

Nama Anggota Kelompok :

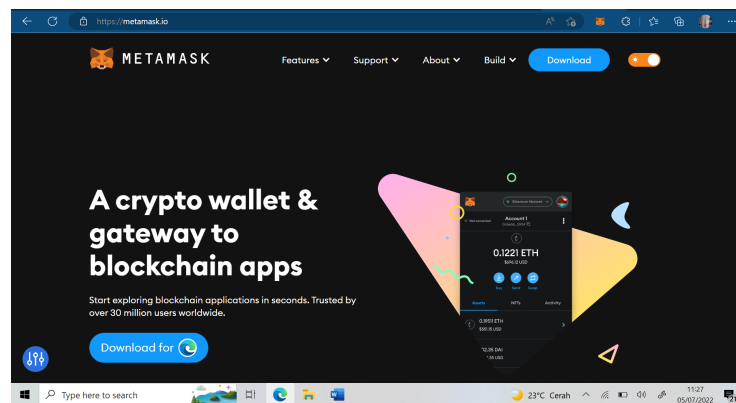
1. Putri Mellia Zahrani (1103190143)
2. Kevin Erico Prasetyo (1103193116)

=====UAS (32 Hour Course)=====

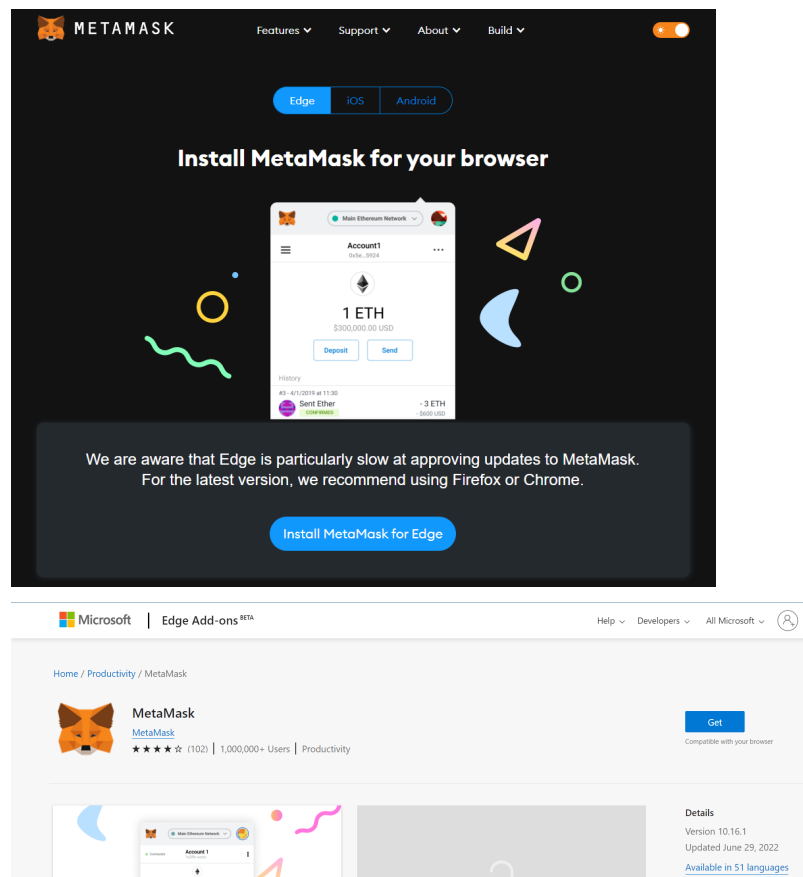
1. First Transaction in Blockchain

Hal yang pertama dilakukan untuk melakukan transaksi di Blockchain adalah melakukan set up wallet. Dibawah ini merupakan langkah- langkah untuk melakukan set up pada wallet :

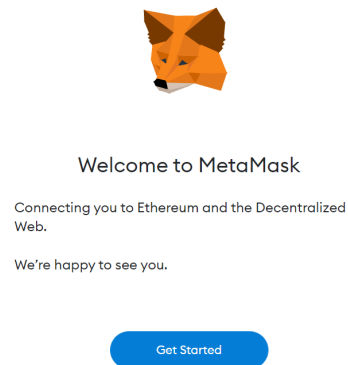
a. Download Metamask



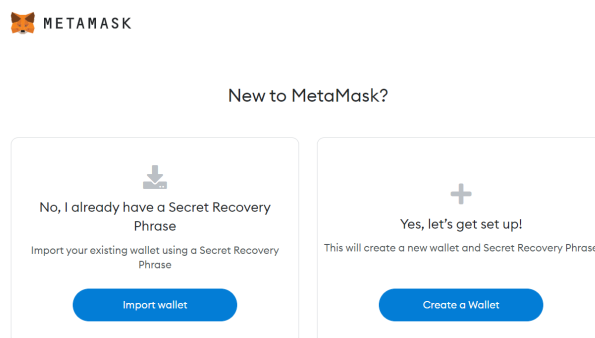
b. Pilih sesuai dengan device yang anda gunakan lalu klik Install Metamask dan klik Get. Lalu pada pop up klik Add Extension.



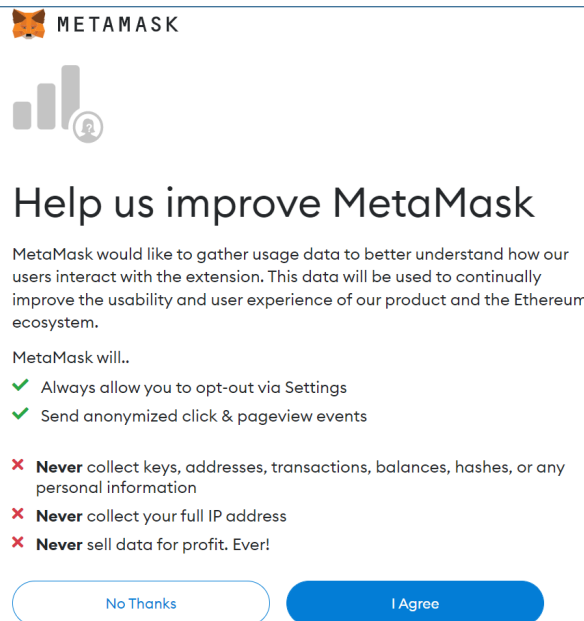
- c. Setelah itu maka akan muncul tampilan seperti di bawah ini, lalu klik Get Started



- d. Jika sudah memiliki wallet maka klik Import walet apabila belum memiliki Wallet maka klik Create a Wallet.



- e. Karena saya belum memiliki wallet maka saya akan Create a Wallet. Setelah mengklik Create a Wallet maka akan muncul tampilan seperti dibawah ini, lalu klik I Agree.



- f. Setelah itu masukan password untuk Wallet Anda dan buatlah password seaman mungkin lalu isi kotak centang dan klik Create.



Create Password

New password (8 characters min)

Confirm password

☐

I have read and agree to the [Terms of Use](#)

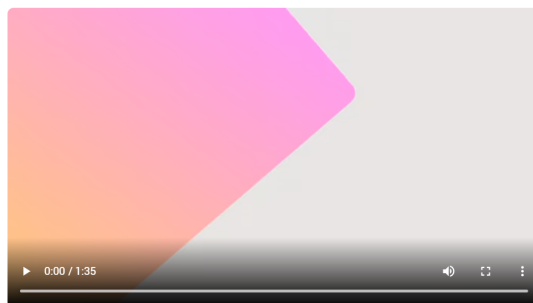
Create

- g. Akan muncul tampilan seperti dibawah yang berisi tentang cara untuk mengamankan akun Metamask anda. Klik Next.



Secure your wallet

Before getting started, watch this short video to learn about your Secret Recovery Phrase and how to keep your wallet safe.



Next

What is a Secret Recovery Phrase?

Your Secret Recovery Phrase is a 12-word phrase that is the "master key" to your wallet and your funds

How do I save my Secret Recovery Phrase?

- Save in a password manager
- Store in a bank vault
- Store in a safe deposit box
- Write down and store in multiple secret places

Should I share my Secret Recovery Phrase?

Never, ever share your Secret Recovery Phrase, not even with MetaMask!

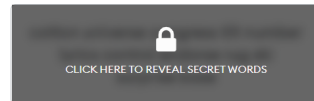
If someone asks for your recovery phrase they are likely trying to scam you and steal your wallet funds.

- h. Setelah itu akan muncul tampilan seperti di bawah ini yang berisi tentang kata kunci jika ingin melakukan recovery akun. Kata kunci tidak boleh diketahui siapapun selain pengguna karena ketika orang lain tau maka orang lain akan mendapatkan seluruh akses Wallet ini.

Secret Recovery Phrase

Your Secret Recovery Phrase makes it easy to back up and restore your account.

WARNING: Never disclose your Secret Recovery Phrase. Anyone with this phrase can take your Ether forever.



[Remind me later](#)

[Next](#)

Tips:

Store this phrase in a password manager like 1Password.

Write this phrase on a piece of paper and store in a secure location. If you want even more security, write it down on multiple pieces of paper and store each in 2 - 3 different locations.

Memorize this phrase.

[Download this Secret Recovery Phrase and keep it stored safely on an external encrypted hard drive or storage medium.](#)

- i. Setelah anda membuka kata kunci dan mengklik next maka tampilan akan berubah seperti pada gambar dibawah dan Anda diminta untuk mengkonfirmasi kata kunci yang Anda dapatkan sebelumnya. Jika, sudah selesai mengisi maka klik Confirm.

Confirm your Secret Recovery Phrase

Please select each phrase in order to make sure it is correct.

cotton	universe	congress	tilt
number	lyrics	control	endorse
rug	ski	surprise	base

base	congress	control	cotton
endorse	lyrics	number	rug
ski	surprise	tilt	universe

[Confirm](#)

- j. Pembuatan akun metamask selesai , lalu klik All Done.



Congratulations

You passed the test - keep your Secret Recovery Phrase safe, it's your responsibility!

