UAS BLOCKCHAIN

TK-42-PIL

Raihan Puspa Anggita Putri (1103184065)
Sutan Faiz Rasyid (1103183160)
Meysa Rizkita (1103184182)

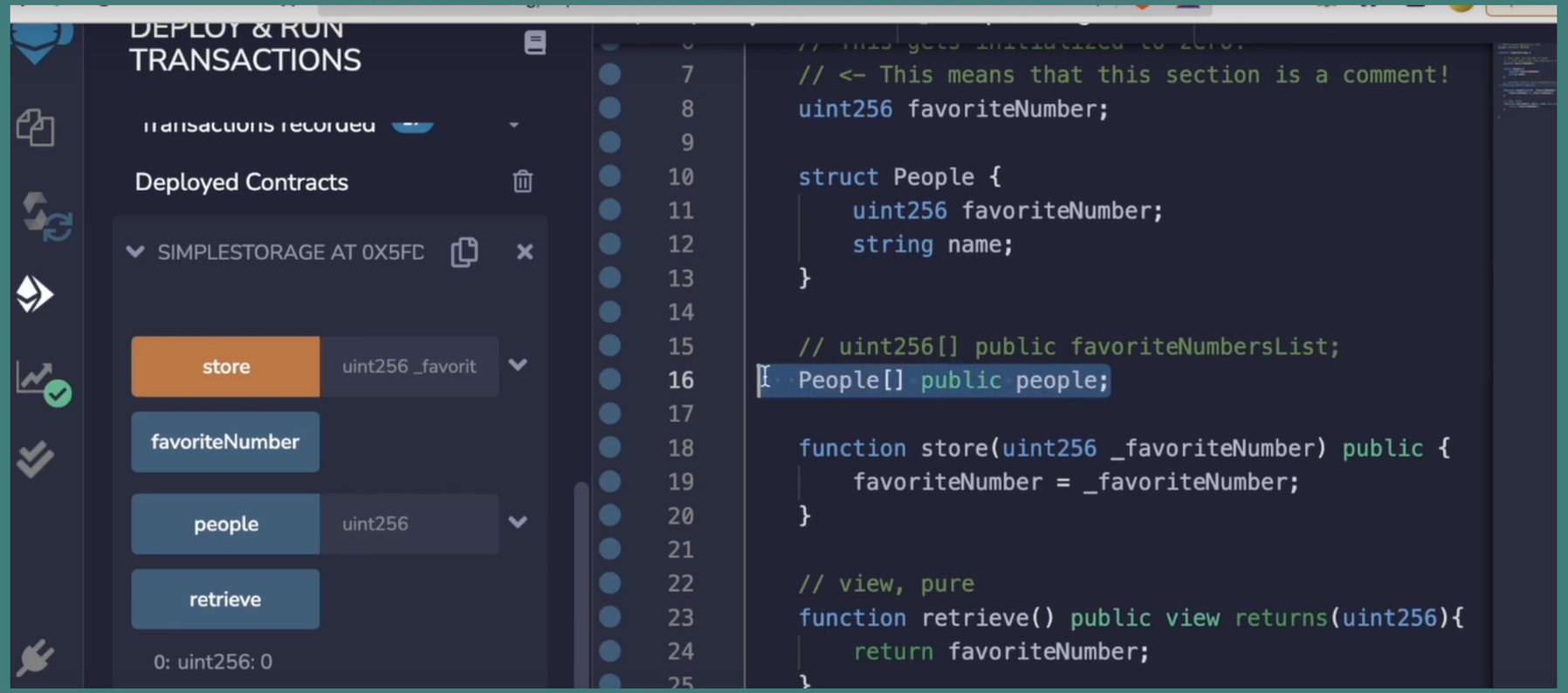
BLOCKCHAIN BASIC

Blockchain terdiri dari kata **Block** yang memiliki arti kelompok dan **chain** yang artinya rantai. Pengertian dari penamaan teknologi ini mencerminkan bagaimana cara kerja blockchain. Dimana, teknologi tersebut memanfaatkan resource komputer untuk menciptakan blok-blok yang terhubung (chain).

Blok-blok yang saling terhubung nantinya digunakan untuk mengeksekusi sebuah transaksi. Teknologi ini memang cukup menarik karena sifatnya yang tidak terpusat. Blockchain mampu berjalan sendiri menggunakan algoritma komputer tanpa ada sistem tertentu yang mengaturnya.

Remix IDE

Remix IDE memungkinkan pengembangan, penerapan, dan pengelolaan kontrak pintar untuk Ethereum seperti blockchain. Ini juga bisa digunakan sebagai platform pembelajaran.



Remix Storage Factory

In Remix, start by deploying contract B, then copy its address and give it to the constructor of A when deploying it. You can now call the **callHello()** function and you will get the result of the **sayHello()** function of contract B.

