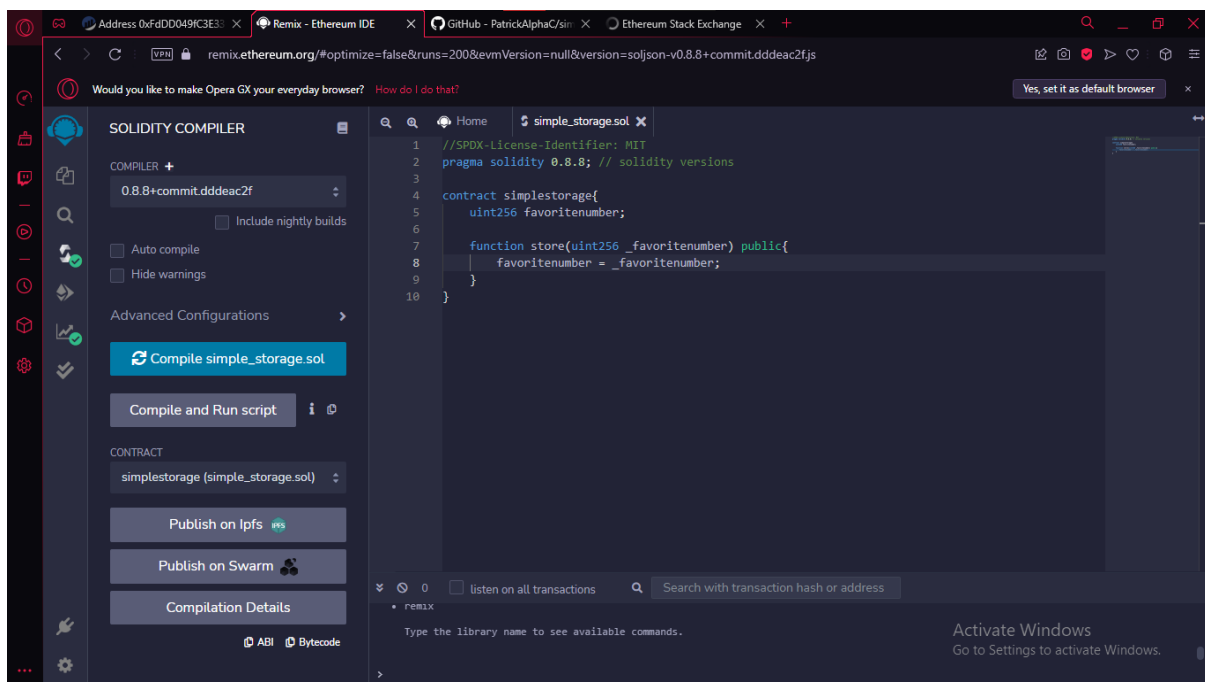


Welcome to Remix! Simple Storage :

Remix adalah sebuah IDE untuk membuat pengembangan smart contract pada jaringan ethereum. Langkah pertama pada remix adalah untuk membuat file solidity baru pada folder contract. Langkah selanjutnya adalah untuk menentukan compiler versi berapa yang digunakan, jika perlu tambahkan kredensial yang diperlukan (menggunakan MIT karena lebih bebas penggunaanya). Setelah menentukan versi compilernya maka kita akan membuat contract dengan perintah “contract *nama_contract* {”. Pada contract kita mengisi isi dari kontrak yang diperlukan, seperti fungsi, variabel, dan lain sebagainya.



Fungsi awal yang akan digunakan adalah store dan retrieve. Fungsi store digunakan untuk memasukan atau mengubah isi dari variable *favoritenumbr*. Fungsi retrieve menggunakan perintah view dan memberikan return *favoritenumbr* yang berguna untuk melihat isi variable. View digunakan agar hash atau data pada blockchain tidak berubah dan hanya membaca saja. Langkah selanjutnya adalah membuat tipe data baru dengan struct dan juga membuat arraynya

```
1 //SPDX-License-Identifier: MIT
2 pragma solidity 0.8.8; // solidity versions
3
4 contract simplestorage{
5     uint256 public favoritenumber;
6
7     struct People {
8         uint256 favoritenumber;
9         string name;
10    }
11    // uint256[] public anArray;
12    People[] public people;
13
14    function store(uint256 _favoritenumber) public{
15        favoritenumber = _favoritenumber;
16    }
17
18    function retrieve() public view returns (uint256){
```

Langkah selanjutnya adalah menambahkan fungsi addPerson. Fungsi ini berguna untuk menambah data pada array yang sudah dibuat sebelumnya. Penambahan data ini menggunakan perintah push.

```
function addPerson(string memory _name, uint256 _favoritenumber) public {
    people.push(People(_favoritenumber, _name));
}
```

Setelah itu buatlah mapping dan tambah perintah baru pada fungsi addPerson. Mapping berguna untuk memetakan jalan dari nama ke *favoritenumber*.

```
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;
}
```

Untuk melakukan deploy pada jaringan test maka ubah environment dari VM menjadi injected web3 untuk mengkoneksikan ke metamask kita.

remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.8+commit.d18e900f

MetaMask Notification

Rinkeby Test Network

Account 1 0xE17...AD43

New address detected! Click here to add to your address book.

DETAILS DATA HEX

Site suggested >

Gas (estimated) 0.00028096

Very likely in < 15 seconds

Max fee: 0.00028096 RinkebyETH

Total 0.00028096

0.00028096 RinkebyETH

Amount + gas fee Max amount: 0.00028096 RinkebyETH

ContractDefinition simplestorage 1 reference(s)

transact to simplestorage.addPerson pending ...

Activate Windows
Go to Settings to activate Windows.

DEPLOY & RUN TRANSACTIONS

Publish to IPFS

OR

At Address Load contract from Address

Transactions recorded 2

Deployed Contracts

SIMPLESTORAGE AT 0xE17...AD43

addPerson Ivan.BBB

store uint256_favoritenumber

nameToFavorite... string

people uint256

retrieve

Low level interactions

CALLDATA

Transact

Download audio from this page

```
1 //SPDX-License-Identifier: MIT
2 pragma solidity 0.8.8; // solidity version
3
4 contract simplestorage{
5     uint256 favoritenumber;
6
7
8
9
10
11     struct People {
12         uint256 favoritenumber;
13         string name;
14     }
15
16     // uint256[] public anArray;
17     People[] public people;
18
19     mapping(string => uint256) public nameToFavoriteNumber;
20
21     function store(uint256 _favoritenumber, string _name) public {
22         favoritenumber = _favoritenumber;
23         nameToFavoriteNumber[_name] = _favoritenumber;
24     }
25
26     function retrieve() public view returns (uint256) {
27         return favoritenumber;
28     }
29
30     function addPerson(string memory _name, uint256 _favoritenumber) public {
31         nameToFavoriteNumber[_name] = _favoritenumber;
32         people.push(People(_favoritenumber, _name));
33     }
34 }
```