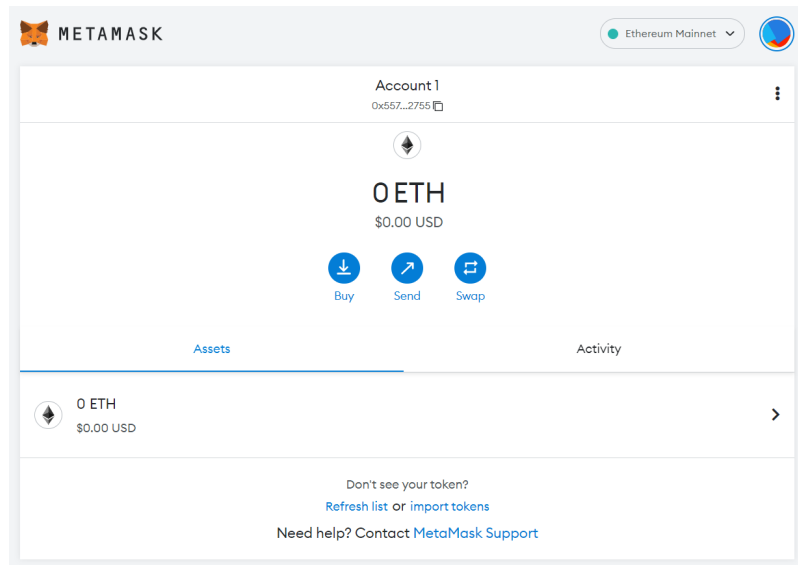
Tips on storing it safely

- Save a backup in multiple places.
- Never share the phrase with anyone.
- Be careful of phishing! MetaMask will never spontaneously ask for your Secret Recovery Phrase.
- If you need to back up your Secret Recovery Phrase again, you can find it in Settings -> Security.
- If you ever have questions or see something fishy, contact our support [here](#).

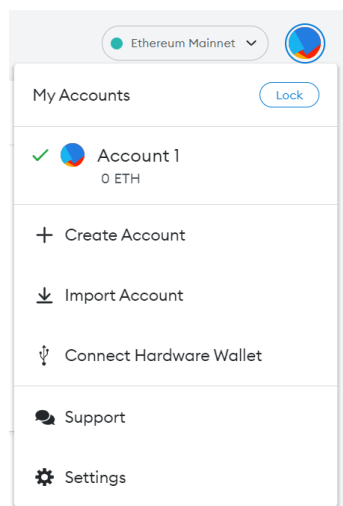
*MetaMask cannot recover your Secret Recovery Phrase. [Learn more](#).

[All Done](#)

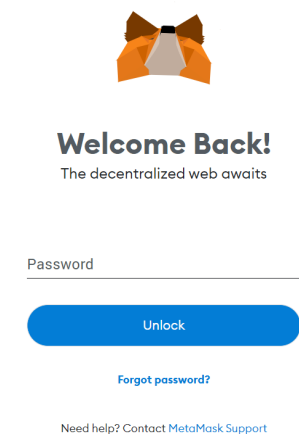
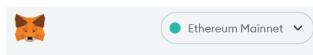
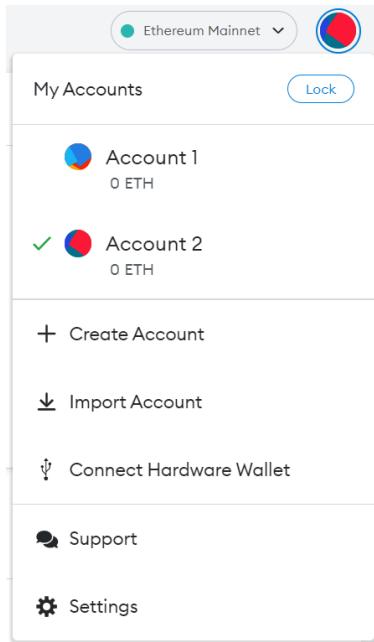
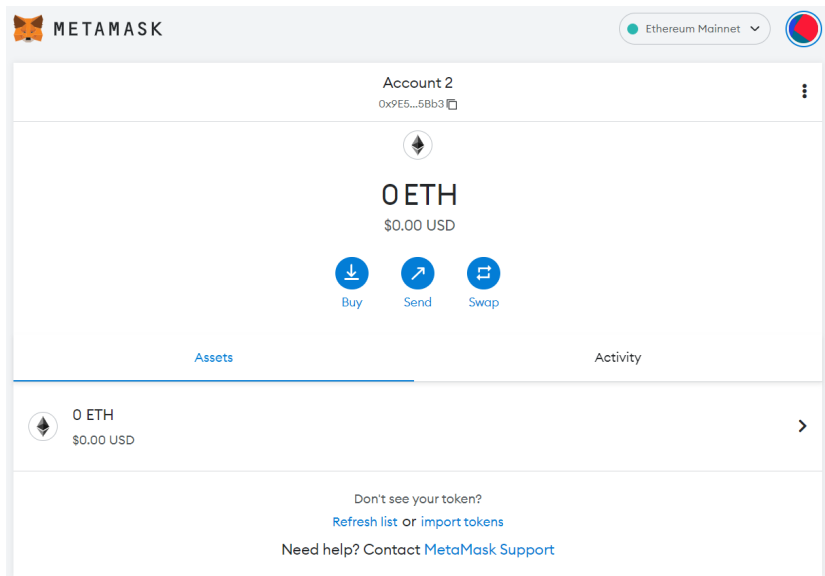
- k. Setelah selesai maka Anda dapat melihat tampilan Wallet Anda



Setelah membuat akun wallet anda bisa membuat account kedua dengan cara mengklik logo akun anda pada pojok kanan atas lalu klik Create Account. Isi nama account anda lalu klik create.



Di bawah ini adalah tampilan dari account kedua Anda :



Dengan mengisi password di atas maka anda dapat langsung mengakses 2 akun yang anda buat di atas. Oleh karena itu, Anda harus merahasiakan password Anda seaman mungkin.

Pada metamask terdapat 3 jenis password yaitu :

1. Mnemonic

Ketika orang lain mengetahui password Mnemonic Anda maka orang tersebut dapat mengakses seluruh akun Metamask yang anda miliki.

2. Private Key

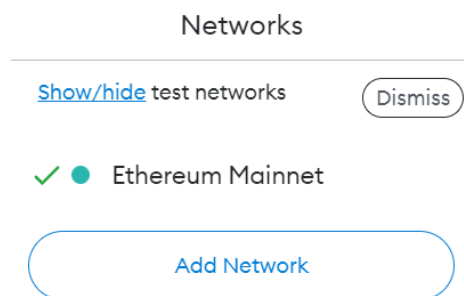
Ketika private key Anda diketahui orang lain maka Anda akan kehilangan account dengan private key tersebut.

3. Public Key

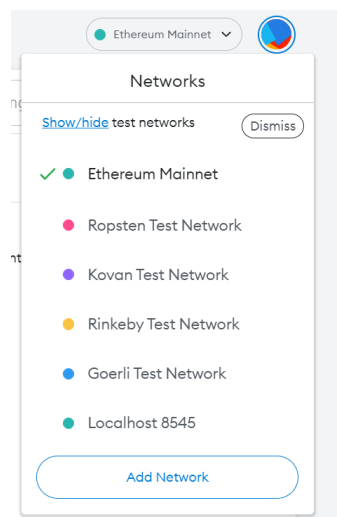
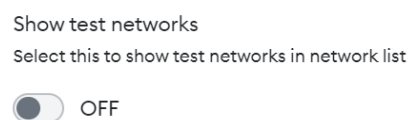
Jika public key Anda diketahui orang lain maka tidak masalah karena dengan public key orang tidak dapat mengakses akun Anda.

Setelah selesai membuat akun kita akan melanjutkan untuk melakukan transaksi pada Ethereum. Berikut adalah langkah-langkah yang harus dilakukan :

1. Ubah Network Metamask menjadi Rinkeby Network. Klik logo akun pada pojok kanan atas lalu klik “Show/hide test network”.



Ubah Show test networks menjadi ON, lalu klik Ethereum Mainnet dan klik Rinkeby Network.

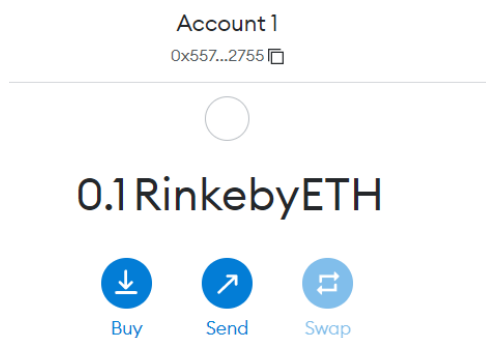


2. Lalu buka <https://faucets.chain.link/>

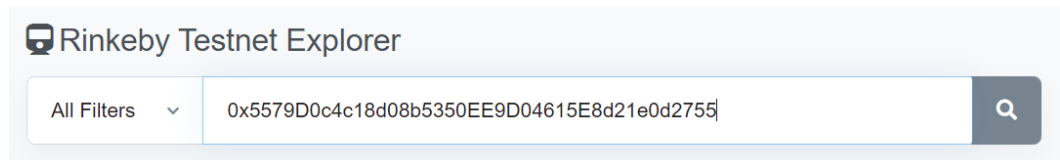


The screenshot shows the Chainlink Faucets website. At the top, there's a navigation bar with the Chainlink logo and 'Faucets' text. On the right, there are dropdown menus for 'Ethereum Rinkeby' and a wallet address '0x5579...2755'. Below the navigation bar, the main content area is titled 'Wallet address' and shows a text input field containing '0x9E54b26935D1E77f951C5cA3b2C645Cb5DfF5Bb3'. Underneath, there's a 'Request type' section with two buttons: '20 test LINK' (disabled) and '0.1 test ETH' (selected). Below that is a Captcha section with a checkbox labeled 'I am human' and the hCaptcha logo. At the bottom, there's a 'Send request' button.

