Guide to ERC721 NFTs

Disarankan sudah membaca Algorithm and Data Types (Solidity) untuk pemahaman dasar seputar terminologi dalam pemrograman dan tipe data, serta mencoba membuat ERC20 token (ada di modul live class). Syntax di sini mengikuti versi Solidity terbaru.

Contents

ERC721

ERC721 adalah standar untuk NFT yang merepresentasikan digital asset yang unik. Guide ini akan membimbing untuk membuat ERC721 NFT dengan Hardhat, IPFS, nftexport.io di Sepolia testnet. Sebelum mulai, pastikan environment sudah di setup untuk local smart contract deployment dan punya Metamask extension. Disarankan pake account khusus development.

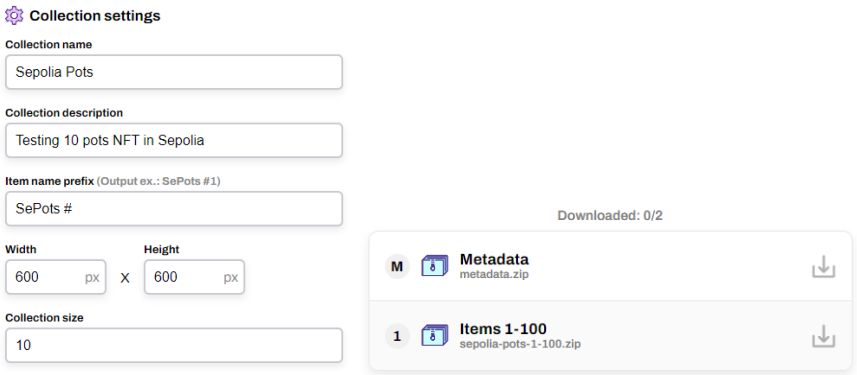
Initiate New Project

Bikin folder baru untuk project-nya, kemudian buka terminal dari directory folder dan run command berikut:

```
npx hardhat init // pilih empty hardhat config
mkdir contracts scripts // atau langsung bikin folder “contracts” dan “scripts” secara manual
```

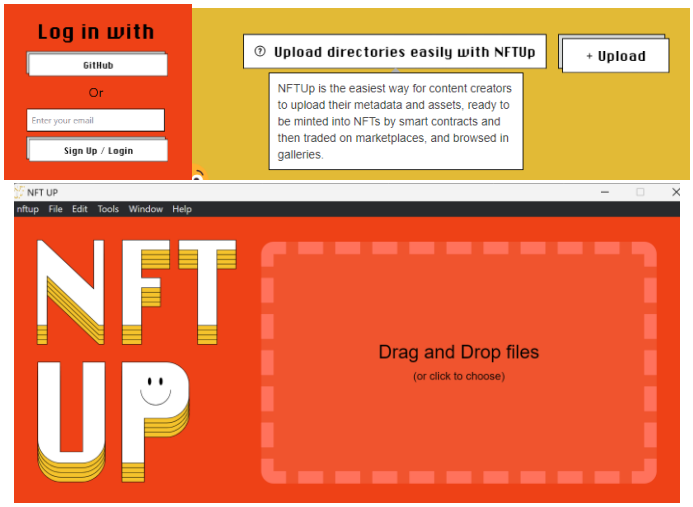
Making the NFTs

NFT Export adalah tools gratis untuk membuat generative NFT dimana ia akan menghasilkan generative art beserta metadata-nya. Metadata adalah data dari suatu data, dalam hal ini adalah data tentang NFT kita nanti. Biar cepat, pake gambar demo saja, tapi kalian bisa pakai gambar sendiri juga. Atur “Collection settings” lalu lanjut sampai tahap download hasil yang di-generate dan unzip file-nya. Download yang “Items..” saja karena itu sudah include gambar dan metadata.

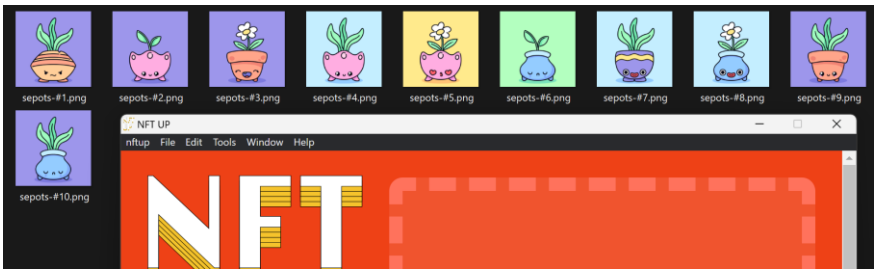


Deploying to IPFS

Bikin account di NFT Storage untuk mendapatkan API key, lalu kalian bisa upload files ke IPFS langusng dari website atau via aplikasinya, NFTUp.



Upload file gambarnya terlebih dahulu. Setelah upload selesai, kalian akan dapat URL IPFS-nya.