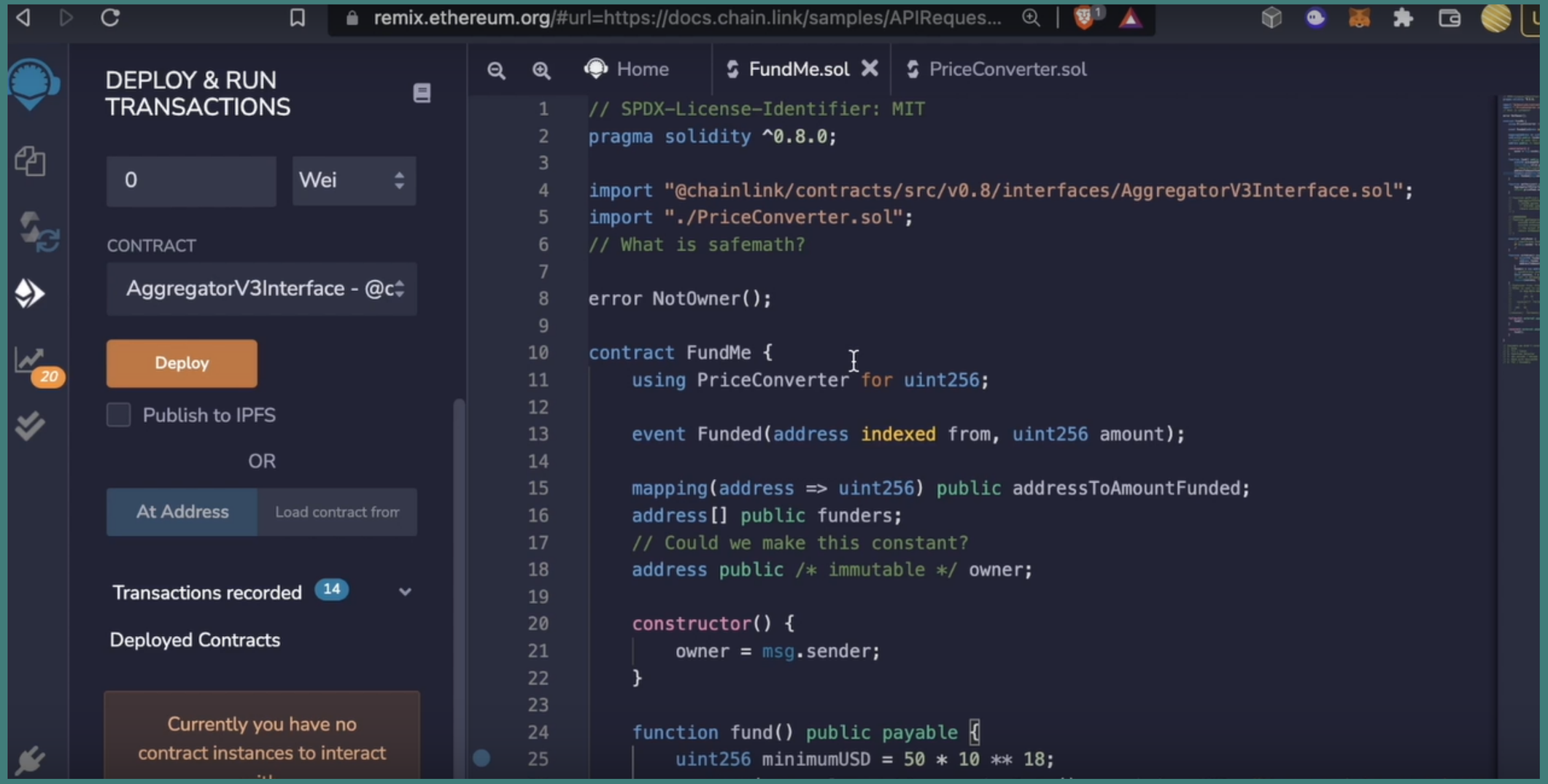
```
contract Factory{
      Child[] public children;
      uint disabledCount;

     event ChildCreated(address childAddress, uint data);

      function createChild(uint data) external{
        Child child = new Child(data, children.length);
        children.push(child);
        emit ChildCreated(address(child), data);
      }

      function getChildren() external view returns(Child[] memory _children){
        _children = new Child[](children.length- disabledCount);
        uint count;
        for(uint i=0;i<children.length; i++){
           if(children[i].isEnabled()){
              _children[count] = children[i];
              count++;
           }
         }
      }
```

Remix Fund Me



Ethers.js Simple Storage

Format solidity code

```
"[solidity]": {
    "editor.defaultFormatter": "NomicFoundation.hardhat-solidity"
},
"[javascript]":{
  "editor.defaultFormatter": "esbenp.prettier-vscode"
}
```

Hardhat Simple Storage

Hardhat dilengkapi dengan Hardhat Network, node jaringan Ethereum lokal yang dirancang untuk pengembangan, mirip dengan Ganache, geth --dev, dll. Ini memungkinkan Anda untuk menerapkan kontrak, menjalankan pengujian, dan men-debug kode yang ada

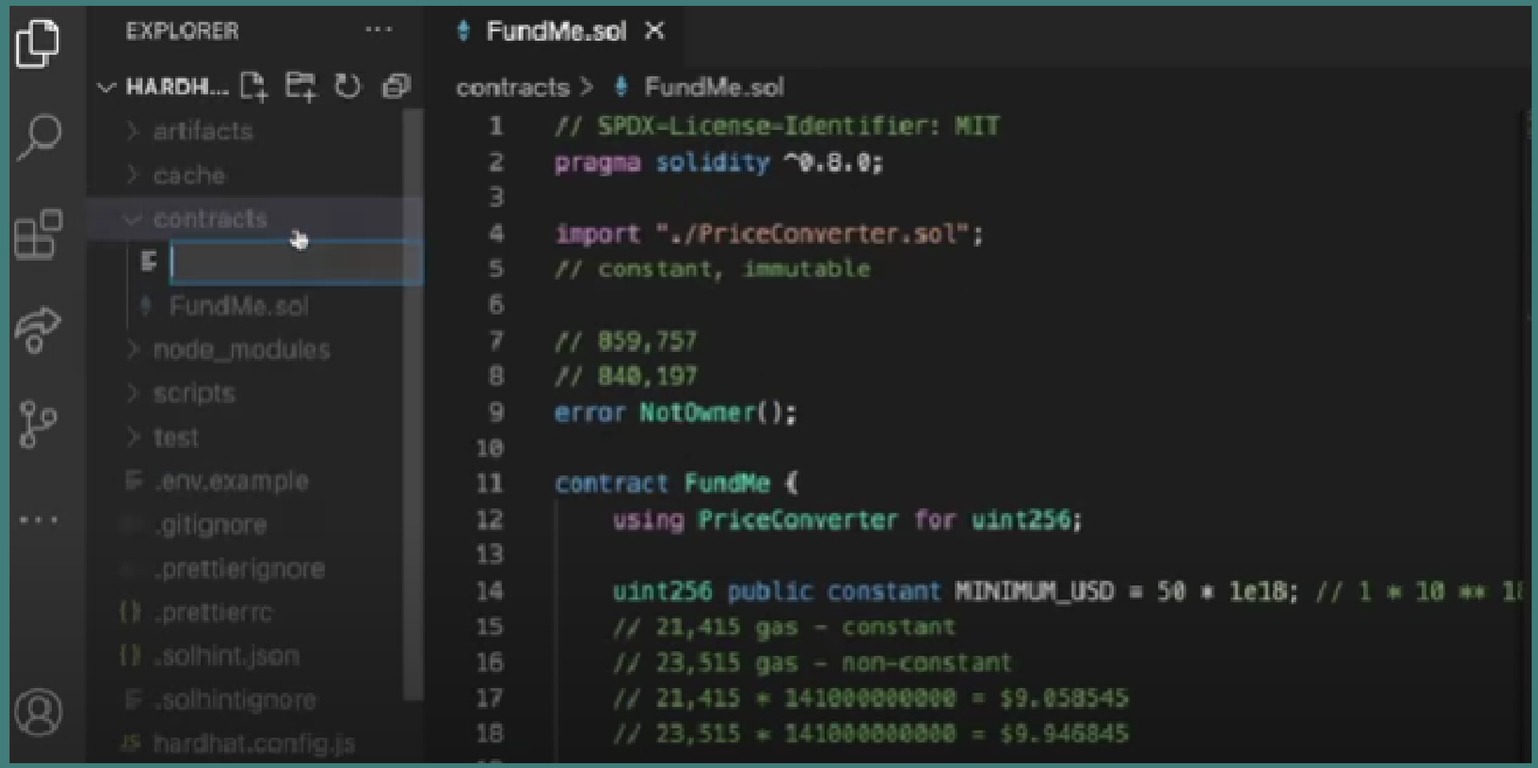
```
$ npx hardhat node
Started HTTP and WebSocket JSON-RPC server at http://127.0.0.1:8545/

Accounts
========
Account #0: 0xf39fd6e51aad88f6f4ce6ab8827279cfffb92266 (10000 ETH)
Private Key: 0xac0974bec39a17e36ba4a6b4d238ff944bacb478cbed5efcae784d7bf4f2ff8

Account #1: 0x70997970c51812dc3a010c7d01b50e0d17dc79c8 (10000 ETH)
Private Key: 0x59c6995e998f97a5a0044966f0945389dc9e86dae88c7a8412f4603b6b78690
...
```