Wallet address akan otomatis terisi dengan address wallet yang sudah terkoneksi, lalu klik 0,1 test ETH dan lakukan verifikasi Captcha lalu klik Send request. Tunggu hingga proses selesai dan kembali ke metamask. Jika transaksi berhasil maka 0 ETH akan berubah menjadi 0,1 ETH.



Untuk melihat informasi transaksi secara detail maka Anda dapat mengakses [TESTNET Rinkeby \(ETH\) Blockchain Explorer \(etherscan.io\)](https://etherscan.io) dan masukkan address wallet Anda lalu klik ikon Search dan akan muncul tampilan seperti dibawah ini :



Etherscan
Rinkeby Testnet Network

Address: 0x5579D0c4c18d08b5350EE9D04615E8d21e0d2755

Overview
Balance: 0.1 Ether

More Info
My Name Tag: Not Available

Transactions | Erc20 Token Txns

Latest 1 from a total of 1 transactions

Txn Hash	Method	Block	Age	From	To	Value	Txn Fee
0x2d4960f4bcf311b7115f...	Transfer	10974990	33 mins ago	0xa7a82dd06901f29ab1...	0x5579d0c4c18d08b535...	0.1 Ether	0.0000525

[Download CSV Export]

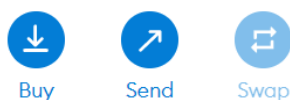
Note :

Rinkeby Network hanya digunakan untuk melakukan testing, jika Anda ingin melakukan real transaction maka anda harus menggunakan Ethereum Mainnet.

Selain itu kita juga dapat melakukan transfer ETH berikut adalah contohnya :

Klik send pada halaman utama wallet Anda :

0.1 RinkebyETH



Lalu akan muncul tampilan seperti dibawah lalu pilih kemana Anda akan melakukan transfer ETH tersebut :

Send to

Cancel

Q

Search, public address (0x), or ENS

Transfer between my accounts

Karena Saya akan melakukan transfer ke akun kedua saya maka pilih Transfer between my accounts, jika Anda ingin mentransfer ke orang lain maka isi dengan public address. Setelah itu pilih account 2 sebagai tujuan.


Send

✓ Account 2

0x9e54b26935d1e77f951c5ca3b2c645cb5dff5bb3

✕

Asset:

 RinkebyETH

Balance: 0.1 RinkebyETH

Amount:

0.05 RinkebyETH

No Conversion Rate Available

↕


Max

Cancel


Next

Lalu masukan nominal yang akan di transfer dan klik Next.

< Edit

 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](#)

✕

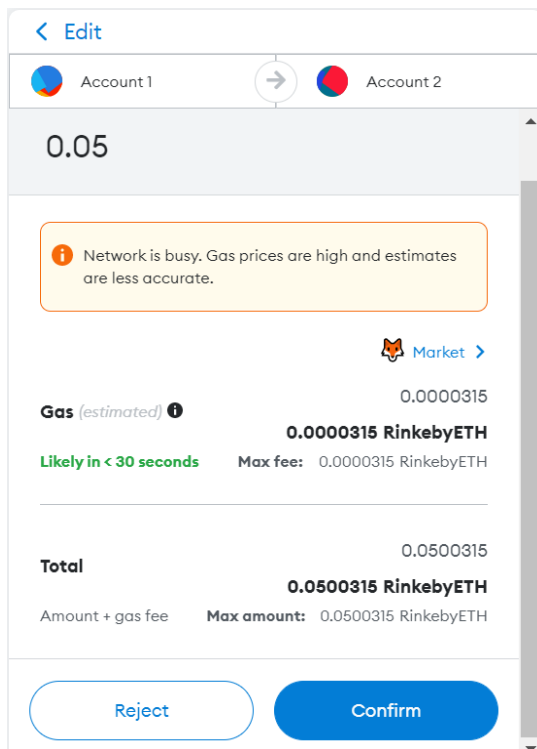
Lalu klik Turn on Enhanced Gas Fee UI in Settings dan Anda akan diarahkan ke menu Settings lalu ubah Pengaturan menjadi ON.

Enable Enhanced Gas Fee UI

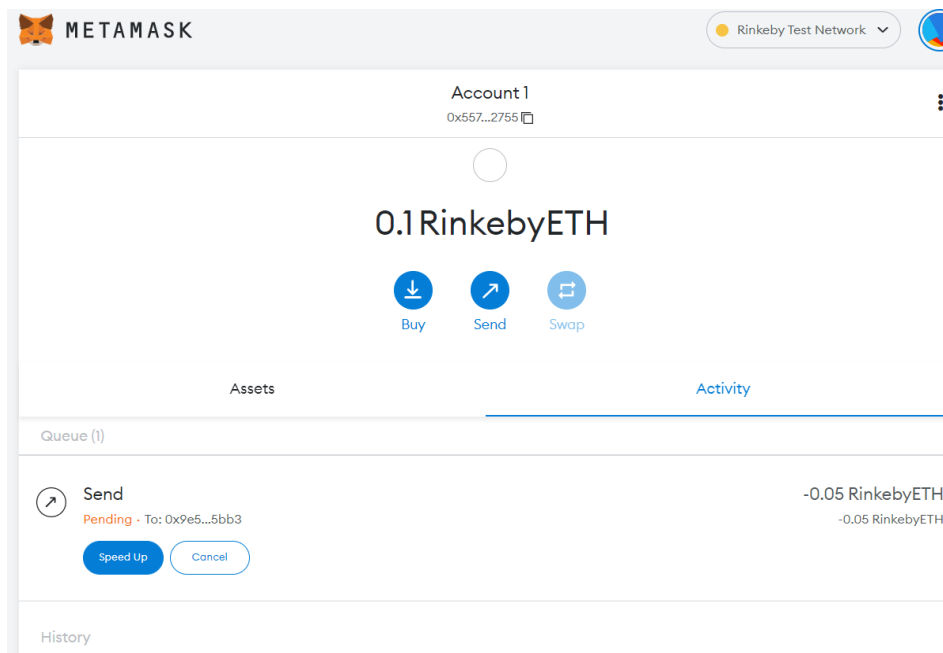
We've updated how gas estimation and customization works. Turn on if you'd like to use the new gas experience. [Learn more](#)

☐ OFF

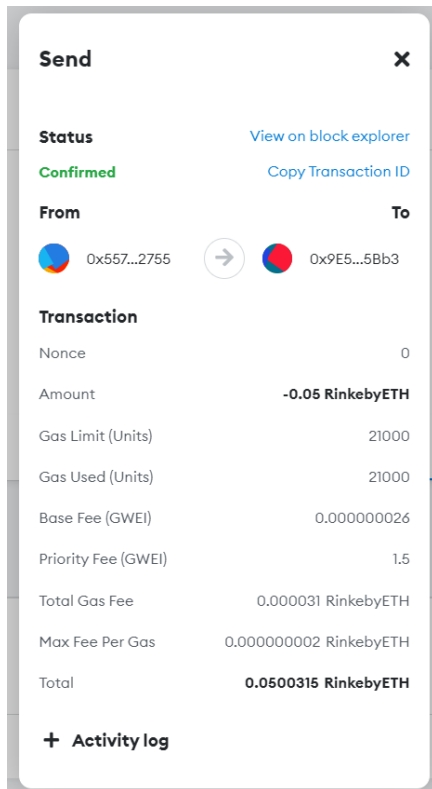
Lalu tampilan akan berubah menjadi seperti di bawah ini lalu klik Confirm.



Tampilan pada wallet account 1 akan menjadi seperti di bawah ini :



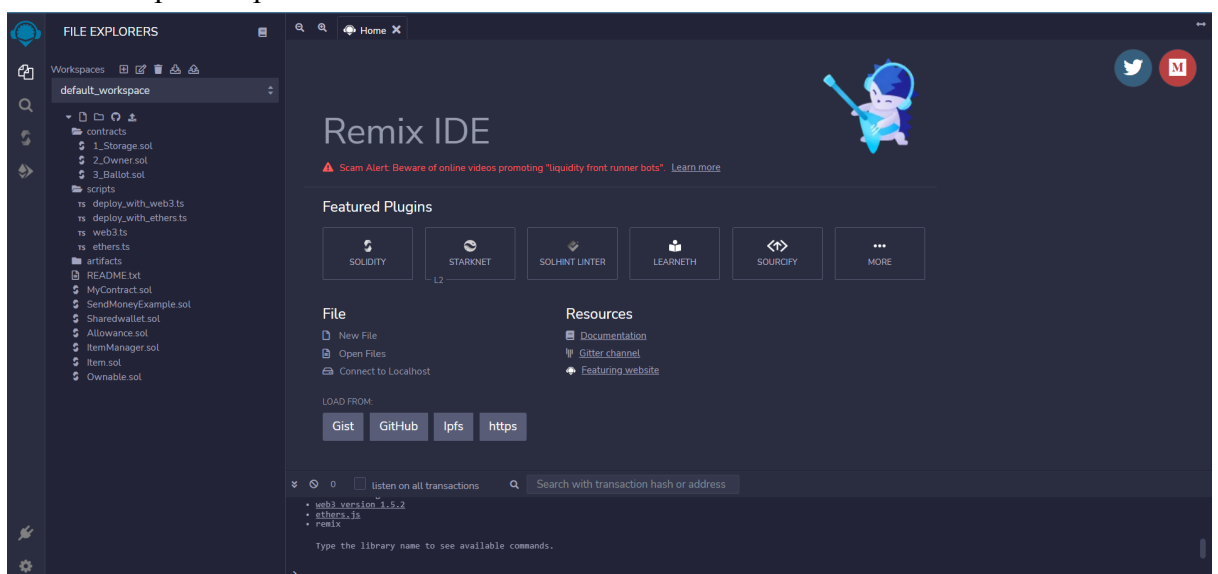
Pengiriman membutuhkan waktu sehingga status pengiriman di atas pending. Tunggu hingga proses Success.



