Course 5: Ethers.js Simple Storage

1. Installasi
   Sebelum kita mulai, ada beberapa hal yang ahrus kita install agar source code kita bisa berjalan dengan lancar, di antara itu adalah:
   -vsode: sebagai code editor
   -git: untuk mengupload kode online
   -node.js: untuk meng-compile javascript
   -yarn: untuk installasi beberapa module
   -ganache : untuk mendeploy fake etherium account secara lokal
2. SimpleStorage.sol
   Buatlah folder baru untuk course baru ini dan di dalam folder tersebut buatlah file "SimpleStorage.sol" dan copypaste kode berikut:

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

pragma solidity 0.8.7;

// 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;
    }
}
```

Ini adalah smart contract yang akan kita pakai untuk course simple storage menggunakan ethers.js.

Apabila ingin mengganti formatting solidity untuk lebih rapih setiap kali kita save, install "solidity + hardhat" dari tab Extension vscode, kemudian klik View>Command Palletes…>settings dan pilih settings(JSON) dan tambahkan kode berikut di baris paling bawah

```
"[solidity]": {
        "editor.defaultFormatter": "NomicFoundation.hardhat-solidity"
    }
```

Setelah itu balik lagi ke command palattes dan ketik settings dan pilih user settings, cari format to save dan centrang box tersebut.

3. Deploy.js

Buatlah file baru bernama "deploy.js" dan copypaste kode berikut:

```
async function main() {

}

main()
    .then(() => process.exit(0))
    .catch((error) => {
        console.error(error)
        process.exit(1)
    })
```

Ini adalah fondasi dari kode yang akan kita pakai untuk mendeploy SimpleStorage.sol kita, tetapi sebelum kita dapat deploy kode solidity kita, kita harus install terlebih dahulu compiler solidity yaitu solc dengan memasukkan perintah berikut kedalam terminal

yarn add solc@0.8.7-fixed

setelah instalasi selesai, aka nada file baru bernama "package.json", di dalam file tersebut tambahkan baris kode berikut dibawah "dependencies". Scripts ini akan meng-compile file solidity kita setiap kali kita mendeploy file js ini.

```
"scripts": {
    "compile": "yarn solcjs --bin --abi --include-path node_modules/ --
base-path . -o . SimpleStorage.sol"
  }
```

Setelah itu install lagi menggunakan yarn dua module yang akan kita pakai selanjutnya, yaitu:

yarn add ethers

yarn add fs-extra

Kemudian kita akan menambahkan baris kode berikut diatas file deploy.js kita

```
const ethers = require("ethers")
const fs = require("fs-extra")
```

Sekarang kita akan menambahkan kode di dalam fungsi main() kita, kodenya seperti ini:

```
let provider = new ethers.providers.JsonRpcProvider(process.env.RPC_URL)
let wallet = new ethers.Wallet(process.env.PRIVATE_KEY, provider)

const abi = fs.readFileSync("./SimpleStorage_sol_SimpleStorage.abi",
"utf8")
const binary = fs.readFileSync("./SimpleStorage_sol_SimpleStorage.bin",
"utf8")

const contractFactory = new ethers.ContractFactory(abi, binary, wallet)
console.log("Deploying, please wait...")
const contract = await contractFactory.deploy()

const deploymentReceipt = await contract.deployTransaction.wait(1)
console.log(`Contract deployed to ${contract.address}`)
```

keterangan:
provider: block yang kita deploy dari aplikasi genache
wallet: private key dari salah satu fake account dari aplikasi genache
abi: file abi yang kita dapat saat file sol tercompile
bin: file bin yang kita dapat saat file sol tercompile
contractFactory: menggunakan module ethers untuk mendeploy file sol kita
contract: menunggu hasil contractFactory
deploymentReciept: mengambil receipt dari transaksi

Sekarang kita akan menambahkan kode yang akan menampilkan hasil transaksi dengan lebih sederhana.

