

Decentralized Application NFT Sale

Elissa Gunawan

00000027328

IF671 - Blockchain & Cryptocurrency Class A

Penjelasan Code pada Smart Contract (Marketplace.sol)

Import library **@openzeppelin** untuk ERC721.

```
import "../client/node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol";
import "../client/node_modules/@openzeppelin/contracts/utils/Counters.sol";
import "../client/node_modules/@openzeppelin/contracts/access/Ownable.sol";
import
"../client/node_modules/@openzeppelin/contracts/token/ERC721/extensions/ERC721URIStorage.sol";
```

Meng-inherit **ERC721URIStorage** dan **Ownable** agar smart contract mewarisi sifat dari standard ERC721.

```
contract Marketplace is ERC721URIStorage, Ownable{
```

Deklarasi variable untuk **productCount** untuk menghitung banyaknya produk dan **tokenId** untuk menyimpan hasil dari minting ERC721.

```
uint public productCount = 0;
uint public tokenId;
```

Variable **mapping** yang memasang uint atau integer dari product sebagai dengan struct product sebagai value.

```
mapping(uint => Product) public products;
```

Constructor merupakan fungsi yang hanya dijalankan sekali pada saat smart contract di deploy dimana di sini mendefinisikan nama sebagai NFTMarket dan simbol dari NFT sebagai Elissa.

```
constructor() ERC721("NFTMarket", "Elissa"){}
```

Struct **Product** merupakan struktur data yang menyimpan data-data product, yaitu id, name, description, imageurl, tokenId, price, owner, dan purchased. Yang memiliki fungsi masing-masing seperti id untuk id product, name untuk nama produk, description untuk deskripsi produk, imageurl

untuk link NFT, tokenId untuk menyimpan hasil minting, price untuk harga produk, owner untuk address pemilik produk, purchased untuk true atau false mengenai barang sudah terjual atau belum.

```
struct Product {
    uint id;
    string name;
    string description;
    string imageUrl;
    uint tokenId;
    uint price;
    address payable owner;
    bool purchased;
}
```

Event merupakan suatu kejadian yang terjadi pada smart contract dan smart contract tersebut ingin berkomunikasi dengan dApps. Dimana kejadian tersebut adalah pada saat produk dibuat (**ProductCreated**) dan produk terjual (**ProductPurchased**).

```
event ProductCreated(
    uint id,
    string name,
    string description,
    string imageUrl,
    uint tokenId,
    uint price,
    address payable owner,
    bool purchased
);

event ProductPurchased(
    uint id,
    string name,
    string description,
    string imageUrl,
    uint tokenId,
    uint price,
    address payable owner,
    bool purchased
);
```

Fungsi **mintNFT** merupakan fungsi untuk melakukan minting NFT dimana menerima 2 argumen yaitu 'recipient' sebagai address yang akan menerima NFT dan 'tokenURI' sebagai link dari NFT yang akan di-mint. Setiap fungsi mintNFT dipanggil maka variable 'tokenIds' akan terus bertambah sehingga id dari setiap NFT yang dibuat akan selalu unik.

```
function mintNFT(address recipient, string memory tokenURI) public returns (uint256){
    _tokenIds.increment();
```

```

uint256 newItemId = _tokenIds.current();
_mint(recipient, newItemId);
_setTokenURI(newItemId, tokenURI);

return newItemId;
}