Hardhat Fund Me

meningkatkan kontrak FundMe dengan alat yang lebih canggih seperti mocking dan pengoptimalan gas, dan mempelajari lebih lanjut tentang penyimpanan dan bytecode tingkat rendah di Solidity.

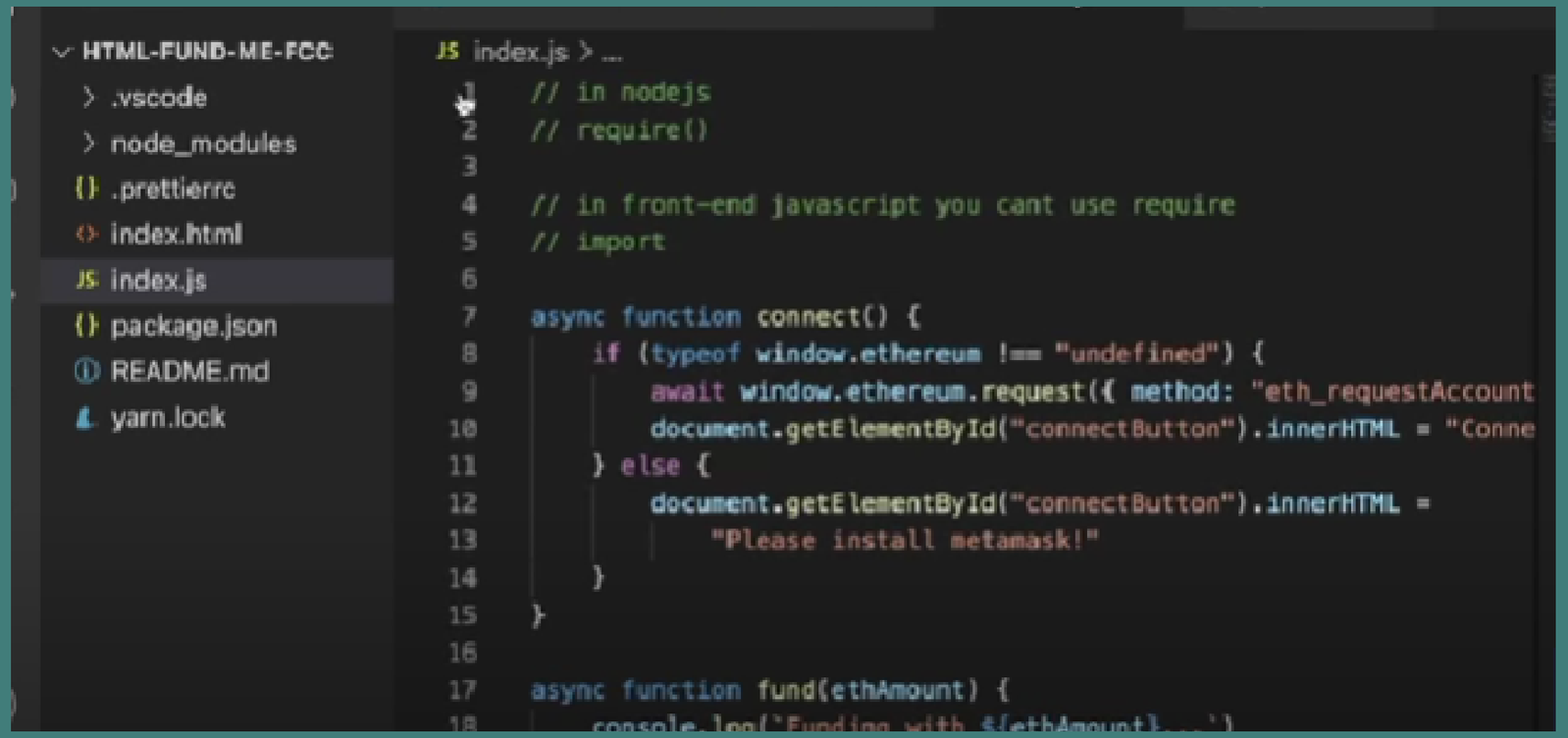


HTML / Javascript Fund Me (Full Stack / Front End)

