



Lesson 5

REMIX STORAGE FACTORY

Hikmah Nisya - 1103184094
Radzis Araaf Jaya Jamaludin - 1103184234
Raudhatul Rafiqah Assyahiddini - 1103180225

Preparing Tools

1. Github: digunakan untuk kepentingan menyimpan code dan dokumentasi kode yang telah kita buat.

2. VS Code: akan digunakan selama lesson 5 dan kedepannya.

3. Solidity: bahasa yang akan digunakan selama kita belajar blockchain.

4. JavaScript: bahasa yang digunakan selama kita belajar blockchain.

5. Node.js: menggunakan node.js untuk melakukan compiler.

6. YARN: menggunakan yarn untuk menginstall beberapa module.

7. Ganache: digunakan untuk membuat local test env.

8. GIT: digunakan untuk upload dan download code.

Setting ENV pada VS Code

Untuk pengguna windows disarankan untuk menggunakan virtual environment. Kita dapat menggunakan fitur WSL pada windows 11, dengan cara melakukan install melalui powershell. Jika sudah berhasil maka kita akan melihat OS Ubuntu pada windows kita, jika berhasil install, maka langkah selanjutnya yaitu melakukan reebot pada pc/laptop kita.

Selanjutnya, disarankan melakukan update dan upgrade pada WSL kita menggunakan perintah “sudo apt-get update && apt-get upgrade”, kemudian tunggu hingga selesai.

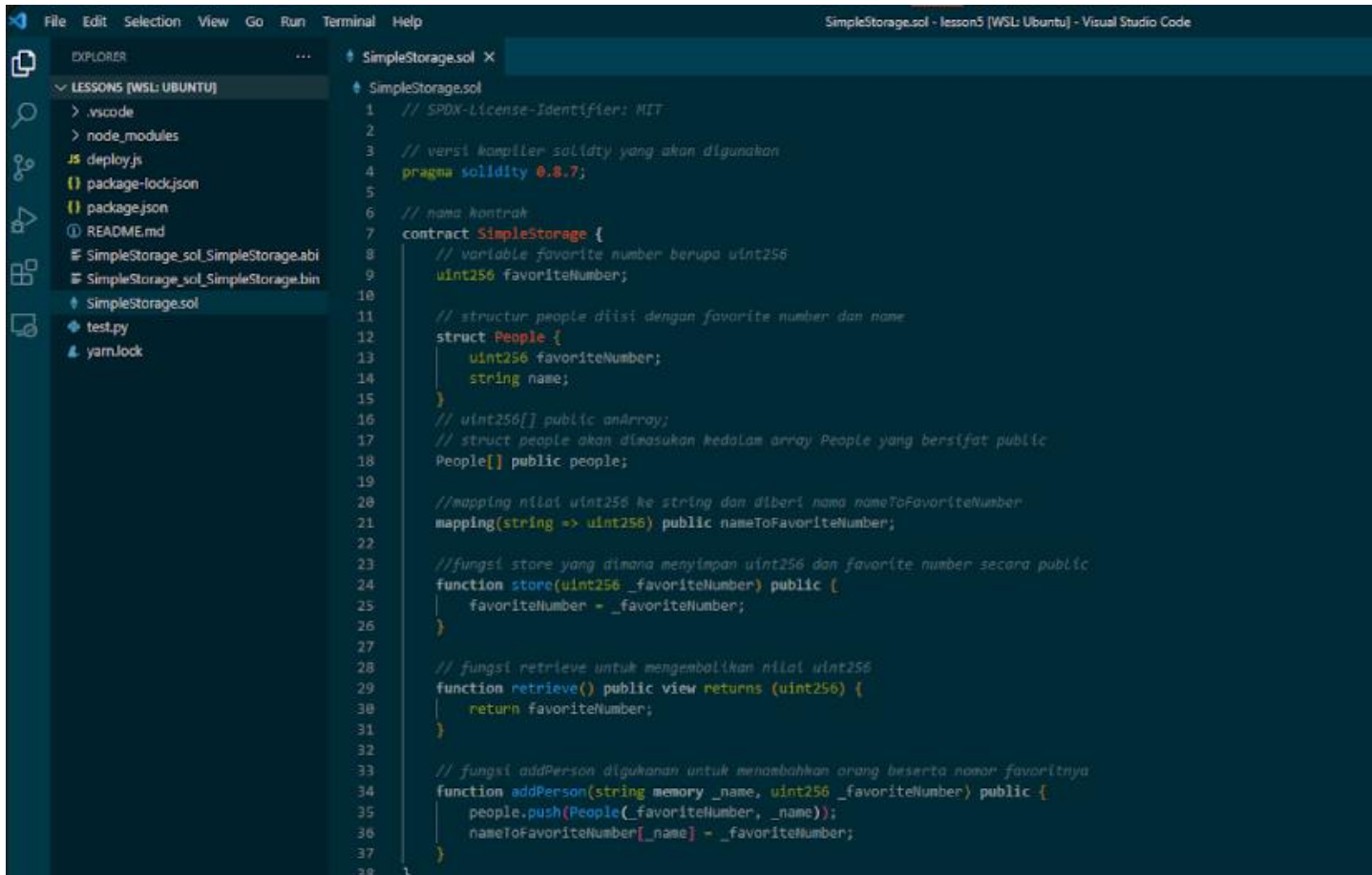
jika sudah melakukan pembaharuan, Langkah selanjutnya adalah untuk menginstall node.js dan yarn dengan perintah “sudo apt-get install nodejs”