```

Fungsi **createProduct** merupakan fungsi untuk membuat produk NFT yang menerima beberapa argumen yaitu `_name` untuk nama produk, `_description` untuk deskripsi produk, `_imageurl` untuk link NFT yang ingin dijual, `_price` untuk harga produk. Untuk menjalankan fungsi ini ada 2 persyaratan yaitu `_name` tidak boleh kosong atau panjangnya harus lebih dari 0 dan `_price` tidak boleh kosong atau lebih dari 0. Setiap produk dibuat, maka `productCount` akan terus bertambah. Lalu dilakukan `mintNFT` untuk `_imageurl` dan disimpan dalam `tokenId`, kemudian membuat produknya dan diletakkan dalam struct yang sudah dibuat sebelumnya serta memberitahukan bahwa terjadi sesuatu dalam fungsi `createProduct` atau memberitahukan bahwa ada produk NFT baru dengan menggunakan event `ProductCreated`.

```

function createProduct(string memory _name, string memory _description, string memory _imageurl,
uint _price) public {
    require(bytes(_name).length > 0);
    require(_price > 0);
    productCount++;
    tokenId = mintNFT(msg.sender, _imageurl);
    products[productCount] = Product(productCount, _name, _description, _imageurl, tokenId,
_price, payable(msg.sender), false);
    emit ProductCreated(productCount, _name, _description, _imageurl, tokenId, _price,
payable(msg.sender), false);
}

```

Fungsi **purchasedProduct** merupakan fungsi untuk membeli produk NFT yang hanya menerima 1 argumen yaitu `id` dimana `id` tersebut merupakan `id` produk. `_seller` merupakan address dari owner yang memiliki produk tersebut. Persyaratan menjalankan fungsi ini adalah `id` produknya lebih dari 0 dan produk `id` tidak melebihi dari `productCount` atau sama dengan `productCount`, pastikan memiliki uang yang cukup untuk membeli, bukan merupakan produk yang sudah terjual, dan pembelinya bukan owner produk tersebut. Dan jika sudah terbeli maka owner produknya akan berubah menjadi address pembeli lalu boolean `purchased` akan menjadi `true` dan barang tersebut berubah kepemilikan yang semula dari penjual menjadi milik pembeli, lalu pembeli akan mengirimkan uangnya kepada penjual. Pada saat semua itu terjadi event `ProductPurchased` memberitahukan bahwa telah terjadi sesuatu dalam fungsi `purchasedProduct` atau memberitahukan bahwa seseorang membeli produk NFT.

```

function purchaseProduct(uint _id) public payable {
    Product memory _product = products[_id];
    address payable _seller = _product.owner;
    require(_product.id > 0 && _product.id <= productCount);
    require(msg.value >= _product.price);
}

```

```
require(!_product.purchased);
require(_seller != msg.sender);
_product.owner = payable(msg.sender);
_product.purchased = true;
products[_id] = _product;
_transfer(ownerOf(_product.tokenid), msg.sender, _product.tokenid);
payable (address(_seller)).transfer(msg.value);
emit ProductPurchased(productCount, _product.name, _product.description,
_product.imageurl, _product.tokenid, _product.price, payable(msg.sender), true);
}
```

Cara kerja dan alur kerja website

Requirement :

- GitHub
- NodeJs
- Npm
- Truffle
- Ganache

jalankan beberapa perintah untuk menjalankan website

\$ git clone <https://github.com/ElissaGunawan/NFT-sale.git>

pergi ke project folder

\$ cd NFT-sale

masuk ke dalam folder client

\$ cd client

install truffle (jika belum memiliki truffle)

\$ npm install -g truffle or npm install -g [truffle@5.4.22](https://www.npmjs.com/package/truffle)

download ganache (jika belum memiliki ganache)

<https://trufflesuite.com/ganache/>

install library yang dibutuhkan

\$ npm install

buka ganache

pilih quickstart

jalankan perintah truffle

\$ truffle compile

\$ truffle migrate

jalankan website

\$ npm start

tambahkan jaringan pada metamask

buka metamask

tambah jaringan dengan nama jaringan bebas, rpc url <http://127.0.0.1:7545>, chain id 1337, currency symbol ETH

tambahkan akun dari ganache agar akun memiliki sejumlah currency yang cukup untuk berexplore dalam website