Ini sample URL gambarnya:

https://bafybeigyv6t2hwojyumdmgb2qfmnukvwh4ex7laipa7kdttfmtrlkwfrdm.ipfs.nftstorage.link/ipfs/bafybeigyv6t2hwojyumdmgb2qfmnukvwh4ex7laipa7kdttfmtrlkwfrdm/sepots-%231.png

Karena gambarnya diupload berbarengan, maka semuanya ada di folder IPFS yang sama, sekarang kita perlu pre-process metadata. 2 hal yang perlu diubah adalah value dari field image yang akan ditambahkan URL IPFS dan nama file metadata menjadi ID dari NFT karena logic default ERC721 untuk mengetahui metadata dari NFT ID tertentu (function tokenURI) adalah dengan menggabungkan baseURI (di sini kita pakai URL folder IPFS metadata) dan NFT ID. Cara lain adalah dengan meng-override function ini, tapi lebih mudah me-rename file metadata.

```
function tokenURI(uint256 tokenId) public view virtual returns (string memory) {
    _requireOwned(tokenId);

    string memory baseURI = _baseURI();
    return bytes(baseURI).length > 0 ? string.concat(baseURI, tokenId.toString()) : "";
}
```

Buka file metadata.ts dari GitHub Repository ini dan ubah value **directoryPath** dan **regex** sesuai format nama file metadata kalian. Run file metadata.ts dengan command:

```
node metadata.ts
```

Upload metadata ke IPFS. Nama file metadata hanya angka dan value dari field image harusnya bisa dibuka. Berikut URL metadatanya:

https://bafybeiaul4fgnopenv3pwegicktesgxre77vqaq74u4pdguyyuctkfbf5u.ipfs.nftstorage.link/1

Code the Contract

Install dependencies. Command di bawah install OpenZeppelin untuk Solidity v0.8.20 jadi sesuaikan versinya di hardhat.config.js.

```
npm i -D @openzeppelin/contracts
npm i -D @nomicfoundation/hardhat-toolbox
npm i -D dotenv
```

Kalau sudah clone repository, tinggal jalanin command berikut untuk install semua dependencies:

```
npm i
```

Code ERC721 contract, nama, symbol, dan baseURI disesuaikan saja.

```
// SPDX-License-Identifier: SEE LICENSE IN LICENSE
pragma solidity 0.8.20;

import "@openzeppelin/contracts/token/ERC721/ERC721.sol";
import "@openzeppelin/contracts/token/ERC721/extensions/ERC721Enumerable.sol";

contract Pots is ERC721Enumerable {
    uint256 public constant MAX_SUPPLY = 10; // set max supply of the NFT

    constructor() ERC721("SePots", "POTS") {}

    // set base URI to IPFS folder of metadata
    function _baseURI() internal pure override returns (string memory) {
        return

"https://bafybeiaul4fgnopenv3pwegicktesgxre77vqaq74u4pdguyyuctkfbf5u.ipfs.nftstorage.link/";
    }

    function mint() external {
        // should add restriction for real project, such as amount of token required to pay or
onlyOwner can initial mint
        require(totalSupply() < MAX_SUPPLY); // validate supply
        _safeMint(msg.sender, totalSupply() + 1); // safeMint ensures receiver can receive NFT
    }
}
```

Compile dengan command:

```
npx hardhat compile
```

Config and Environment Variables

Bikin file **.gitignore** dan **.env**, serta update config Hardhat untuk deployment ke Sepolia.

```
// .gitignore
.env
node_modules
cache
artifacts
```

```
// .env
SEPOLIA_API_KEY= // bisa ambil dari Infura
PRIVATE_KEY= // private key wallet deployer
ETHERSCAN_API_KEY= // bikin account Etherscan
```

```
/** @type import('hardhat/config').HardhatUserConfig */
require("@nomicfoundation/hardhat-toolbox");
require('dotenv').config();

const PRIVATE_KEY = process.env.PRIVATE_KEY ?? "";
const SEPOLIA_API_KEY = process.env.SEPOLIA_API_KEY ?? "";
const ETHERSCAN_API_KEY = process.env.ETHERSCAN_API_KEY ?? "";

module.exports = {
  solidity: "0.8.20",
  networks: {
    hardhat: {},
    localnet: {
      url: "http://127.0.0.1:8545/"
    },
    sepolia: {
      url: SEPOLIA_API_KEY,
      accounts: [PRIVATE_KEY]
    }
  },
  etherscan: {
    apiKey: ETHERSCAN_API_KEY
  }
};
```

Deployment

Bikin script **deploy.js** untuk deployment dalam folder **scripts**.

```
async function main() {
    const factory = await ethers.getContractFactory('Pots');
    const nft = await factory.deploy();
    const nftAddress = await nft.getAddress();
    console.log("Address: ", nftAddress);
}

main();
```

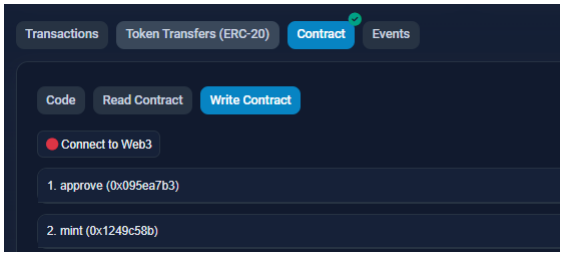
Lalu run command deployment:

```
npx hardhat run --network sepolia scripts/deploy.js
npx hardhat verify --network sepolia <contract address>
```

Kalau kena *ethers is not defined*, tambahin **const { ethers } = require("hardhat");** .

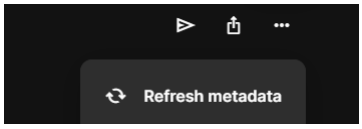
Testing

Buka contract di scanner Sepolia dan klik **Contract** dan coba **mint**. Kalau belum punya Sepolia ETH, bisa ke faucet Sepolia ETH (disarankan pakai VPN/Brave browser).



Kalau sudah ada NFT yang di mint, bisa cek [OpenSea testnet] dan cari nama **Collection**-nya atau cek **Profile** untuk lihat NFT yang sudah di mint. Kalau gambarnya tidak langsung muncul, pastikan saja **tokenURI**-nya sudah benar. Bisa di cek dari etherscan, bagian **Contract**, klik **Read Contract**.

Terkadang load metadata-nya agak lama, bisa coba klik **Refresh Metadata**.



And you're done! Congrats.