Hardhat NFTs :

Standard untuk Non-Fungible Token pada EIP-712 memerlukan beberapa spesifikasi yang dapat dilihat pada website EIP

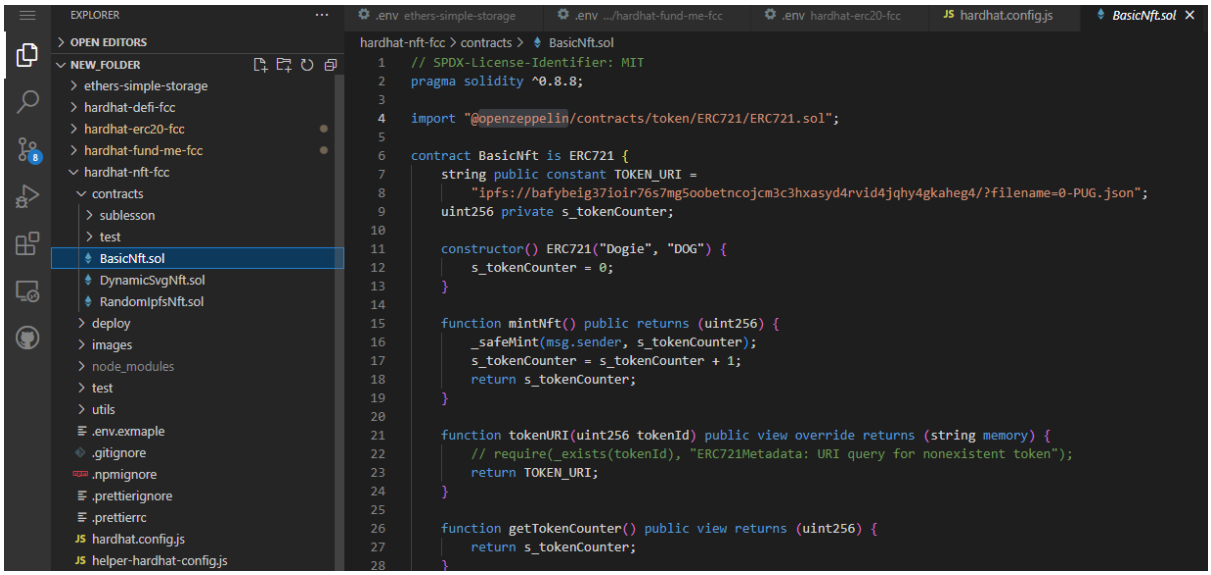
Specification

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY",

Every ERC-721 compliant contract must implement the `ERC721` and `ERC165` interfaces (subject to "caveats" below):

```
pragma solidity ^0.4.20;

/// @title ERC-721 Non-Fungible Token Standard
/// @dev See https://eips.ethereum.org/EIPS/eip-721
///  Note: the ERC-165 identifier for this interface is 0x80ac58cd.
interface ERC721 /* is ERC165 */ {
    /// @dev This emits when ownership of any NFT changes by any mechanism.
    ///  This event emits when NFTs are created (`from` == 0) and destroyed
    ///  (`to` == 0). Exception: during contract creation, any number of NFTs
    ///  may be created and assigned without emitting Transfer. At the time of
    ///  any transfer, the approved address for that NFT (if any) is reset to none.
    event Transfer(address indexed _from, address indexed _to, uint256 indexed _tokenId);

    /// @dev This emits when the approved address for an NFT is changed or
    ///  reaffirmed. The zero address indicates there is no approved address.
    ///  When a Transfer event emits, this also indicates that the approved
    ///  address for that NFT (if any) is reset to none.
    event Approval(address indexed _owner, address indexed _approved, uint256 indexed _tokenId);

    /// @dev This emits when an operator is enabled or disabled for an owner.
    ///  The operator can manage all NFTs of the owner.
    event ApprovalForAll(address indexed _owner, address indexed _operator, bool _approved);

    /// @notice Count all NFTs assigned to an owner
    /// @dev NFTs assigned to the zero address are considered invalid, and this
    ///  function throws for queries about the zero address.
    /// @param _owner An address for whom to query the balance
    /// @return The number of NFTs owned by `_owner`, possibly zero
    function balanceOf(address _owner) external view returns (uint256);

    /// @notice Find the owner of an NFT
    /// @dev NFTs assigned to zero address are considered invalid, and queries
    ///  about them do throw.
    /// @param _tokenId The identifier for an NFT
    /// @return The address of the owner of the NFT
```