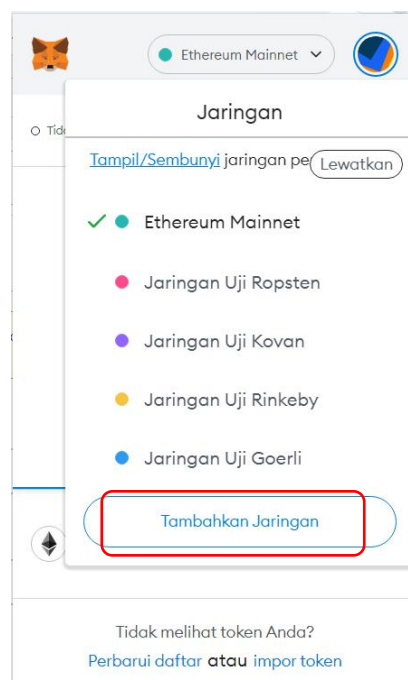
buka metamask

klik akun

pilih impor akun

masukin private key dari ganache

Cara menambahkan jaringan pada metamask (dengan gambar) :



Pertama buka MetaMask dan klik pada jaringan lalu pilih “Tambahkan Jaringan”.

METAMASK Ethereum Mainnet

Pengaturan Search in settings

Jaringan > Tambahkan jaringan

Penyedia jaringan jahat dapat berbohong tentang status blockchain dan merekam aktivitas jaringan Anda. Hanya tambahkan jaringan kustom yang Anda percayai.

Nama Jaringan

URL RPC Baru

ID Rantai

Simbol Mata Uang

URL Block Explorer (Optional)

Batal Simpan

Setelah klik “Tambahkan Jaringan” maka akan dipindahkan ke halaman seperti gambar di atas dapat dilihat bahwa kita memerlukan beberapa data yang perlu di isi.

METAMASK Ethereum Mainnet

Pengaturan Search in settings

Jaringan > Tambahkan jaringan

Penyedia jaringan jahat dapat berbohong tentang status blockchain dan merekam aktivitas jaringan Anda. Hanya tambahkan jaringan kustom yang Anda percayai.