Front End : pengembangan pada bagian situs web yang berfokus pada mengembangkan user interface

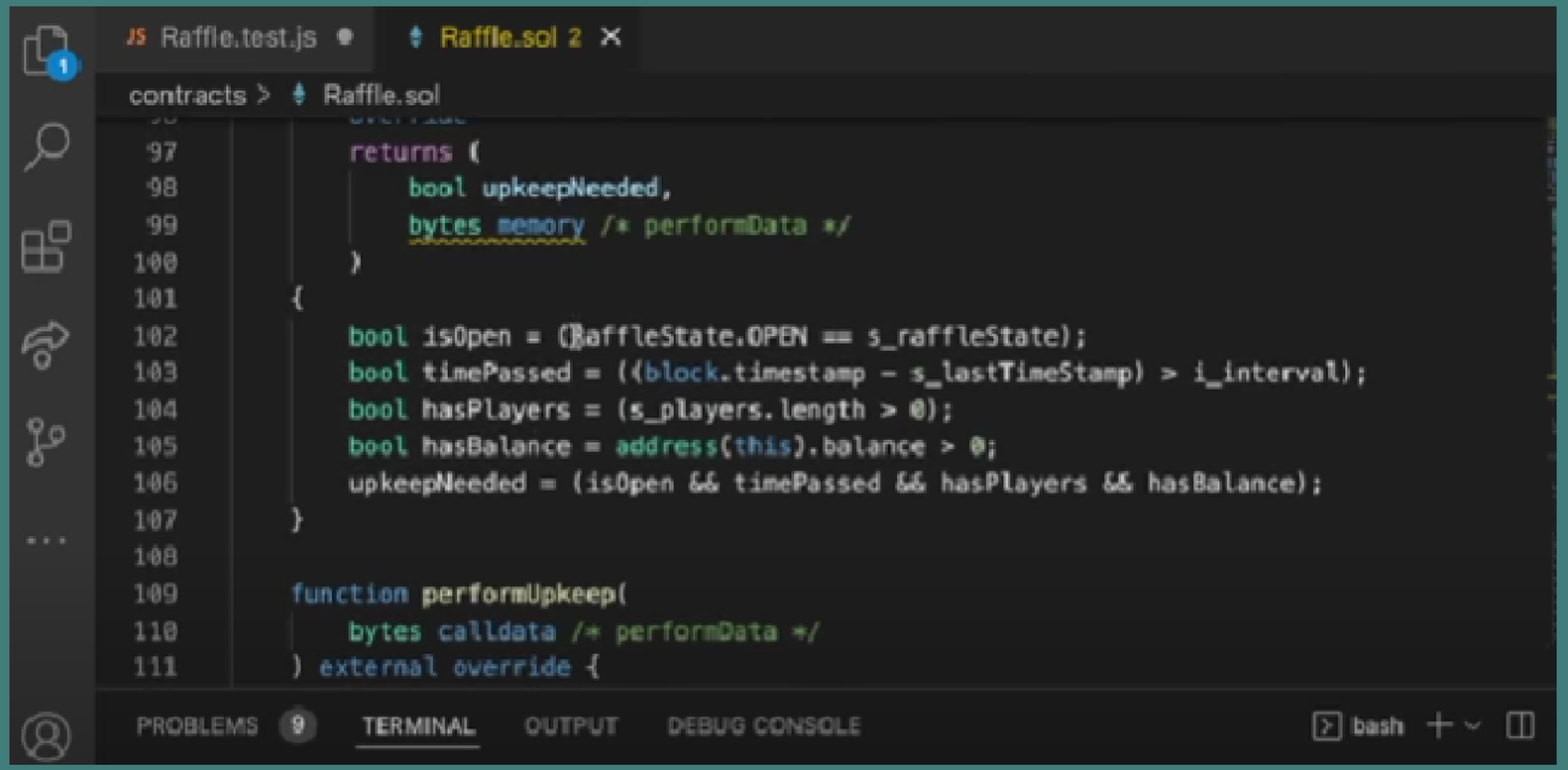
Full Stack : sebuah kombinasi front end dan back end.

situs web berinteraksi dengan dompet dan kontrak pintar untuk memahami dasar-dasar aplikasi Web3 full-stack.