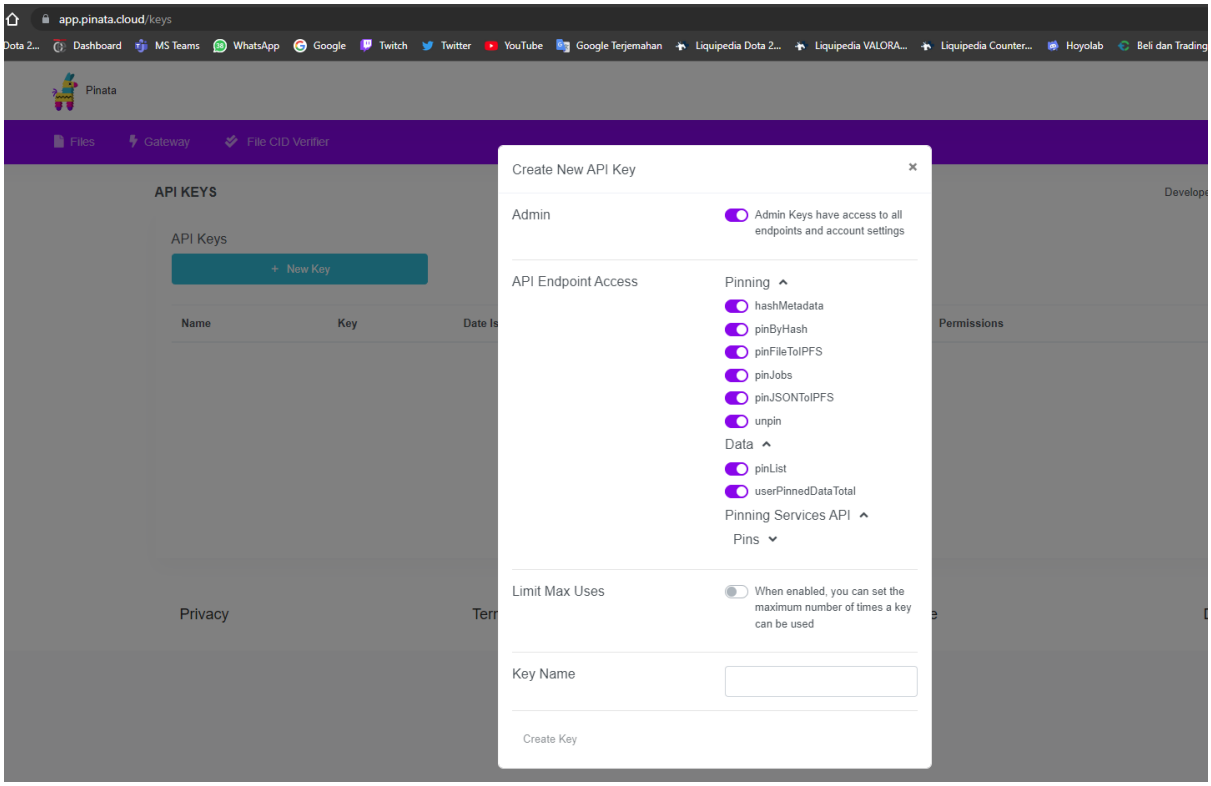
Spesifikasi ERC721 tersedia dalam openzeppelin sehingga kita hanya perlu mengimportnya kedalam smart contract