Nama Jaringan ganache

URL RPC Baru http://127.0.0.1:7545

ID Rantai 1337

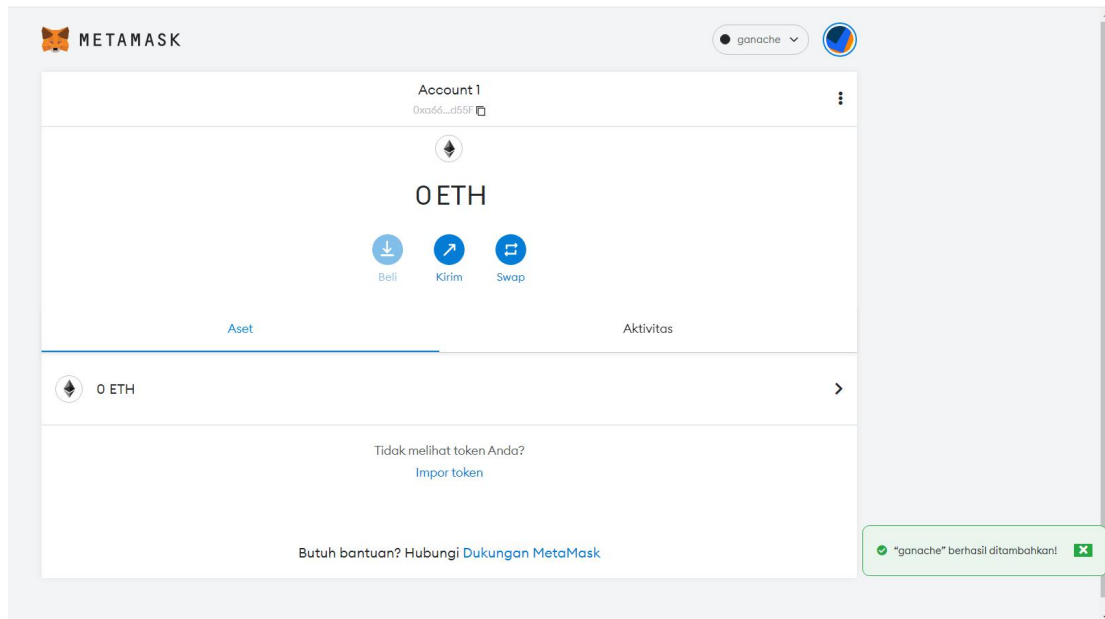
Simbol Mata Uang ETH

The network with chain ID 1337 may use a different currency symbol (CPAY) than the one you have entered. Please verify before continuing.

URL Block Explorer (Optional)

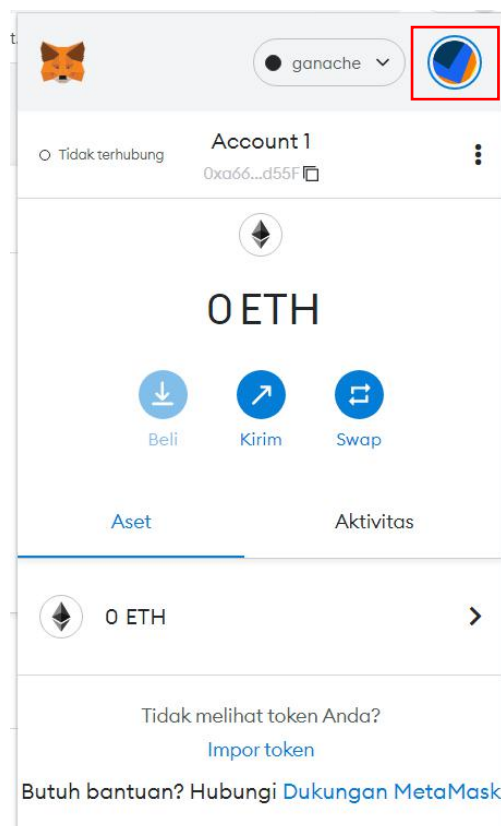
Batal Simpan

Masukkan data-data tersebut dan jika sudah selesai, klik pada tombol “Simpan”.

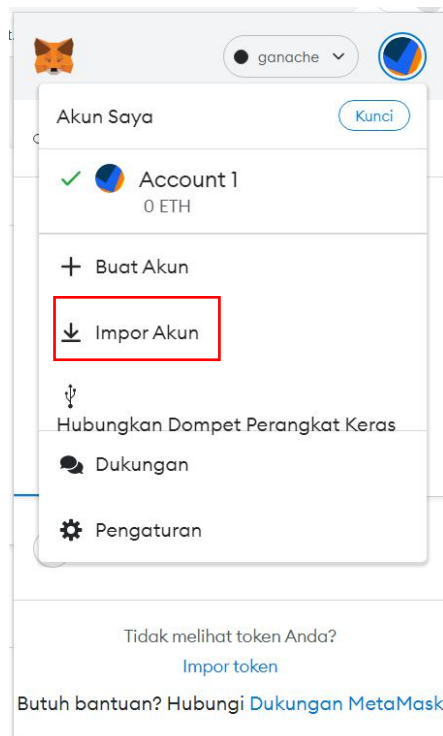


Jika sudah berhasil di simpan maka jaringan “ganache” yang dibuat sudah berhasil ditambahkan pada MetaMask atau terlihat notifikasi kecil pada pojok kanan bawah seperti gambar diatas.