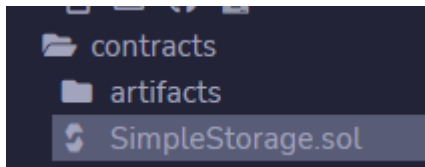
Gambar di atas menunjukkan detail pengiriman. Jika kita lihat total ETH yang dikeluarkan tidak 0.05 melainkan 0.0500315. Setiap transaksi pada Ethereum akan dikenakan fee. Fee tergantung pada tingkat kesulitan transaksi yang dilakukan dan banyaknya Gas yang terpakai. Semakin sulit transaksi dan semakin banyak Gas yang digunakan maka fee nya akan semakin besar.

2. Simple Storage

Hal yang pertama dilakukan untuk membuat sebuah penyimpanan sederhana adalah membuka <https://remix.ethereum.org/> . Setelah membuka link tersebut maka akan muncul tampilan seperti di bawah ini :



Setelah itu hapus semua folder kecuali contracts. Lalu hapus file yang ada pada folder contracts dan buat file baru bernama SimpleStorage.sol.



Lalu masukan code di bawah ini kedalam Smart Contracts Anda.

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

pragma solidity ^0.8.8;

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;
    }

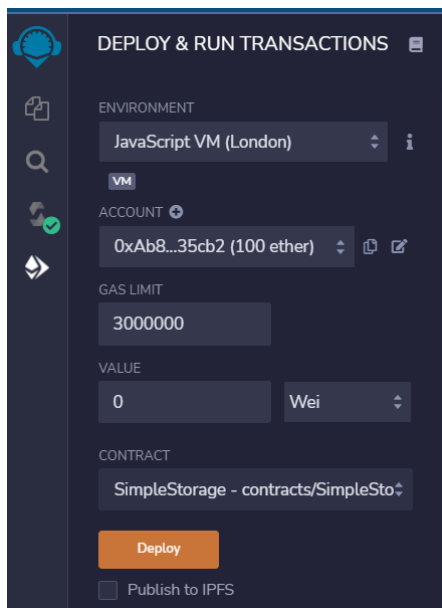
    function retrieve() public view returns (uint256){
        return favoriteNumber;
    }

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

```
}
```

Setelah menuliskan kode di atas kita dapat melakukan testing menggunakan akun metamask.

Jika Anda tidak ingin menggunakan akun Ethereum asli Anda, Anda dapat *men-deploy* menggunakan pengaturan dibawah ini. Jika Anda ingin langsung melakukan testing menggunakan akun Anda maka ubah environment menjadi “Inject Web 3” lalu hubungkan dengan Metamask Anda.



DEPLOY & RUN TRANSACTIONS

ENVIRONMENT
JavaScript VM (London)

VM

ACCOUNT
0xAb8...35cb2 (100 ether)

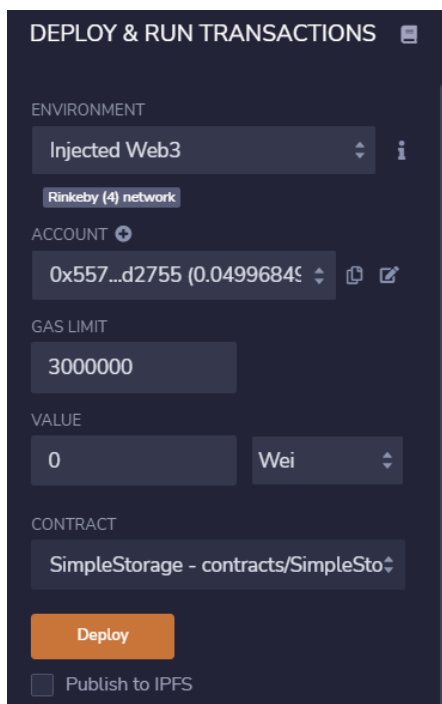
GAS LIMIT
3000000

VALUE
0 Wei

CONTRACT
SimpleStorage - contracts/SimpleSto

Deploy

☐ Publish to IPFS



DEPLOY & RUN TRANSACTIONS

ENVIRONMENT
Injected Web3

Rinkeby (4) network

ACCOUNT
0x557...d2755 (0.0499684)

GAS LIMIT
3000000

VALUE
0 Wei

CONTRACT
SimpleStorage - contracts/SimpleSto

Deploy

☐ Publish to IPFS

Setelah itu klik deploy dan akan muncul pop up seperti di bawah ini :

The screenshot shows the 'CONTRACT DEPLOYMENT' confirmation window in the Remix IDE. At the top, it displays 'Account 1' and a 'New Contract' button. Below this is the URL 'https://remix.ethereum.org' and a 'CONTRACT DEPLOYMENT' button. The window has two tabs: 'DETAILS' (selected) and 'DATA'. Under 'DETAILS', there is a warning box stating 'Network is busy. Gas prices are high and estimates are less accurate.' Below the warning, it shows 'Gas (estimated)' as '0.00115918' and '0.00115918 RinkebyETH'. A green status indicator says 'Very likely in < 15 seconds'. The 'Max fee' is also '0.00115918 RinkebyETH'. The 'Total' cost is '0.00115918' and '0.00115918 RinkebyETH'. At the bottom, there are 'Reject' and 'Confirm' buttons.

Account 1 New Contract

https://remix.ethereum.org

CONTRACT DEPLOYMENT

DETAILS DATA

Network is busy. Gas prices are high and estimates are less accurate.

Site suggested

Gas (estimated) 0.00115918

0.00115918 RinkebyETH

Very likely in < 15 seconds

Max fee: 0.00115918 RinkebyETH

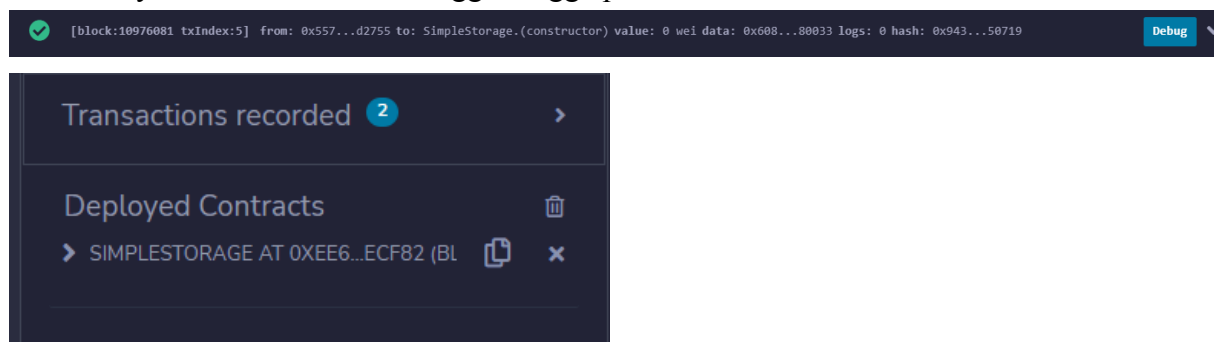
Total 0.00115918

0.00115918 RinkebyETH

Amount + gas fee Max amount: 0.00115918 RinkebyETH

Reject Confirm

Ingat bahwa setiap transaksi maka akan dikenakan biaya Gas sesuai dengan tingkat kesulitannya. Klik confirm dan tunggu hingga proses selesai.



Jika Anda mengcopy alamat pada Simple Storage dan memasukkannya ke [TESTNET Rinkeby \(ETH\) Blockchain Explorer \(etherscan.io\)](#) maka kita dapat melihat contract yang kita buat.

Etherscan Rinkeby Testnet Network

Contract 0xEe6eAbD03166ED8C8A4C301CcA22d5fca7aeCF82

Contract Overview

Balance: 0 Ether

More Info

My Name Tag: Not Available

Contract Creator: 0x5579d0c4c18d08b535... at txn 0x31d9ac776b5acb6fee8...

Transactions Erc20 Token Txns Contract Events

Latest 1 from a total of 1 transactions

Txn Hash	Method	Block	Age	From	To	Value	Txn Fee
0x31d9ac776b5acb6fee8...	0x60806040	10976081	3 mins ago	0x5579d0c4c18d08b535...	IN Contract Creation	0 Ether	0.00115917

[Download CSV Export]

Deployed Contracts

SIMPLESTORAGE AT 0XEE6...ECF82

- addPerson string _name, uint256 _fav
- store uint256 _favoriteNumber
- nameToFavorit... string
- people uint256
- retrieve

Ketika Anda mengklik kotak yang berwarna biru maka Metamask tidak akan memunculkan pop up karena merupakan Blue View Function. Kita akan mencoba mengisi store dengan angka 18 maka Metamask akan memunculkan pop up dan kita perlu mengkonfirmasi transaksi tersebut.

DEPLOY & RUN TRANSACTIONS

SimpleStorage - contracts/SimpleStorage.sol

Deploy

At Address Load contract from Address

Transactions recorded

Deployed Contracts

SIMPLESTORAGE AT 0XEE6...ECF82

- addPerson string _name, uint256 _fav
- store 18
- nameToFavorit... string
- people uint256
- retrieve

Low level interactions

CALLDATA

Transact

Both 'receive' and 'fallback' functions are not defined

```

1
2
3
4
5
6 contract SimpleStorage {
7     uint256 favoriteNumber;
8
9     struct People {
10         uint256 favoriteNumber;
11         string name;
12     }
13
14     // uint256[] public anArray;
15     People[] public people;
16
17     mapping(string => uint256) public nameToFavoriteNumber;
18
19     function store(uint256 _favoriteNumber) public {
20         favoriteNumber = _favoriteNumber;
21     }
22
23     function retrieve() public view returns (uint256){
24         return favoriteNumber;
25     }
26
27     function addPerson(string _name, uint256 _favoriteNumber) public {
28         people.push(People(_favoriteNumber, _name));
29     }
30 }

```

StructDefinition People 3 references

Creation of SimpleStorage pending...

view on etherscan

[block:19976081 txindex:5] from: 0x557...d2755 to: SimpleStorage.(constructor) value: 0 wei data: 0x608...800

transact to SimpleStorage.store pending ...

Account 1 New Contract

https://remix.ethereum.org

CONTRACT INTERACTION

DETAILS DATA

Network is busy. Gas prices are high and estimates are less accurate.