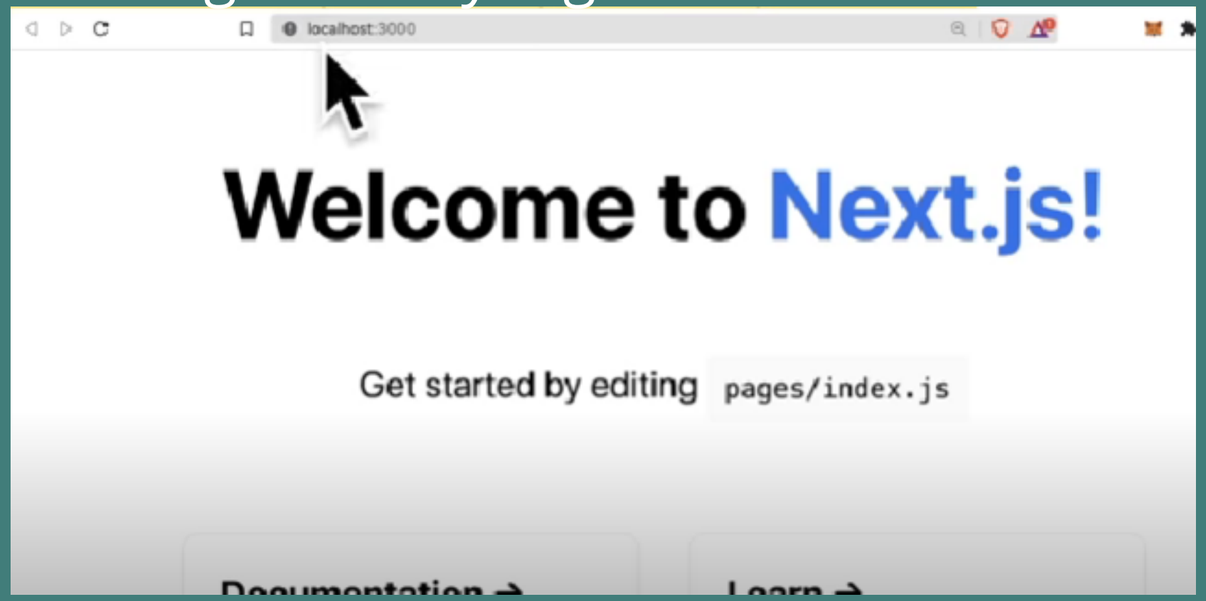
Hardhat Smart Contract Lottery

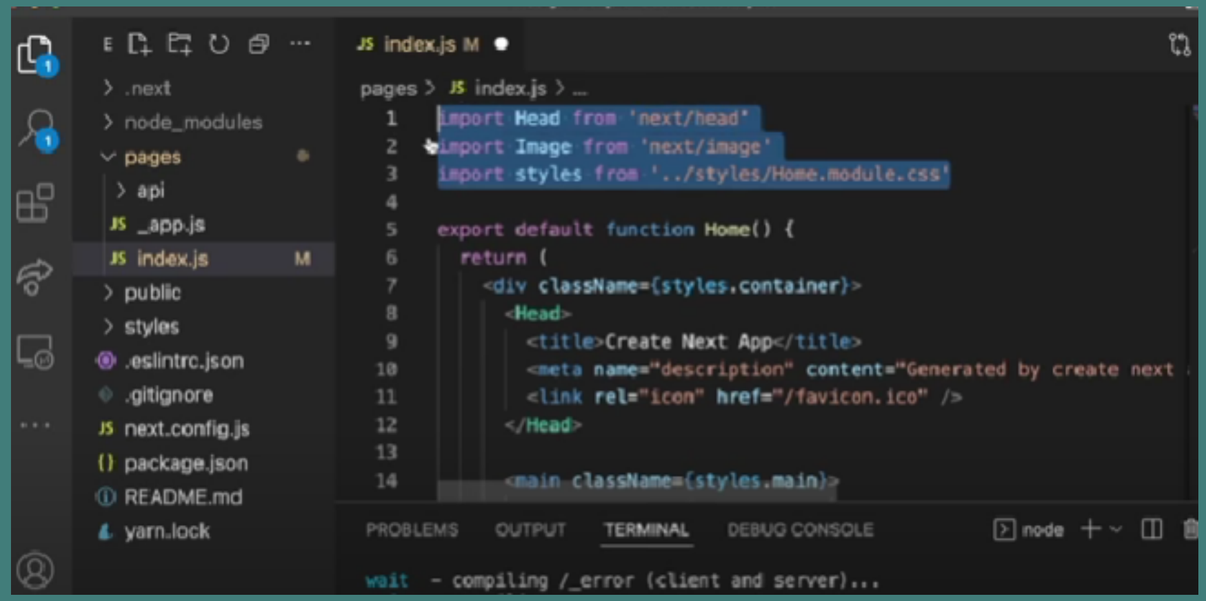
Pada Materi ini membuat lotre/undian yang adil secara kriptografis—sesuatu yang tidak mungkin ada di luar dunia blockchain. Untuk melakukan ini, belajar cara menggunakan Chainlink VRF dan Chainlink Keepers untuk menggabungkan keacakan yang dapat diverifikasi dengan otomatisasi terdesentralisasi.



NextJS Smart Contract Lottery (Full Stack / Front End)

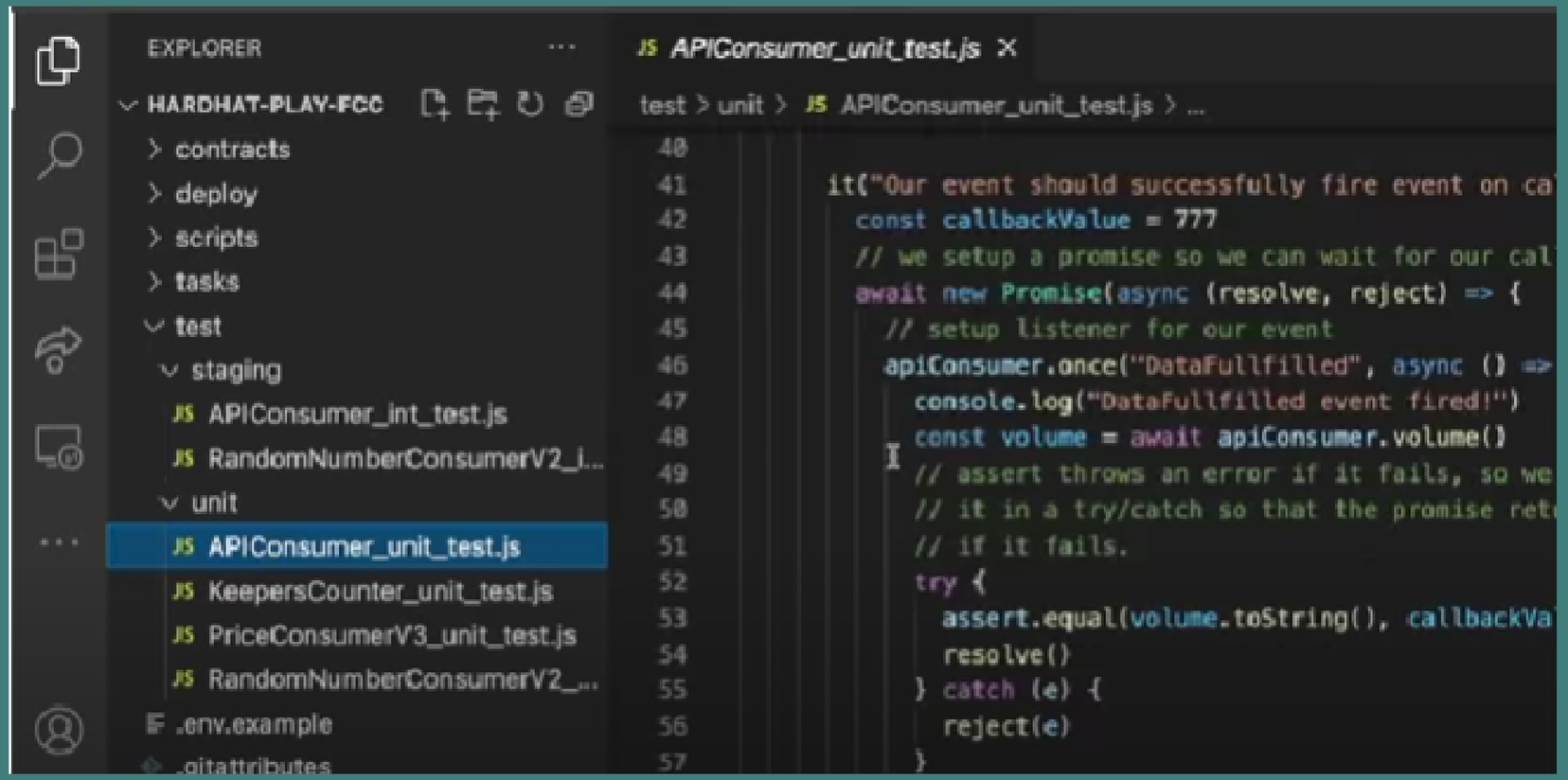
React adalah kerangka kerja frontend paling populer di planet ini sejauh ini, dan pelajaran ini mengajarkan cara-cara canggih di mana protokol miliaran dolar membangun situs web . disin menggunakan IPFS dan Fleek untuk meng-host situs web ini dengan cara yang terdesentralisasi.



