

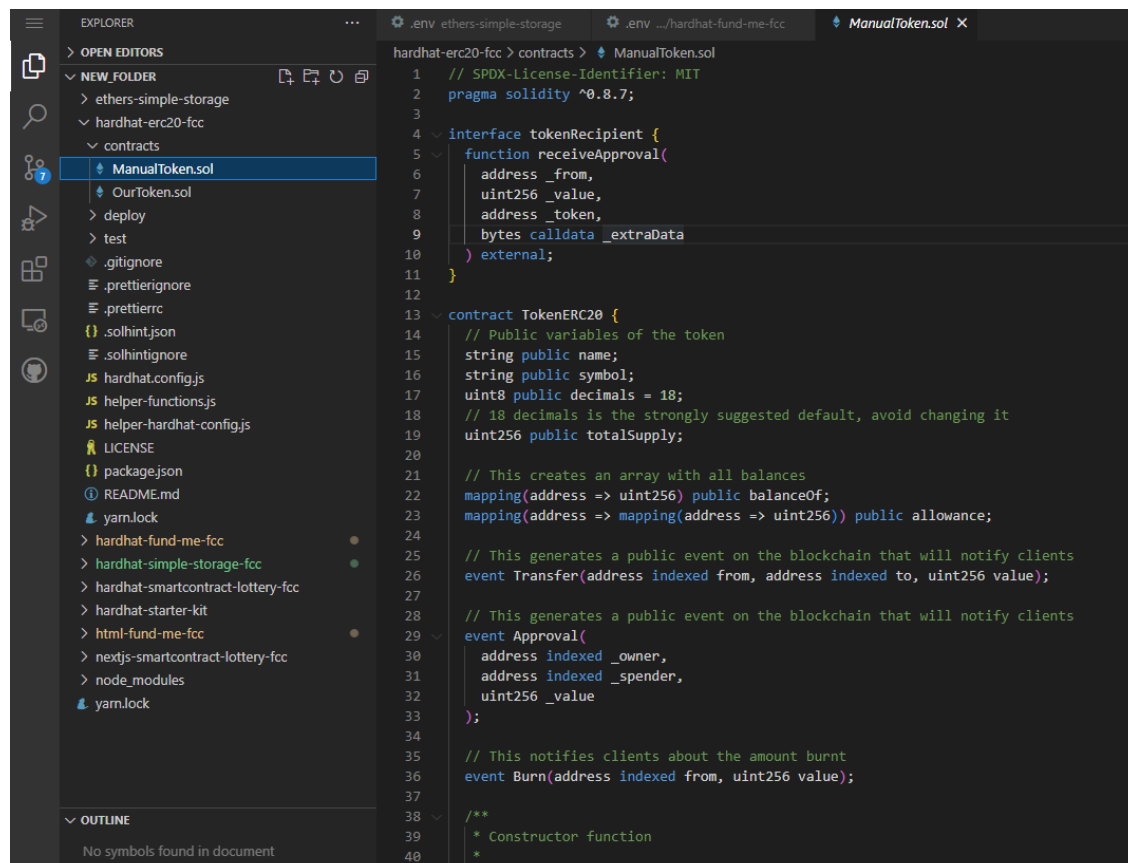
Hardhat ERC20s :

ERC20 Token merupakan standar token yang sebenarnya adalah smart contract dengan fungsi fungsi yang merepresentasikan seberapa token yang dimiliki oleh sebuah alamat. Namun token ini berbeda dengan token layer pertama seperti Ethereum, Polygon, dll yang disebut sebagai blockchain native token.

Ethereum Improvement Proposals	
All Core Networking Interface ERC Meta Informational	
EIP-20: Token Standard ↻	
Author	Fabian Vogelsteller, Vitalik Buterin
Status	Final
Type	Standards Track
Category	ERC
Created	2015-11-19

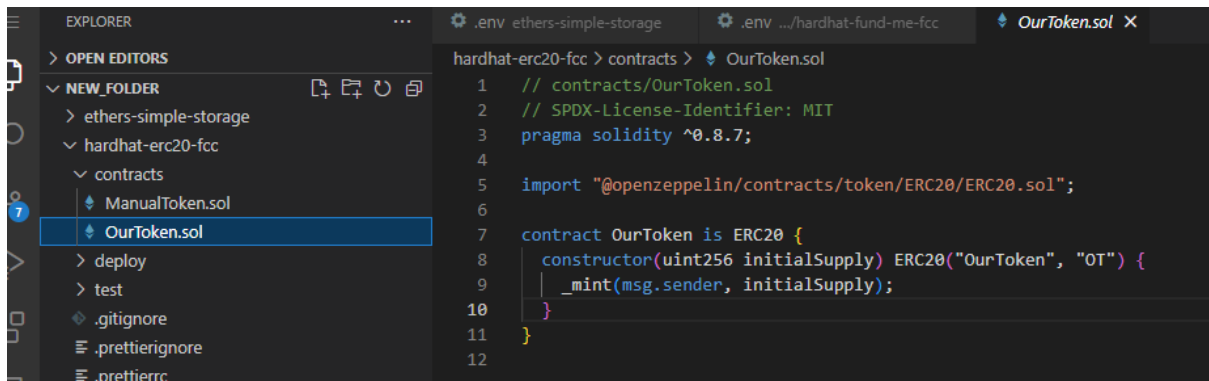
Table of Contents	
<ul style="list-style-type: none">• Simple Summary• Abstract• Motivation• Specification• Token<ul style="list-style-type: none">◦ Methods◦ Events• Implementation• History• Copyright	

Kita dapat membuat sebuah token dengan semua spesifikasi yang telah ditentukan



```
hardhat-erc20-fcc > 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, avoid changing it
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)) public allowance;
24
25     // This generates a public event on the blockchain that will notify clients
26     event Transfer(address indexed from, address indexed to, uint256 value);
27
28     // This generates a public event on the blockchain that will notify clients
29     event Approval(
30         address indexed _owner,
31         address indexed _spender,
32         uint256 _value
33     );
34
35     // This notifies clients about the amount burnt
36     event Burn(address indexed from, uint256 value);
37
38     /**
39      * Constructor function
40      */
41 }
```

Atau kita dapat menggunakan openzeppelin untuk mengimport sebuah token



```
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", "OT") {
9         _mint(msg.sender, initialSupply);
10    }
11 }
12
```