Site suggested

Gas (estimated) 0.00115918 RinkebyETH

Very likely in < 15 seconds

Max fee: 0.00115918 RinkebyETH

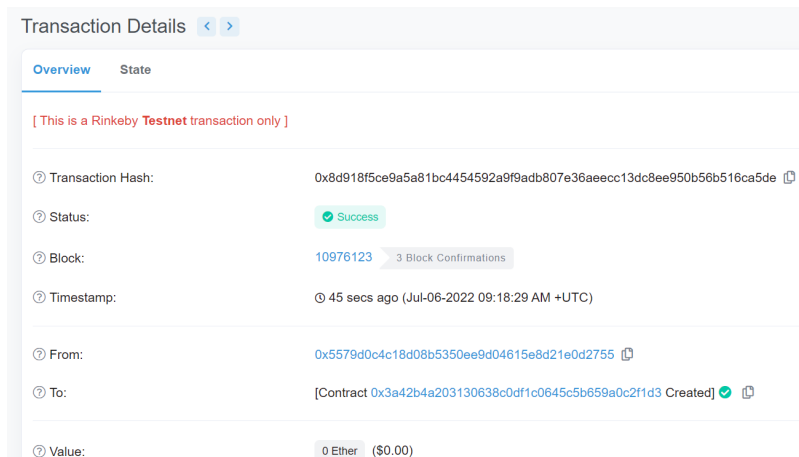
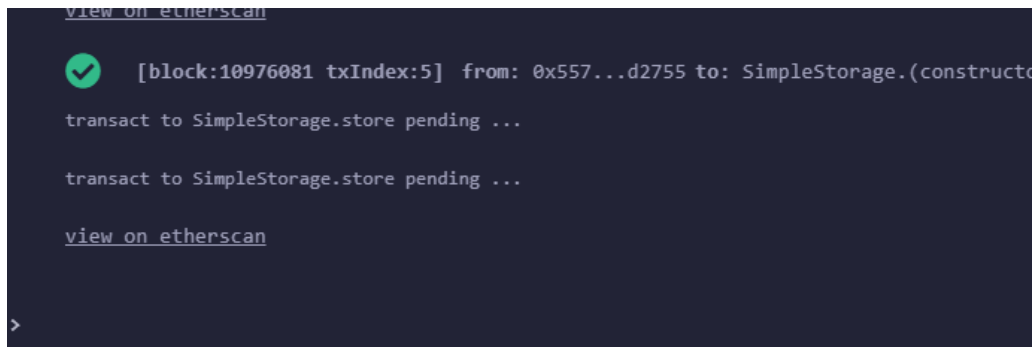
Total 0.00115918

0.00115918 RinkebyETH

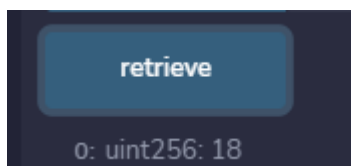
Amount + gas fee 0.00115918 RinkebyETH

Reject Confirm

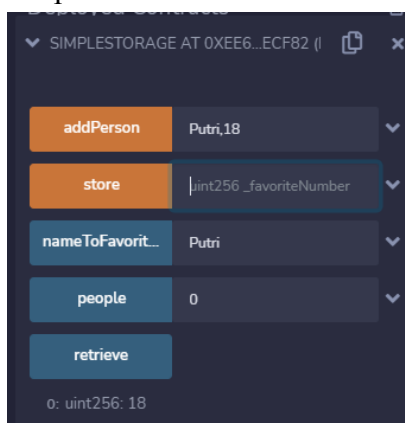
Anda dapat melihat detail transaksi maka Anda dapat mengklik “view on etherscan”.



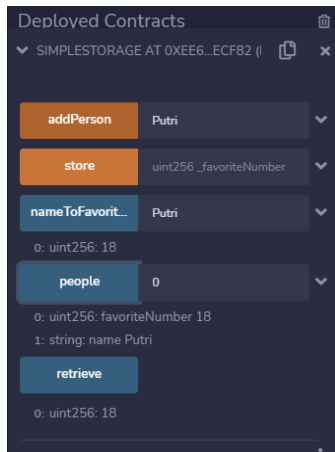
Dan jika kita mengklik retrieve pada Remix maka akan muncul seperti di bawah ini dan berarti bahwa 18 berhasil di store.



Kita akan melakukan testing kembali untuk mengecek apakah Contract sudah sempurna.

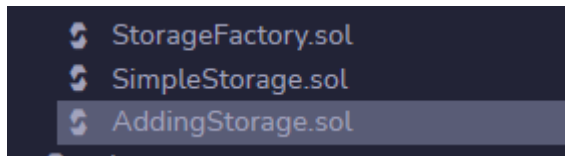


Klik addPerson dan akan muncul popup Metamask lalu klik Confirm dan tunggu hingga proses transaksi selesai. Klik nameToFavorite dan people maka akan tampil seperti di bawah ini dan berarti bahwa Smart Contract Anda sempurna.



3. Remix Storage Factory

Hal yang pertama membuat file storagefactory, SimpleStorage, dan Adding Storage, pada folder artifact



Kemudian masukkan Codingan berikut

```
// SPDX-License-Identifier: MIT

pragma solidity ^0.8.0;

import "./SimpleStorage.sol";

contract StorageFactory {

    SimpleStorage[] public simpleStorageArray;

    function createSimpleStorageContract() public {
        SimpleStorage simpleStorage = new
SimpleStorage();
        simpleStorageArray.push(simpleStorage);
    }

    function sfStore(uint256 _simpleStorageIndex, uint256
_simpleStorageNumber) public {
        // Address
        // ABI
    }
}
```

```

//
SimpleStorage(address(simpleStorageArray[_simpleStorageIndex])).store(_simpleStorageNumber);

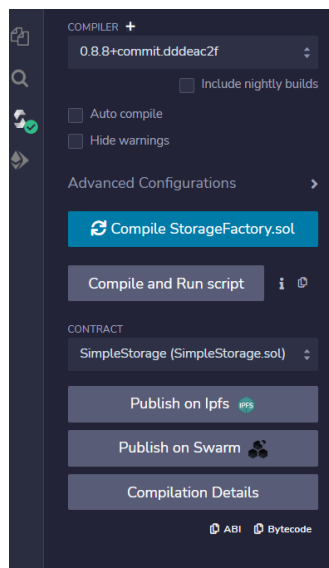
simpleStorageArray[_simpleStorageIndex].store(_simpleStorageNumber);
    }

    function sfGet(uint256 _simpleStorageIndex) public view returns (uint256) {
        // return
SimpleStorage(address(simpleStorageArray[_simpleStorageIndex])).retrieve();

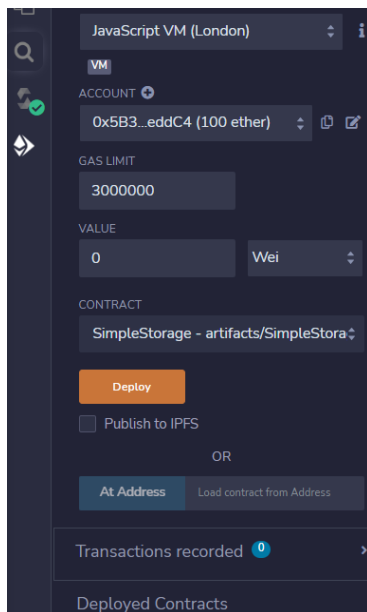
        return
simpleStorageArray[_simpleStorageIndex].retrieve();
    }
}

```

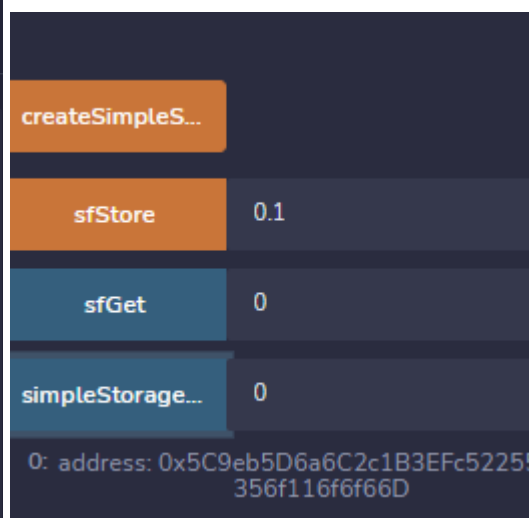
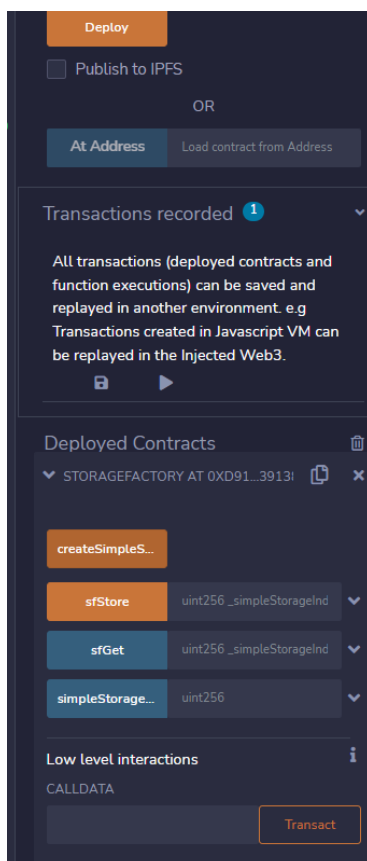
Lalu masuk ke solidity compiler dan lakukan compiler pada storage Factory



Setelah itu masuk ke menu Deploy & Run Transactions, dan lakukan DEPLOY pada Storage Factory



Lalu setelah selesai Deploy lakukan Create Simple Factory, sekaligus isi tabel sesuai contoh berikut



Setelah selesai, kita sudah sampai pada tahap fitur soliditas seperti mengimpor penyimpanan.