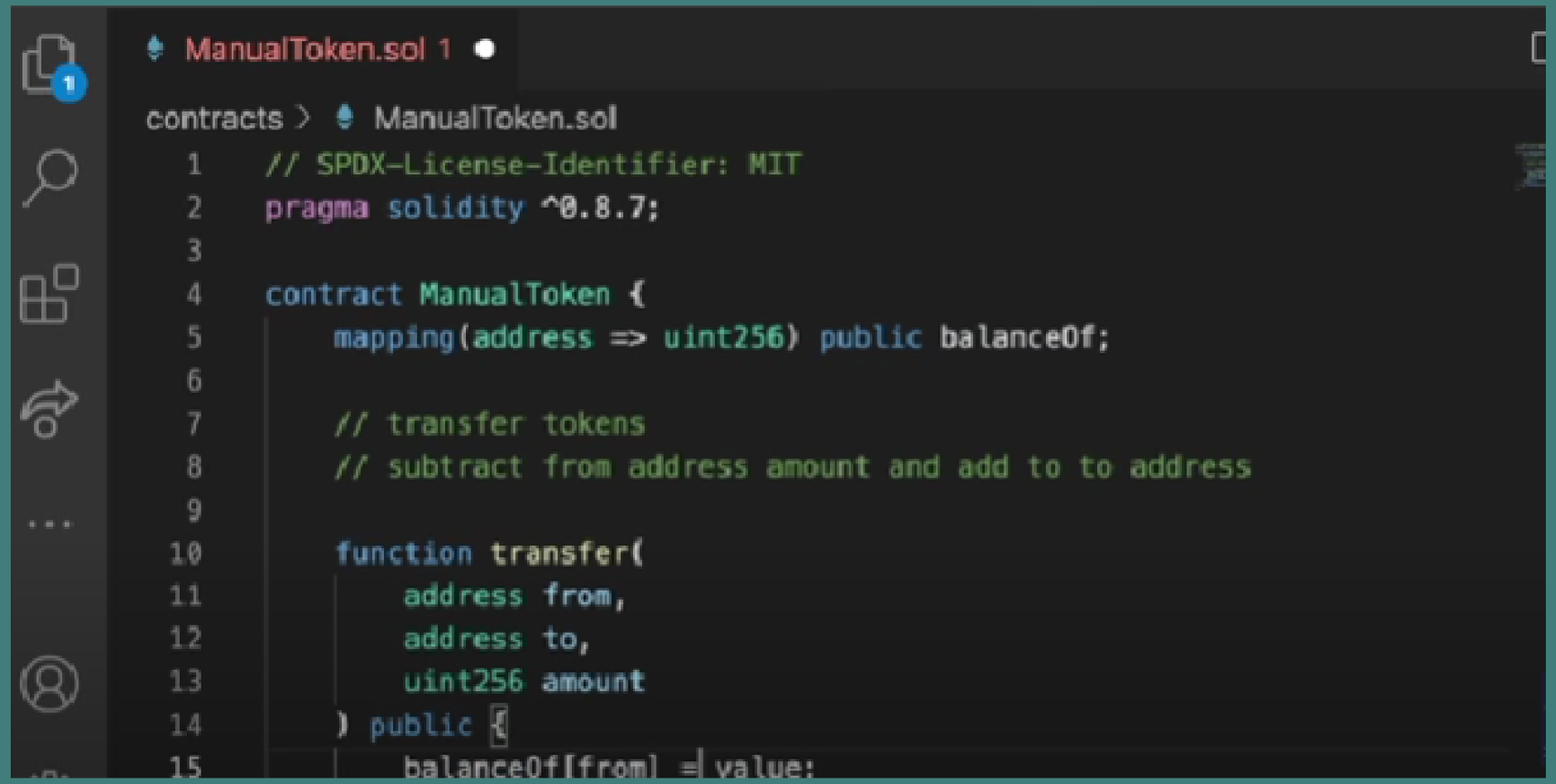


Hardhat Starter Kit

memberikan rincian repo template Chainlink Hardhat-starter-kit



Hardhat ERC20s



DeFI

DeFi menggunakan teknologi blockchain serta cryptocurrency yang menghapus adanya perantara saat bertransaksi, ungkap Mike Edward seorang CEO perusahan investasi DeFi kedua di Inggris bernama Dispersion Holdings. Tidak ada aktor di tengah itu membuat aktivitas berjalan menjadi lebih cepat, murah, efisien serta aman.