Membuat file yang di antara isinya adalah sebagai berikut:

- SimpleStorage.sol
- Deploy.js
- Yarn.lock
- Package.lock

Semua file akan digunakan nantinya.

Membuat file pada WSL



```
1 // SPDX-License-Identifier: MIT
2
3 // versi kompilasi solidity yang akan digunakan
4 pragma solidity 0.8.7;
5
6 // nama kontrak
7 contract SimpleStorage {
8     // variable favorite number berupa uint256
9     uint256 favoriteNumber;
10
11     // struktur people diisi dengan favorite number dan name
12     struct People {
13         uint256 favoriteNumber;
14         string name;
15     }
16
17     // uint256[] public anArray;
18     // struct people akan dimasukkan ke dalam array People yang bersifat public
19     People[] public people;
20
21     //mapping nilai uint256 ke string dan diberi nama nameToFavoriteNumber
22     mapping(string => uint256) public nameToFavoriteNumber;
23
24     //fungsi store yang dimana menyimpan uint256 dan favorite number secara public
25     function store(uint256 _favoriteNumber) public {
26         favoriteNumber = _favoriteNumber;
27     }
28
29     // fungsi retrieve untuk mengembalikan nilai uint256
30     function retrieve() public view returns (uint256) {
31         return favoriteNumber;
32     }
33
34     // fungsi addPerson digunakan untuk menambahkan orang beserta nomor favoritnya
35     function addPerson(string memory _name, uint256 _favoriteNumber) public {
36         people.push(People(_favoriteNumber, _name));
37         nameToFavoriteNumber[_name] = _favoriteNumber;
38     }
39 }
```

Membuat Code SimpleStorage

```
1 // SPDX-License-Identifier: MIT
2
3 // versi kompilier solidity yang akan digunakan
4 pragma solidity 0.8.7;
5
6 // nama kontrak
7 contract SimpleStorage {
8     // variable favorite number berupa uint256
9     uint256 favoriteNumber;
10
11     // struktur people diisi dengan favorite number dan name
12     struct People {
13         uint256 favoriteNumber;
14         string name;
15     }
16     // uint256[] public anArray;
17     // struct people akan dimasukan kedalam array People yang bersifat public
18     People[] public people;
19
20     //mapping nilai uint256 ke string dan diberi nama nameToFavoriteNumber
21     mapping(string => uint256) public nameToFavoriteNumber;
22
23     //fungsi store yang dimana menyimpan uint256 dan favorite number secara public
24     function store(uint256 _favoriteNumber) public {
25         favoriteNumber = _favoriteNumber;
26     }
27
28     // fungsi retrieve untuk mengembalikan nilai uint256
29     function retrieve() public view returns (uint256) {
30         return favoriteNumber;
31     }
32
33     // fungsi addPerson digunakan untuk menambahkan orang beserta nomor favoritnya
34     function addPerson(string memory _name, uint256 _favoriteNumber) public {
35         people.push(People(_favoriteNumber, _name));
36         nameToFavoriteNumber[_name] = _favoriteNumber;
37     }
38 }
```

Membuat code Deploy

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER > node + v [] [X]

chainId: 1337,
v: 2709,
r: '0xbe35fd5349fdc540fac4418ed501a3d63ddff241a971f13b88374327380a439d',
s: '0x1e5a067c1362e75b68f0aca279cf92885a52595093b4fc6f2b0d990acd727d3b',
from: '0x17Fab9d0B204d46A820C6F15eE72e9ba8B588621',
hash: '0x36c94e8e64f4edf3893605319d1d3a0767d2e4a508f3aa54c9953fa5f775c7a0',
type: null,
confirmations: 0
},
transactionHash: '0x36c94e8e64f4edf3893605319d1d3a0767d2e4a508f3aa54c9953fa5f775c7a0'
}
```

Membuat Deploy SC