Selanjutnya kita akan belajar mengimpor kontrak, ke kontrak lain
Pertama copy & paste, kodingan berikut ke SimpleStorage.sol di Remix.

```

// 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;
    }
}

```

Kemudian sebelum kita memulai kontrak, kita buat storagefactory dan setting kodingan sesuai contoh berikut.

```
StorageFactory.sol
SimpleStorage.sol
.deps

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

contract StorageFactory {

    function createSimpleStorageContract() public {

    }

}
```

Kemudian tambahkan kodingan berikut dibawah pragma solidity.

```
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;
    }

}
```

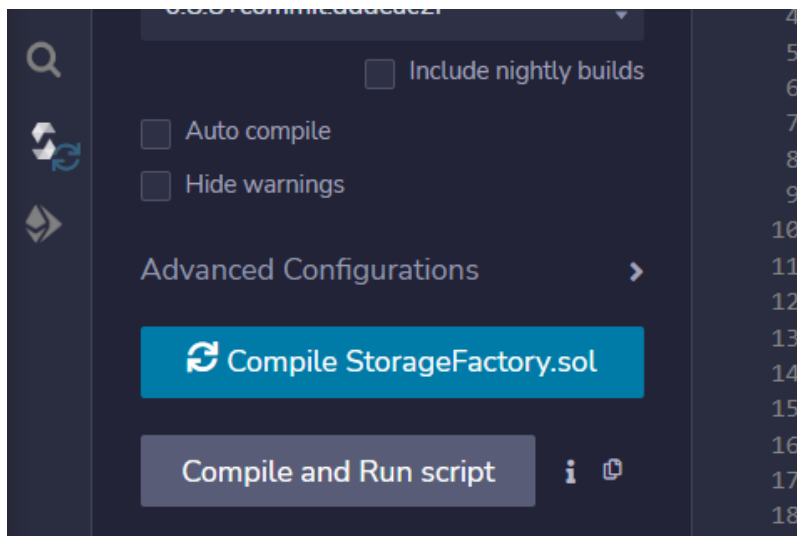
```

    }

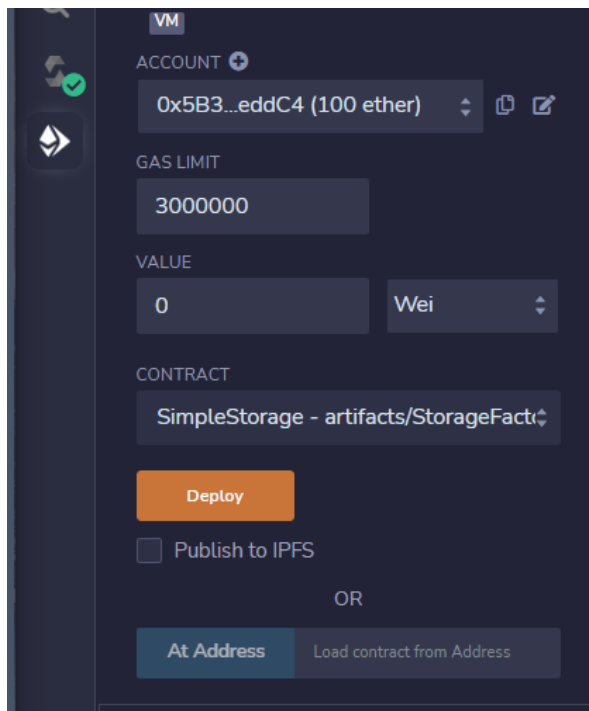
    function addPerson(string memory _name, uint256
_favoriteNumber) public {
        people.push(People(_favoriteNumber, _name));
        nameToFavoriteNumber[_name] = _favoriteNumber;
    }
}

```

Kemudian lakukan Compile dan save



Kemudian masuk pada tab menu Deploy



Kemudian tambahkan kodingan pada StorageFactory.sol sesuai contoh berikut

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

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;
    }

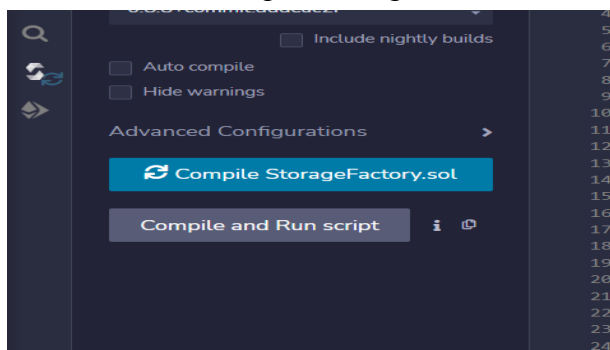
    function addPerson(string memory _name, uint256
_favoriteNumber) public {
        people.push(People(_favoriteNumber, _name));
        nameToFavoriteNumber[_name] = _favoriteNumber;
    }
}

contract StorageFactory {
    SimpleStorage public simpleStorage;

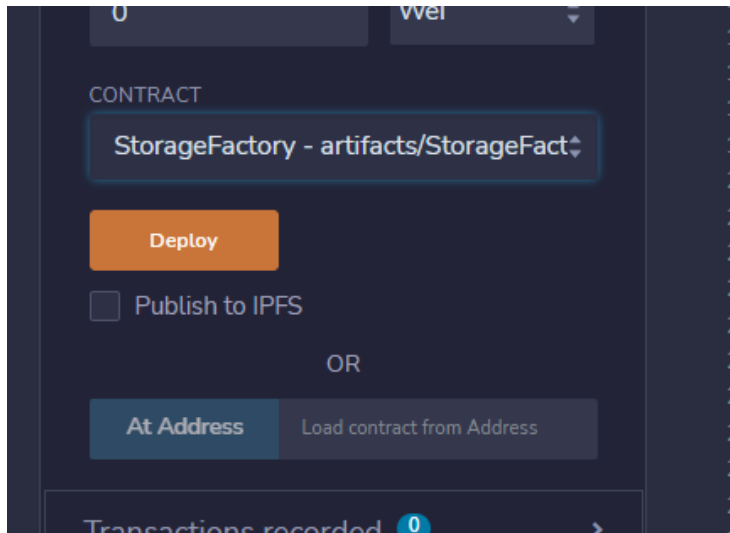
    function createSimpleStorageContract() public {
        simpleStorage = new SimpleStorage();
    }
}

```

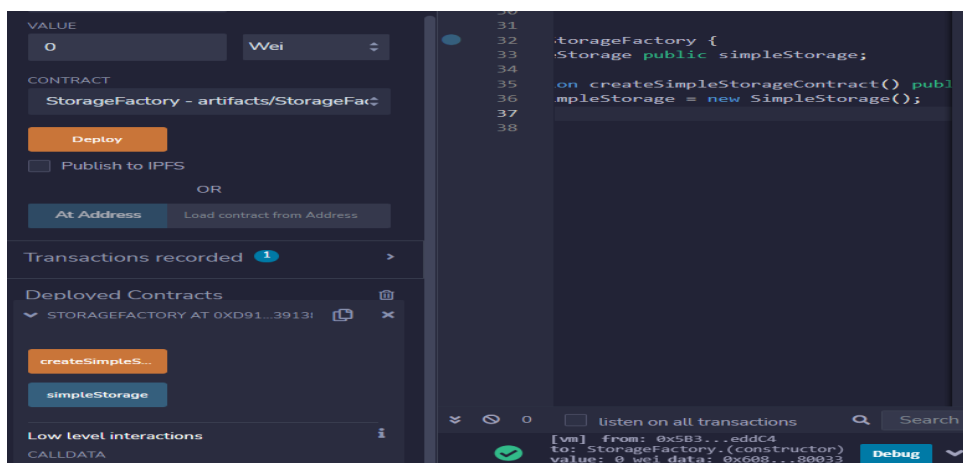
Kemudian klik Compile Storage



Setelah selesai compile, ubah kontrak yang awalnya simpleStorage menjadi StorageFactory lalu DEPLOY



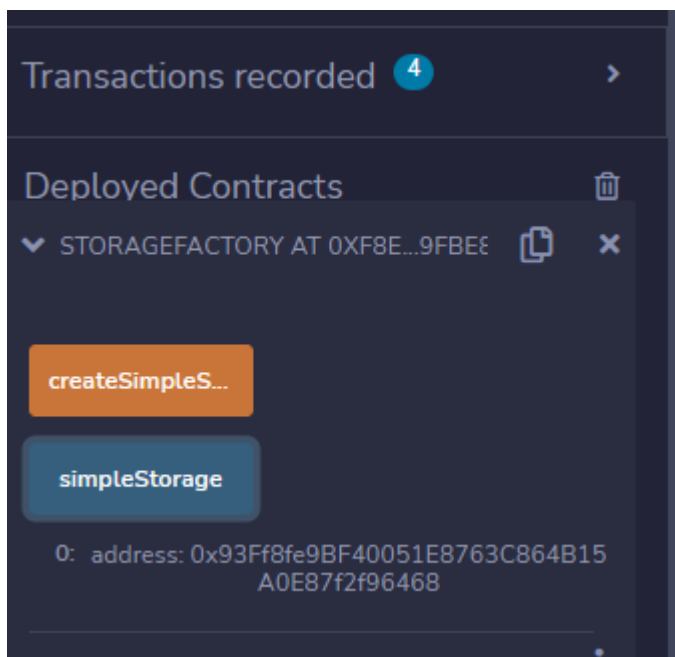
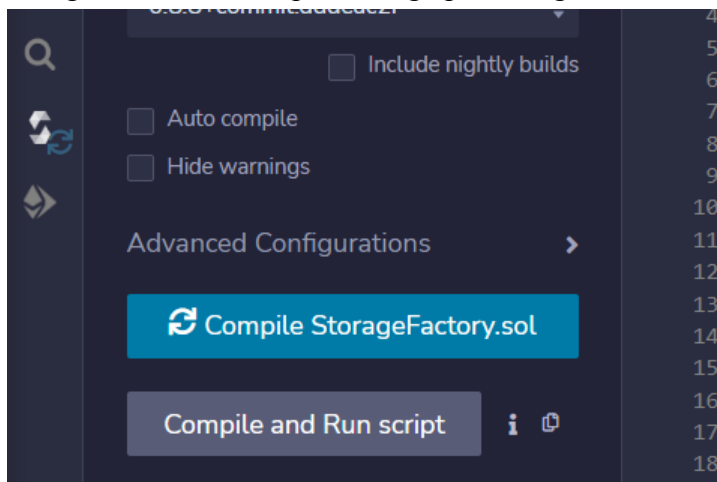
Setelah Deploy, lakukan disini kita memiliki 2 pilihan storage factory contract, yang pertama untuk membuat storage dan yang kedua untuk melihat storage contract kita. Disini kita bisa memulai membuat kontrak maupun hanya melihat kontrak sesuai kemauan kita.



Kemudian langkah selanjutnya, tambahkan kodingan pada StorageFactory seperti berikut. Kodingan ini berfungsi sama seperti sistem copy paste pada SimpleStorage.



Kemudian lakukan Compile dan lakukan pengulangan Deploy, dan klik Simple storage dan CreateSimple Storage pada langkah sebelumnya.



Kemudian lakukan tambahkan kodingan “StorageFactory.sol” seperti contoh di bawah ini, lalu lakukan Pengulangan Deploy dan Create.

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

import "./SimpleStorage.sol";

contract StorageFactory {
    SimpleStorage[] public simpleStorage;

    function createSimpleStorageContract() public {
        SimpleStorage simpleStorage = new
SimpleStorage();
        simpleStorageArray.push(simpleStorage);
    }
}
```

Interaksi dengan kontrak lain

Selanjutnya disini kita mempelajari cara interaksi dengan kontrak lain.