```
let currentFavoriteNumber = await contract.retrieve()

console.log(`Current Favorite Number: ${currentFavoriteNumber}`)
console.log("Updating favorite number...")
let transactionResponse = await contract.store(7)
let transactionReceipt = await transactionResponse.wait()
currentFavoriteNumber = await contract.retrieve()
console.log(`New Favorite Number: ${currentFavoriteNumber}`)
```

Seperti file sol kita sebelumnya, apabila kita ingin editing format yang raph untuk file js kita, kita dapat mnginstall extension “Prettier – code formatter”, dan seperti sebelumnya kita akan menambahkan baris kode dalam settings.json kita yaitu seperti ini:

```
"[javascript]": {
        "editor.defaultFormatter": "esbenp.prettier-vscode"
    }
```

4. .env

   Untuk mengisi beberapa info yang kurang seperti block genache dan private key kita agar aman dari publik, kita akan menggunakan module dotenv yang akan menyimpan data private kita. Pertama, kita akan menginstall dotenv dengan perintah pada terminal

   yarn add dotenv

   setelah itu kita akan membuat file bari bernama ".env", dalam file ni kita akan mengisi informasi private kita untuk di import ke dalam deploy.js kita. Isi file .env adalah sebagai berikut:

```
PRIVATE_KEY=/*isi dengan private key yang akan dipakai*/
RPC_URL=http://{ip address block yang akan dipakai}
PRIVATE_KEY_PASSWORD=/*password yang akan digunakan untuk enkripsi
di bagian selanjutnya*/
```

   Agar file .env kita tidak terupload oleh git apabila kita akan menpublish kode kita maka kita akan membuat file baru bernama ".gitignore" dimana isi dari file ini adalah list file atau folder yang kita tidak ingin untuk di tarik oleh git.

5. (optional) encryptKey.js

   Apabila kita menginginkan keamanan yang lebih kuat kita dapat meng-enkripsi private key kita dengan membuat file js baru bernama "encryptKey.js". isi ari file ini adalah

```
const ethers = require("ethers")
const fs = require("fs-extra")
require("dotenv").config()

async function main() {
    console.log(process.env.PRIVATE_KEY)
    console.log(process.env.PRIVATE_KEY_PASSWORD)
    const wallet = new ethers.Wallet(process.env.PRIVATE_KEY)
    const encryptedJsonKey = await wallet.encrypt(
        process.env.PRIVATE_KEY_PASSWORD,
        process.env.PRIVATE_KEY
    )
    console.log(encryptedJsonKey)
    fs.writeFileSync("./.encryptedKey.json", encryptedJsonKey)
}

main()
    .then(() => process.exit(0))
    .catch((error) => {
        console.error(error)
        process.exit(1)
    })
```

   Kita akan langsung mendeploy file ini dengan perintah terminal

   Deploy encryptKey.js

Setalah selesai berjalan kita akan mendapatkan file baru bernama “encryptedKey.json”. file ini berisi hasil enkripsi kita yang akan kita akan pakai mulai sekarang, dan apabila kita akan memakai private key baru, kita tinggal mendeploy file encryptKey.js lagi.

Cara menggunakan file hasil enkripsi kita adalah dengan merubah isi file deploy.js kita, hasil edit yang menggunakan file enkripsi kita dalah sebagai berikut:

```
    let provider = new
ethers.providers.JsonRpcProvider(process.env.RPC_URL)
    // let wallet = new ethers.Wallet(process.env.PRIVATE_KEY, provider)
    const encryptedJson = fs.readFileSync("./.encryptedKey.json", "utf8");
    let wallet = new ethers.Wallet.fromEncryptedJsonSync(
       encryptedJson,
       process.env.PRIVATE_KEY_PASSWORD
     );
     wallet = wallet.connect(provider);
    const abi = fs.readFileSync("./SimpleStorage_sol_SimpleStorage.abi",
"utf8")
    const binary = fs.readFileSync(
        "./SimpleStorage_sol_SimpleStorage.bin",
        "utf8"
    )
```

6. (optional) .prettierrc

Bagian ini adalah bagian opsional apabila kita ingin membuat kode kita terlihat lebih rapi dengan cara yang mudah. Kita akan menginstall module prettier dari terminal dengan perintah *yarn add prettier prettier-plungin-solidity*
setelah terinstall, maka kita akan membuat file baru bernama “.prettierrc” yang berisi setting style kode kita seperti berikut:

```
{
  "tabWidth": 4,
  "useTabs": false,
  "semi": false,
  "singleQuote": false
}
```

7. Alchemy testnet

-Registrasi ke Alchemy dan buatlah etherium blockcahin ecosystem (contoh menggunakan network rinkeby)
-setelah registrasi selesai copy RCP HTTP URL dan paste kan ke dalam file .env menggantikan RCP_URL yang sebelumnya menggunakan genache
-untuk private key rinkeby kita dapat menggunakan fake private key apabila untuk percobaan saja atau menggunakan service seperti metamask yang menggunakan network rinkeby untuk menggantikan PRIVATE_KEY di dalam file .env.
-setelah semua yang diatas telah selesai, run deploy.js