Lalu masukan id subscription chainlink dalam helper-hardhat-config.ts

```
    // Default one is ETH/USD contract on Kovan
    4: {
        name: "rinkeby",
        ethUsdPriceFeed: "0x8A753747A1Fa494EC906cE90E9f37563A8AF630e",
        vrfCoordinatorV2: "0x6168499c0cFfCaCD319c818142124B7A15E857ab",
        gasLane: "0xd89b2bf150e3b9e13446986e571fb9cab24b13cea0a43ea20a6049a85cc807cc",
        callbackGasLimit: "500000", // 500,000 gas
        mintFee: "10000000000000000", // 0.01 ETH
        subscriptionId: "8118", // add your ID here!
    },
}
```

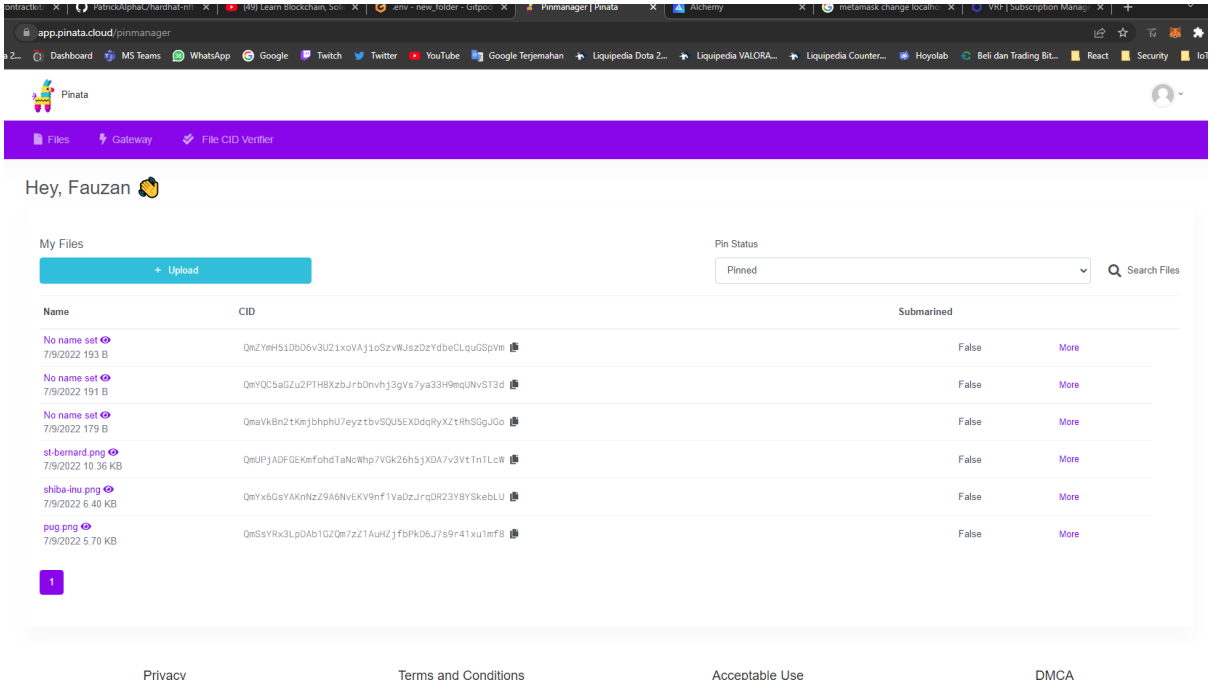
Membuat akun pinata dan generate API key dan masukan data tersebut ke .env