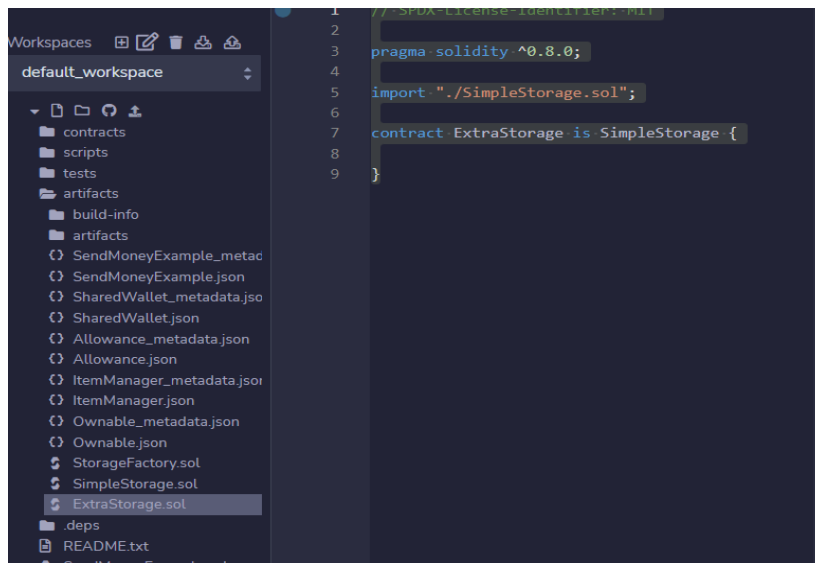
Pertama ketik kodingan berikut pada StorageFactory.sol kalian.

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

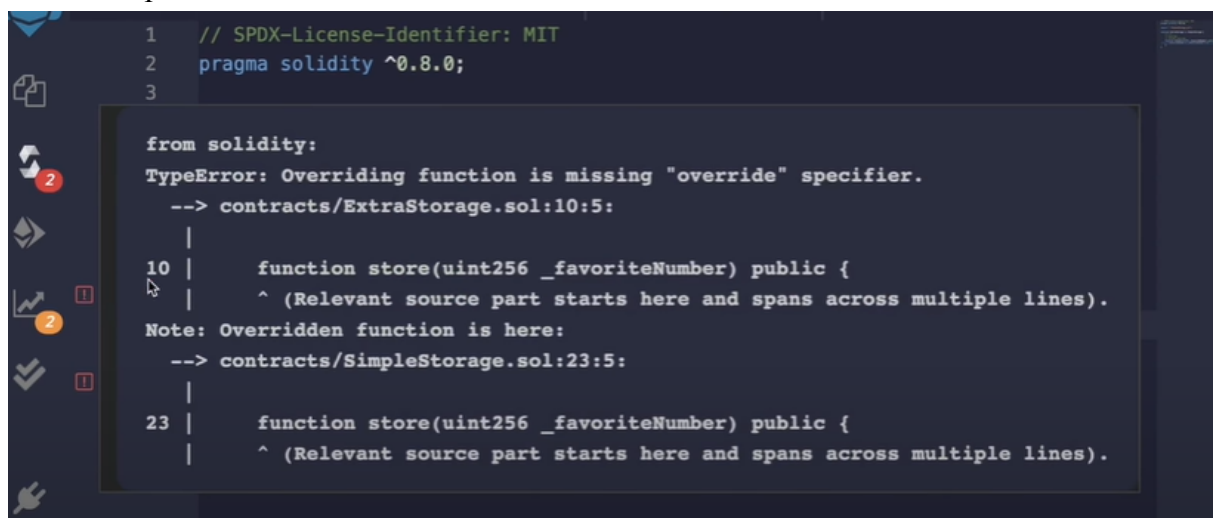
import "./SimpleStorage.sol";

contract StorageFactory {
    SimpleStorage[] public simpleStorage;
```

Kemudian buat kontrak baru dengan nama file ExtraStorage.sol dengan input kodingan dibawah ini, lalu lakukan compile dan deploy.



Ketika terjadi error seperti dibawah ini, untuk mengatasinya tambhakan function override seperti contoh dibawah ini.



```
// SPDX-License-Identifier: MIT

pragma solidity ^0.8.0;

import "./SimpleStorage.sol";

contract ExtraStorage is SimpleStorage {
    // + 5
    // override
    // virtual override
    function store(uint256_favoriteNummber
public override {
```



```

        _favoriteNumber = _favoriteNummber +5;
    }
}

```

4. Remix Fund Me

Sebelum memulai untuk penggabungan dana copy paste kodingan berikut beserta nama kontrak sesuai contoh berikut. Masukkan kodingan kedalam FundMe.sol lalu lakukan compile dan Deploy.

 FundMe.sol
 PriceConverter.sol

```

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

import
"@chainlink/contracts/src/v0.8/interfaces/AggregatorV3Int
erface.sol";
import "../PriceConverter.sol";

error NotOwner();

contract FundMe {
    using PriceConverter for uint256;

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

    // Could we make this constant? /* hint: no! We
should make it immutable! */
    address public /* immutable */ i_owner;
    uint256 public constant MINIMUM_USD = 50 * 10 ** 18;

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

```

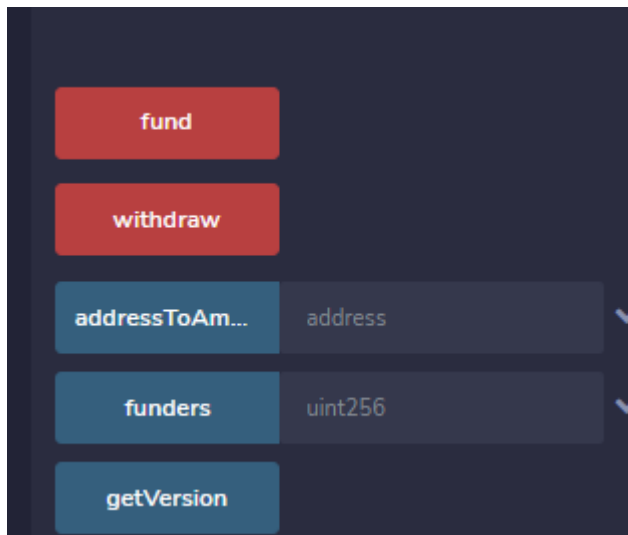
```

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

    function getVersion() public view returns (uint256){
        AggregatorV3Interface priceFeed =
AggregatorV3Interface(0x8A753747A1Fa494EC906cE90E9f37563A
8AF630e);
        return priceFeed.version();
    }

```

Kemudian, setelah menDeploy lanjut untuk “fund” pada menu Deploy. Ketika seseorang memencet fund maka sebenarnya dilakukan penarikan dana kembali dan mereset menjadi 0. Hal ini berguna untuk anda mengosongkan metamask anda dan mengubah kontrak baru.



5. Ethers.js Simple Storage

Sebelum memulai, buka kontrak FundMe.sol lalu hapus kodingan Public Payable menjadi Public

```
21     }
22
23     function fund() public {
24         require(msg.value.getConversionRate() >= MINIMUM
25         // require(PriceConverter.getConversionRate(msg.
26         addressToAmountFunded[msg.sender] += msg.value;
27         funders.push(msg.sender);
28     }
29
30     function getVersion() public view returns (uint256){
31         AggregatorV3Interface priceFeed = AggregatorV3In
32         return priceFeed.version();
```

Kemudian akan muncul error seperti berikut. Disini jika kamu kesusahan dalam mencari kesalahan coding, kamu dapat mencari bantuan dari forum gitHub ataupun mencari repository milik orang lain.

COMPILER

TypeError:
"msg.value" and
"callvalue()"
can only be used
in payable
public
functions. Make
the function
"payable" or use
an internal

```
23     address public immutable i_owner
24     // 21,508 gas - immutable
25     // 23,644 gas - non-immutable
26
27     constructor(){
28         i_owner = msg.sender;
29     }
30
31     function fund() public {
32         require(msg.value.getConversionRate() >= MINIMUM
33         funders.push(msg.sender);
34         addressToAmountFunded[msg.sender] += msg.value;
35     }
36
37     function withdraw() public onlyOwner
```

Start a new discussion

Category: General General Thread for practicing question formatting!

Write Preview

H B I

```
modifier onlyOwner {
    // require(msg.sender == i_owner, NotOwner());
    if(msg.sender != i_owner) { revert NotOwner(); }
}