Aave

Aave adalah protokol keuangan terdesentralisasi (DeFi), open-source dan non-kustodian pada blockchain smart contract Ethereum yang memberikan akses pada setiap user Aave ini dapat memberikan pinjaman dan meminjam aset digital. Aave ini didirikan pada September 2018 setelah penawaran koin awal (ICO) yang sukses tahun sebelumnya untuk token ETHLend-nya yang mengumpulkan USD $16,2 juta.

Aave Protocol

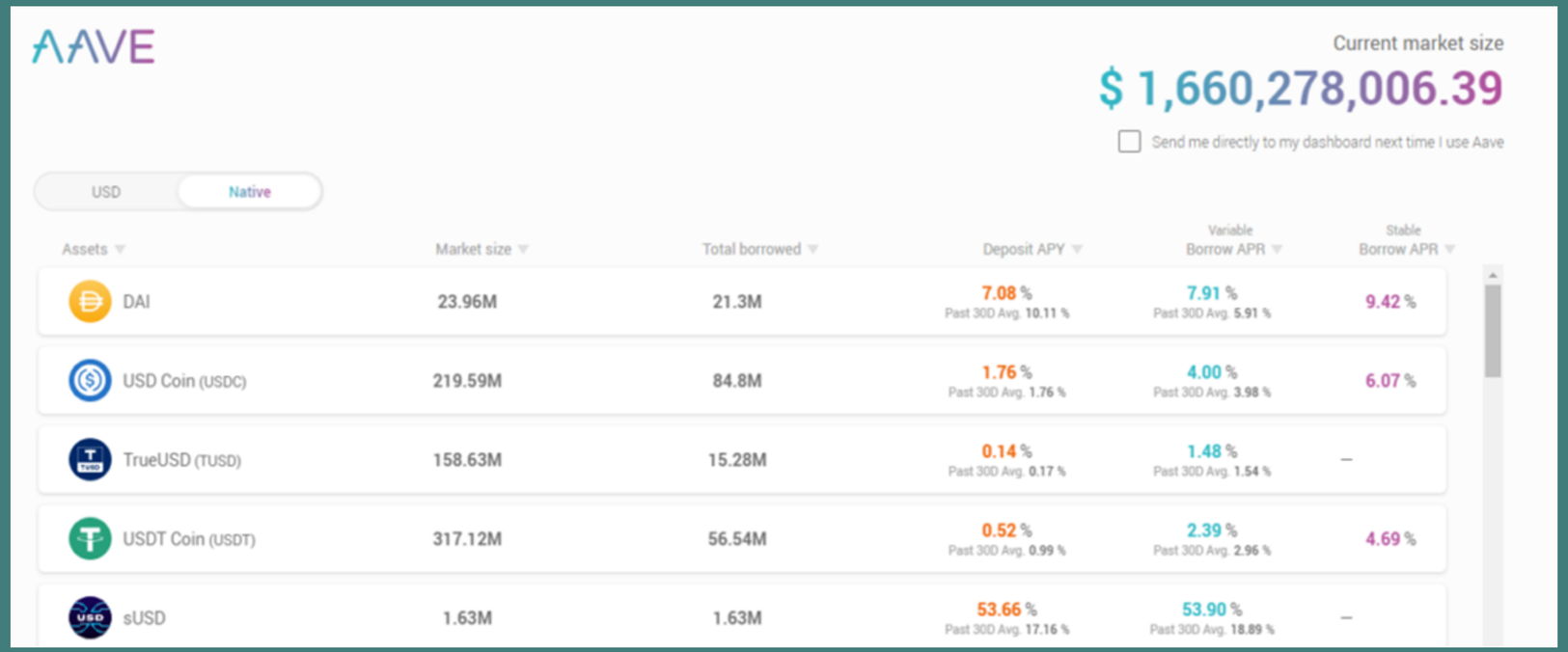
Aave Protocol adalah open source protokol non-penahanan yang memungkinkan pengguna untuk membuat pasar uang terdesentralisasi mereka sendiri di blockchain Ethereum.

Figure 1: The Aave Protocol

Token Crypto yang Tersedia di Aave

- DAI, USD Coin (USDC)
- TrueUSD (TUSD),
- USDT Coin (USDT)
- sUSD,
- Binance USD (BUSD)
- Ethereum (ETH)
- Basic Attention Token (BAT) dan sebagainya

Setiap aset memiliki persyaratan agunan yang berbeda dikarenakan perbedaan volatilitas harga.



NFTs

NFT ialah token digital yang ditautkan ke sistem besar blockchain. Sebenarnya nggak jauh beda dengan beberapa aset mata uang kripto, bedanya cuma NFT tidak bisa dipertukarkan, tapi bisa diperjual lbelikan. Sejauh ini NFT banyak digunakan untuk mewakili sebuah barang, kebanyakan adalah karya seni di forum digital. Bisa dibilang NFT adalah sebuah sertifikasi kepemilikan sebuah barang.

How to sell

NFT juga dijual di pasar dan prosesnya bervariasi dari platform ke platform. Kamu harus mengunggah konten dulu ke pasar, lalu ikuti petunjuk untuk mengubahnya menjadi NFT. Selanjutnya kamu akan diminta memasukkan hal-hal spesifik seperti deskripsi pekerjaan dan harga yang disarankan. Sebagian besar NFT dibeli menggunakan ethereum, tapi juga dapat dibeli dengan token ERC-20 lainnya seperti WAX dan Flow

How to buy

Jangan asal beli NFT, kamu harus memutuskan dari pasar mana kamu belinya, jenis dompet digital apa yang dipake, dan jenis cryptocurrency apa yang diperlukan. Beberapa pasar NFT yang paling umum termasuk OpenSea, Mintable, Nifty Gateway, dan Rarible. Ada juga pasar khusus untuk jenis NFT yang lebih spesifik, seperti NBA Top Shot untuk sorotan video bola basket atau barang berharga untuk melelang tweet seperti Dorsey yang saat ini siap untuk ditawar.