Setelah itu dapat kita jalankan script untuk mengupload gambar ke pinata dengan yarn hardhat deploy --tags randomipfs,mocks

```
gitpod /workspace/new_folder/hardhat-nft-fcc (main) $ yarn hardhat deploy --tags randomipfs,mocks
yarn run v1.22.19
warning package.json: No license field
$ /workspace/new_folder/hardhat-nft-fcc/node_modules/.bin/hardhat deploy --tags randomipfs,mocks
Nothing to compile
Local network detected! Deploying mocks...
deploying "VRFCoordinatorV2Mock" (tx: 0xb1811099ad4653047586f3d30dea9d8cae51f74a48277c9ab76ac8ab3ed95b61)...: deployed at 0x5FbDB2315678afecb367f032d93F642f64180aa3 with 1803306 gas
deploying "MockV3Aggregator" (tx: 0x39b14b58ecc5340285c257614ff14eb9f80c75fa4148dd8d33441bfbc135b710)...: deployed at 0xe7f1725E7734CE288F8367e1Bb143E90bb3F0512 with 569671 gas
Mocks Deployed!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
You are deploying to a local network, you'll need a local network running to interact
Please run `yarn hardhat console --network localhost` to interact with the deployed smart contracts!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Uploading pug...
Uploading shiba-inu...
Uploading st-bernard...
Token URIs uploaded! They are:
[
  'ipfs://QmaVkBn2tKmjbhphU7eyztbvSQU5EXDdqRyXZtRhSGgJGo',
  'ipfs://QmYQC5aGZu2PTH8XzbJrbDnvhj3gVs7ya33H9mqUNvST3d',
  'ipfs://QmZYmH5iDbD6v3U2ixoVAjioSzvWJszDzYdbeCLquGSpVm'
]
----------------------------------------------------
deploying "RandomIpfsNft" (tx: 0x81c5a85ecf8a58536ab560586a066aeac3b2f2c23c89ad6720e6edfbdd27a867)...: deployed at 0xDc64a140Aa3E981100a9becA4E685f962f0cF6C9 with 3846074 gas
Done in 10.77s.
gitpod /workspace/new_folder/hardhat-nft-fcc (main) $
```

Bila upload sukses, kita dapat melihat gambar yang selesai terupload ke pinata.



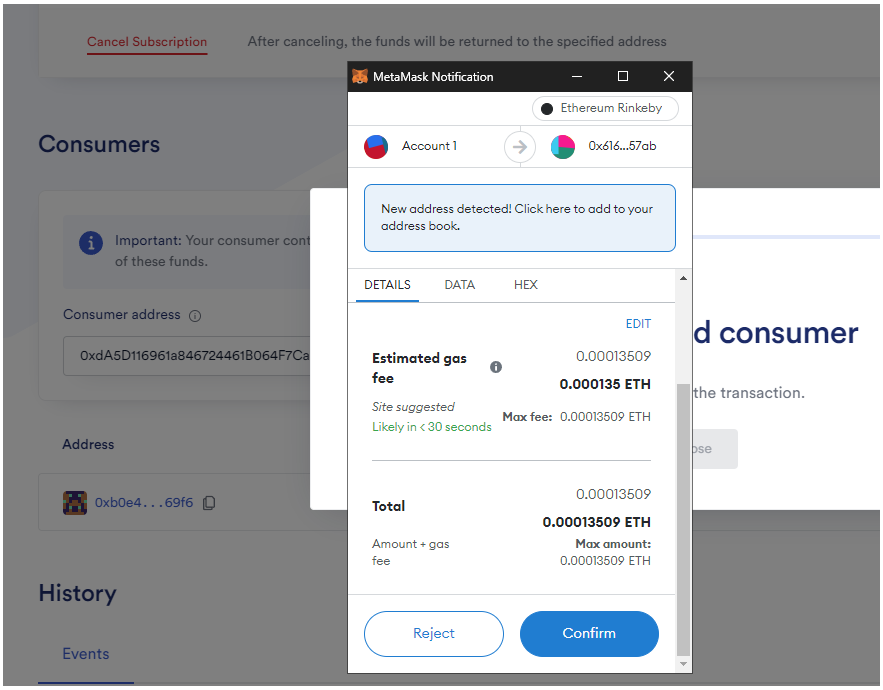
Metadata dari gambar tersebut juga dapat dilihat dengan memilih highlit ungu.


```
{"name":"st-bernard","description":"An adorable st-bernard pup!","image":"ipfs://QmUPjADFGEKmfohdTaNcWhp7VGk26h5jXDA7v3VtTnTLcW","attributes":[{"trait_type":"Cuteness","value":100}]}
```


Untuk menghubungkannya dengan test network, kita perlu mendeploy smart contract ini dengan perintah yarn hardhat deploy --network rinkeby --tags main.

