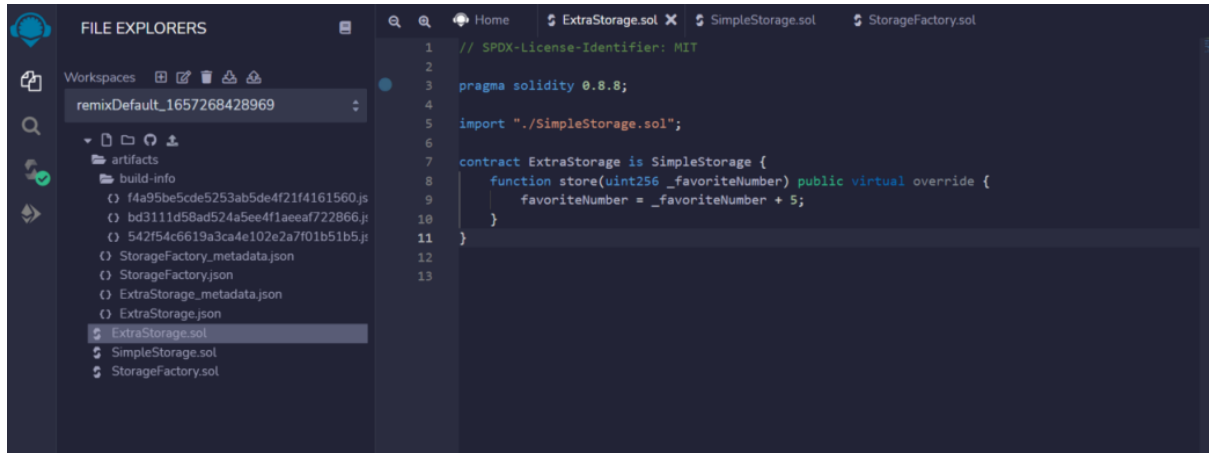


Nama : Raihan Puspa Anggita Putri

NIM : 1103184065

Kelas : TK-42-PIL

## Lesson 3 : Remix Storage Factory



The screenshot shows the Remix IDE interface. On the left, the 'FILE EXPLORERS' panel displays a workspace named 'remixDefault\_1657268428969'. Under the 'artifacts' folder, there is a 'build-info' directory containing several files. Below that, there are three JSON files: 'StorageFactory\_metadata.json', 'StorageFactory.json', and 'ExtraStorage\_metadata.json'. At the bottom of the file list, three Solidity files are shown: 'ExtraStorage.sol' (selected), 'SimpleStorage.sol', and 'StorageFactory.sol'. The main editor area displays the code for 'ExtraStorage.sol'. The code starts with a comment '// SPDX-License-Identifier: MIT', followed by 'pragma solidity 0.8.8;', and an 'import' statement for './SimpleStorage.sol'. The contract 'ExtraStorage' is defined as inheriting from 'SimpleStorage'. It contains a single function 'store' that takes a 'uint256 \_favoriteNumber' and increments the 'favoriteNumber' by 5.

```
1 // SPDX-License-Identifier: MIT
2
3 pragma solidity 0.8.8;
4
5 import "./SimpleStorage.sol";
6
7 contract ExtraStorage is SimpleStorage {
8     function store(uint256 _favoriteNumber) public virtual override {
9         favoriteNumber = _favoriteNumber + 5;
10    }
11 }
12
13
```



The screenshot shows the Remix IDE interface with the 'StorageFactory.sol' file selected in the main editor. The 'FILE EXPLORERS' panel on the left shows the same workspace and file structure as the previous screenshot, but 'StorageFactory.sol' is now selected. The code for 'StorageFactory' starts with a comment '// SPDX-License-Identifier: MIT', followed by 'pragma solidity ^0.8.0;', and an 'import' statement for './SimpleStorage.sol'. The contract 'StorageFactory' contains two functions. The first, 'createSimpleStorageContract', is a public function that creates a new 'SimpleStorage' instance, pushes its address into a public array 'simpleStorageArray', and returns the address. The second, 'sfStore', is a public function that takes a '\_simpleStorageIndex' and a '\_simpleStorageNumber', and calls the 'store' function on the 'SimpleStorage' instance at that index.

```
1 // SPDX-License-Identifier: MIT
2
3 pragma solidity ^0.8.0;
4
5 import "./SimpleStorage.sol";
6
7 contract StorageFactory {
8
9     SimpleStorage[] public simpleStorageArray;
10
11     function createSimpleStorageContract() public {
12         SimpleStorage simpleStorage = new SimpleStorage();
13         simpleStorageArray.push(simpleStorage);
14     }
15
16     function sfStore(uint256 _simpleStorageIndex, uint256 _simpleStorageNumber) public {
17         // Address
18         // ABI
19         // SimpleStorage(address(simpleStorageArray[_simpleStorageIndex])).store(_simpleStorageNumber);
20         simpleStorageArray[_simpleStorageIndex].store(_simpleStorageNumber);
21     }
22 }
23
```