUL SEPERTI

```
Pragma version^0.8.0 (contracts/Reentrancy.sol#2)
versions
Pragma version^0.8.0 (contracts/Vault.sol#8) allow
ions
Reference: https://github.com/crytic/slither/wiki/Documentation#incorrect-versions-of-solidity

BadRNG.enterRaffle() (contracts/BadRNG.sol#11-14)
als with too many digits:
    - require(bool)(msg.value >= 10000000000000000000) (contracts/BadRNG.sol#12)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits

MetamorphicContract.owner (contracts/MetamorphicContract.sol#6) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant
```

TECATAT ADA 17 MASALAH YANG
analyzed (12 contracts with 75 detectors), 17 result(s) found

MAUPUN YANG KECIL

Hasil yang berwarna merah itu berarti adalah issue yang mempunyai high impact jika diperbaiki, maka saat kita mendeploy contract yang sudah diperbaiki dan contract yang belum diperbaiki akan terlihat banyak perbedaan setalah perubahan dan kita akan menemukan issue yang sangat banyak pada contract yang belum diperbaiki.

Sedangkan hasil yang berwarna hijau adalah issu yang mempunyai low impact jika diperbaiki, contohnya seperti version solidity yang berbeda sedikit di semua file, lalu jika ada version solidity yang sudah tua, dan lainnya.

PENJELASAN HASIL ANALISIS SLITHER PADA CONTRACT