NextJS NFTs Marketplace Front End

- Set up Tailwind CSS

```
npm install -D tailwindcss@latest postcss@latest autoprefixer@latest
npx tailwindcss init -p
```

- Now configure paths in tailwind.config.js

```
module.exports = {
  content: [
    "./pages/**/*.{js,ts,jsx,tsx}",
    "./components/**/*.{js,ts,jsx,tsx}",
  ],
  theme: {
    extend: {},
  },
  plugins: [],
}
```

- **Now delete code in styles/globals.css and update it as given:**

```
@tailwind base;
@tailwind components;
@tailwind utilities;
```

- **Go to the pages folder and create four .js files as given create.js , myNft.js , dashboard.js and resellNft.js .**
- **Now let's create routes, open _app.js and edit as given**

Hardhat Upgrades

TS hardhat.config.ts × | Base.sol | Upgradeable.sol

TS hardhat.config.ts > ...

```ts
import { config as dotEnvConfig } from "d
dotEnvConfig();
import "@nomiclabs/hardhat-etherscan";
import "@nomiclabs/hardhat-waffle";
import "@nomiclabs/hardhat-ethers";
import "@typechain/hardhat";
import "@openzeppelin/hardhat-upgrades";
```

contracts > Base.sol

```sol
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

import "@openzeppelin/contracts/access/Ownable.sol";
import "@openzeppelin/contracts/security/ReentrancyGuard.sol";

contract Base is ReentrancyGuard, Ownable {
    uint256 private storedValue;

    event Received(address, uint256);

    receive() external payable {
        emit Received(msg.sender, msg.value);
    }

    function setStoredValue(uint256 newValue) external {
        storedValue = newValue;
    }

    function getStoredValue() external view returns (uint256) {
        return storedValue;
    }

    function withdraw(address to, uint256 value)
        public
        payable
        onlyOwner
        nonReentrant
    {
        (bool sent, ) = to.call{value: value}("");
        require(sent, "Failed to send Ether");
    }
}
```

Hardhat DAOs

Following the same pattern from our general tutorial, the DAO's contracts addresses are defined in interface/packages/contracts/src/addresses.js:

```
const networksAndAddresses = {

  hardhat: {
    network: "Hardhat Network",
    tokenNetwork: {
      address: "0xa513E6E4b8f2a923D98304ec87F64353C4D5C853", // default Hardhat Network deployment address with npx hardhat node
    },
    dao: {
      governor: {
        address: "0x9fE46736679d2D9a65F0992F2272dE9f3c7fa6e0", // default Hardhat Network deployment address with npx hardhat node
      },
      daoToken: {
        address: "0xe7f1725E7734CE288F8367e1Bb143E90bb3F0512", // default Hardhat Network deployment address with npx hardhat node
      }
    }
  },

...
```

Note, again, that only the Governor and DAO Tokens contract addresses are referenced. The Timelock contract is hidden in the background.Then, in services/contract-functions.js, we get the contracts addresses with

```
import { addresses, abis } from "@project/contracts";
```

And later we reference the contracts in methods such as queue, execute, and cancel:

```
export async function queue(w3provider, proposalId) {
  let signer = w3provider.getSigner();
  let contract = new Contract(addresses.dao.governor.address, abis.governor.abi, w3provider);
  let signed = await contract.connect(signer);

//... then execute the queue function in try/catch block:
    let queueCall = await signed.queue(proposalId);
```

Which are then imported into the user interface such as the queue-execute-propose-modal.js:

```
async function submitQueue() {
    setIsSubmitting(true);
    let newResult;
    try {
      let queueCall = await queue(
        props.provider,
        props.id
      );
      newResult = queueCall.toString()
    } catch (e) {
      newResult = e.message;
    }
    setIsSubmitting(false);
    setResult(newResult);
  }
```

Security and Auditing

adalah evaluasi sistematis keamanan sistem informasi perusahaan dengan mengukur seberapa baik kesesuaiannya dengan serangkaian kriteria yang ditetapkan. Audit menyeluruh biasanya menilai keamanan konfigurasi fisik dan lingkungan sistem, perangkat lunak, proses penanganan informasi, dan praktik pengguna.

Why are security audits important?

Ada beberapa alasan untuk melakukan audit keamanan. Mereka termasuk enam tujuan ini:

- Identifikasi masalah dan celah keamanan, serta kelemahan sistem.
- Tetapkan dasar keamanan yang dapat dibandingkan dengan audit di masa mendatang.
- Mematuhi kebijakan keamanan organisasi internal.
- Mematuhi persyaratan peraturan eksternal.
- Tentukan apakah pelatihan keamanan memadai.
- Identifikasi sumber daya yang tidak perlu.

Types of security audits

Internal audits.

Dalam audit ini, bisnis menggunakan sumber dayanya sendiri dan departemen audit internal. Audit internal digunakan ketika sebuah organisasi ingin memvalidasi sistem bisnis untuk kepatuhan kebijakan dan prosedur.

External audits.

Dengan audit ini, organisasi luar dibawa untuk melakukan audit. Audit eksternal juga dilakukan ketika sebuah organisasi perlu memastikan bahwa itu sesuai dengan standar industri atau peraturan pemerintah.

What systems does an audit cover?

Selama audit keamanan, setiap sistem yang digunakan organisasi dapat diperiksa kerentanannya di area berikut:

- Network vulnerabilities.
- Security controls.
- Encryption.
- Software systems
- Architecture management capabilities
- Telecommunications controls.
- Systems development audit.
- Information processing

Steps involved in a security audit

Agree on goals

Libatkan semua pemangku kepentingan dalam diskusi tentang apa yang harus dicapai dengan audit

Define the scope of the audit.

Buat daftar semua aset yang akan diaudit, termasuk peralatan komputer, dokumentasi internal, dan data yang diproses.

Conduct the audit and identify threats.

Daftar potensi ancaman yang terkait dengan setiap Ancaman dapat mencakup hilangnya data, peralatan, atau catatan melalui alam

Evaluate security and risks.

Menilai risiko dari setiap ancaman yang teridentifikasi yang terjadi, dan seberapa baik organisasi dapat bertahan melawannya.