}
```

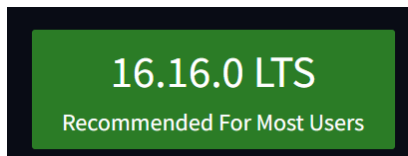
Hi

Attach files by dragging & dropping, selecting or pasting them.

Start discussion

Untuk membuat Hard Hat yang lebih baik maka kita akan membuat sistem Hardhat pada Visual Studio code, terdapat beberapa instalasi yang di perlukan untuk membuat Hard Hat pada Visual Studio Code diantaranya :

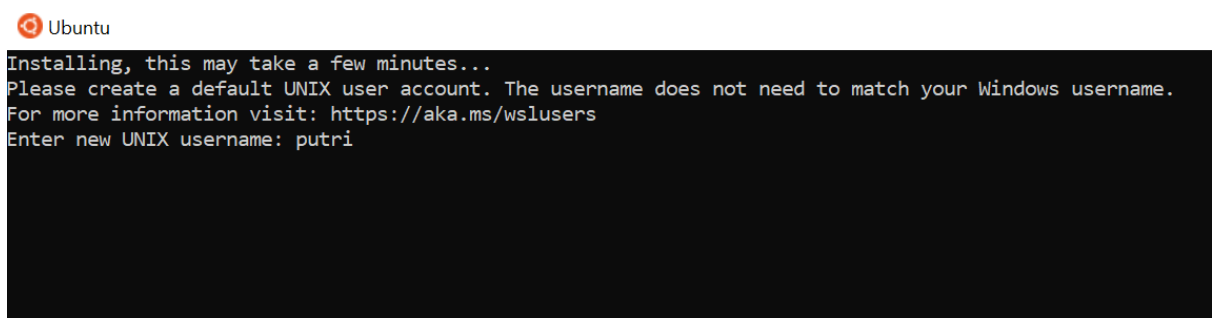
1. Install Node.JS pada link <https://nodejs.org/en/> dan pilihlah yang versi LTS.



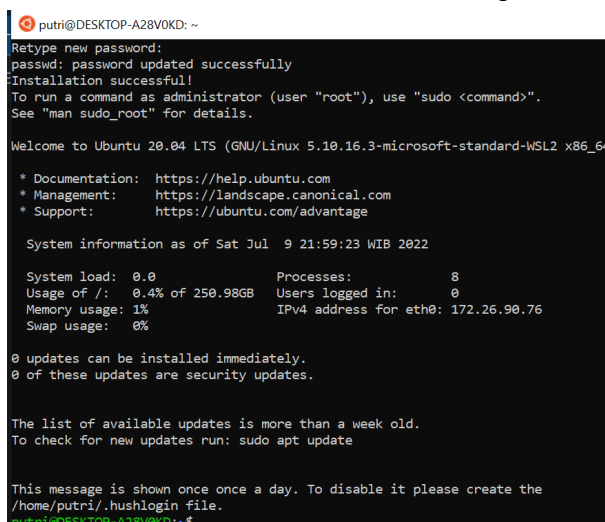
2. Setelah itu, download dan install Node.JS
3. Lalu, buka Visual Studio Code dan buka terminal lalu ketik `node --version` dan akan muncul versi berapa Node.JS yang digunakan.

```
PS C:\Users\Putri Mellia Zahrani> node --version
v16.16.0
```

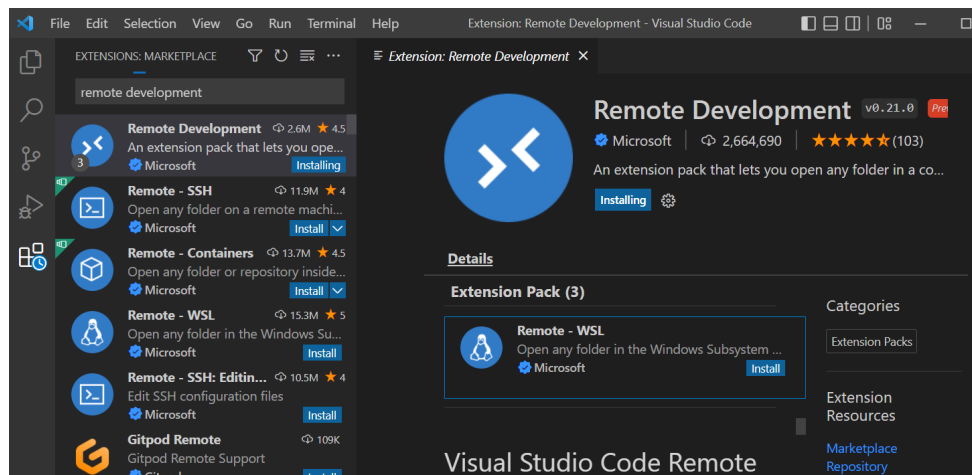
4. Setelah itu ketikan `wsl --install` untuk menginstall, wsl atau Windows Subsystem for Linux ini digunakan agar windows dapat berjalan dalam mode Linux. Jika terjadi error atau wsl tidak terinstall maka cari aplikasi VSCode dan klik kanan lalu run as administrator lalu buka terminal dan ketikan ulang `wsl --install`. Tunggu hingga wsl selesai terinstall lalu reboot komputer anda agar WSL dapat berjalan.
5. Setelah itu akan muncul pop up untuk mengatur username dan password pada Ubuntu.



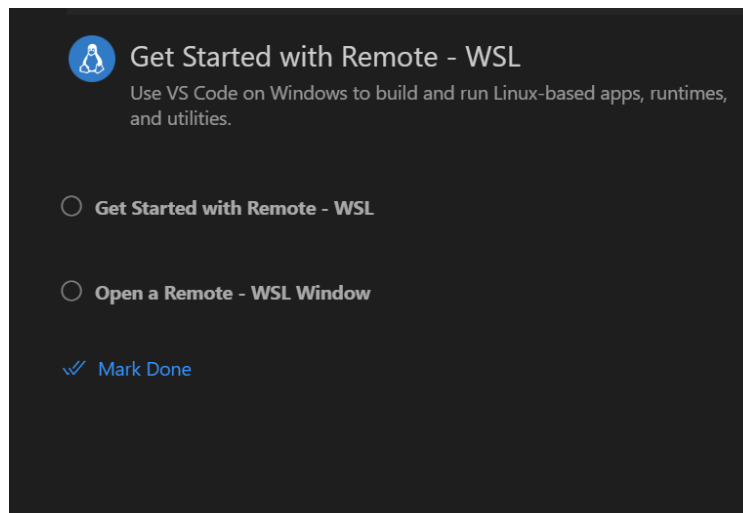
6. Setelah mengisi username dan password lalu enter dan tutup aplikasi Ubuntu, setelah itu buka kembali untuk memastikan dapat masuk ke dalam aplikasi kembali.



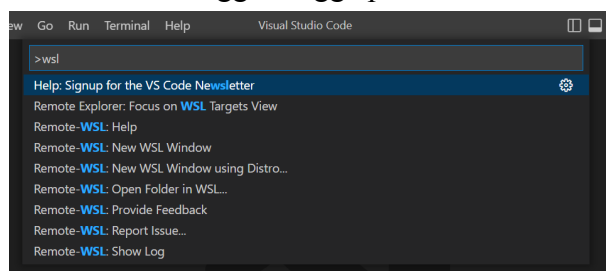
7. Install extension Remote Development pada Visual Studio Code.



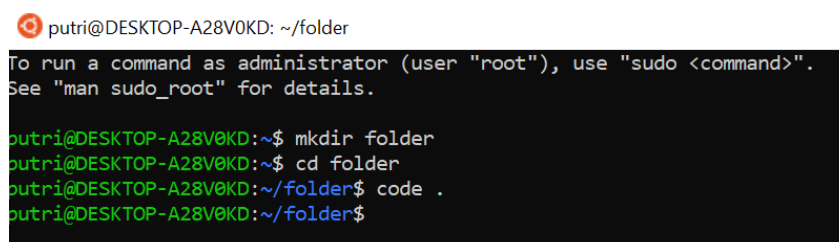
8. Lalu pilih mark done.



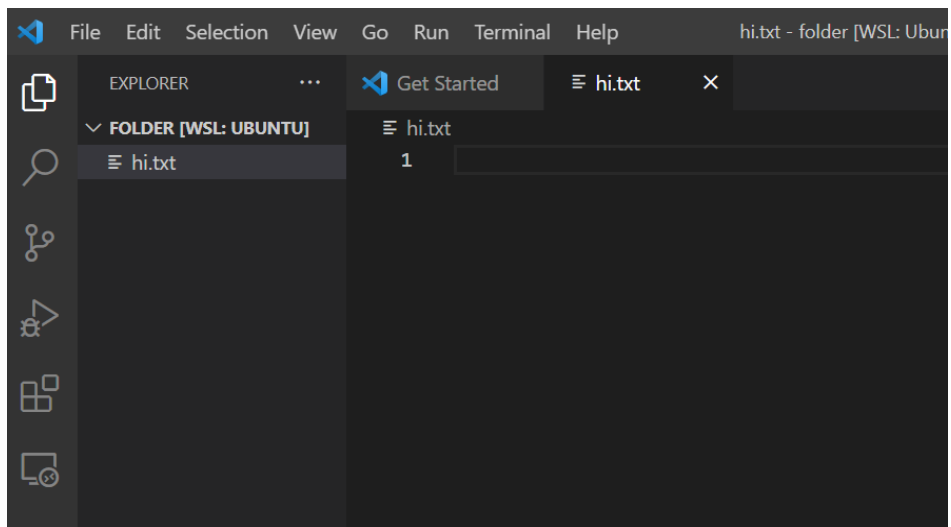
9. Setelah itu buka terminal lalu tekan Ctrl+Shift+P dan ketik wsl lalu pilih New WSL Window dan tunggu hingga proses selesai maka



10. Setelah proses selesai maka anda dapat menjalankan code menggunakan Linux dan bukan lagi menggunakan Windows Powershell.
11. Buka Ubuntu lalu ketikan seperti di bawah untuk membuat file pada VS Code



12. Maka tampilan VScode akan berubah menjadi



13. Selanjutnya kita akan melakukan instalasi NVM pada VScode.