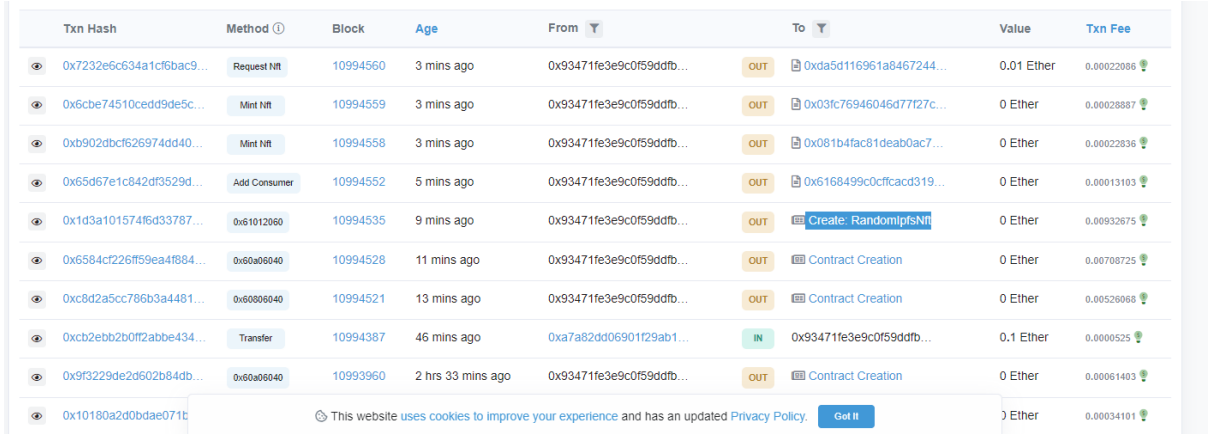
```
Compiled 24 Solidity files successfully
Successfully submitted source code for contract
contracts/DynamicSvgNft.sol:DynamicSvgNft at 0x03fC76946046d77f27c20fc413641889694F4BEF
for verification on the block explorer. Waiting for verification result...

Already verified!
Uploading pug...
Uploading shiba-inu...
Uploading st-bernard...
Token URIs uploaded! They are:
[
  'ipfs://QmaVkBn2tKmjbhphU7eyztbvSQU5EXDdqRyXZtRhSGgJGo',
  'ipfs://QmYQC5aGZu2PTH8XzbJrbDnvhj3gVs7ya33H9mqUNvST3d',
  'ipfs://QmZYmH51DbD6v3U2ixoVAjioSzvWJszDzYdbeCLquGSpVm'
]
----------------------------------------------------
deploying "RandomIpfsNft" (tx: 0x1d3a101574f6d3378744d6fab427fdbf6c885bdf1e71e21d623ffc675b2b30f0)...: deployed at 0xdA5D116961a846724461B064F7CafdD0198d4221 with 3846086 gas
Verifying...
Verifying contract...
Nothing to compile
Successfully submitted source code for contract
contracts/RandomIpfsNft.sol:RandomIpfsNft at 0xdA5D116961a846724461B064F7CafdD0198d4221
for verification on the block explorer. Waiting for verification result...

Successfully verified contract RandomIpfsNft on Etherscan.
https://rinkeby.etherscan.io/address/0xdA5D116961a846724461B064F7CafdD0198d4221#code
Done in 329.19s.
gitpod /workspace/new_folder/hardhat-nft-fcc (main) $
```

Setelah selesai, copy address RandomIpfsNft kedalam counsumer chainlink dan konfirmasi.



Jalankan script mint pada jaringan rinkeby dengan yarn hardhat deploy --tags mint --network rinkeby

Dan bila kita lihat transaksi pada address wallet kita pada etherscan, pilih RandomIpfsNft dan salin addressnya



Pada website testnet.opensea.io, apabila kita memasukan address tersebut maka kita dapat melihat NFT yang telah dideploy