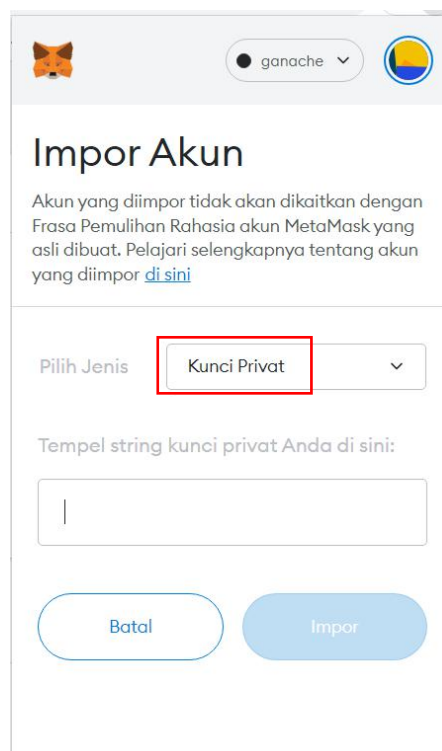
Cara memasukkan akun dummy dari ganache ke dalam metamask (dengan gambar) :



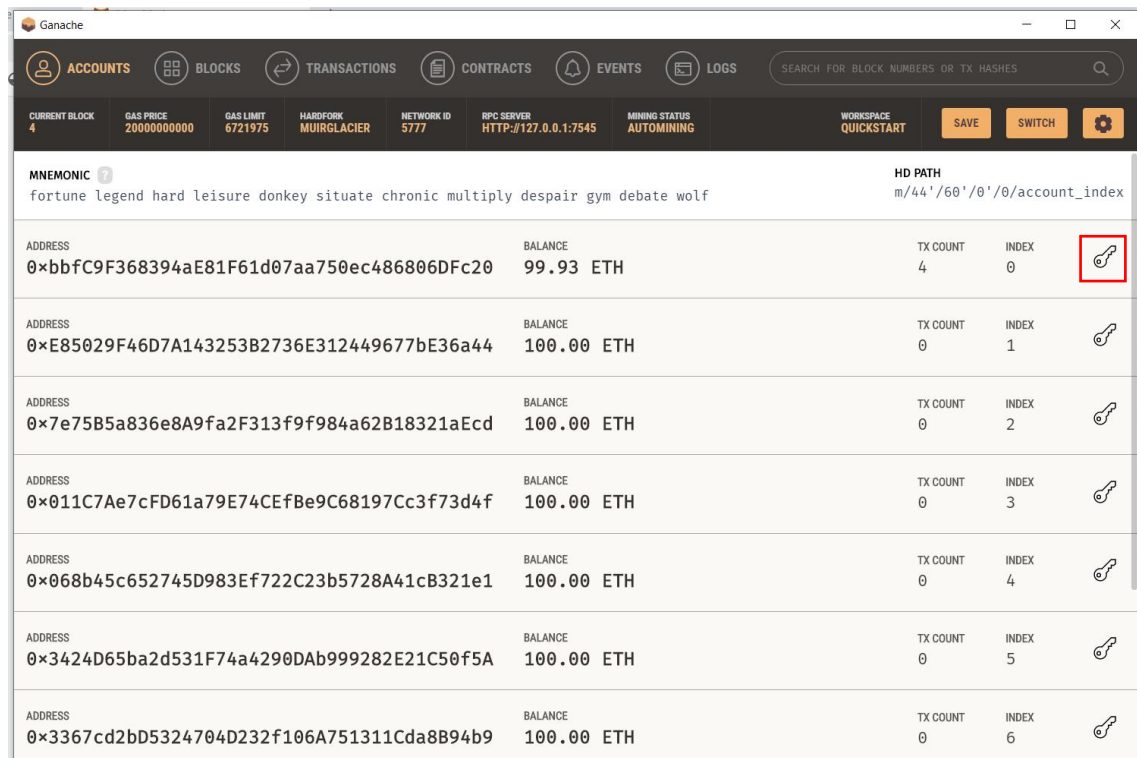
Buka MetaMask lalu klik pada akun atau ikutin gambar di atas.



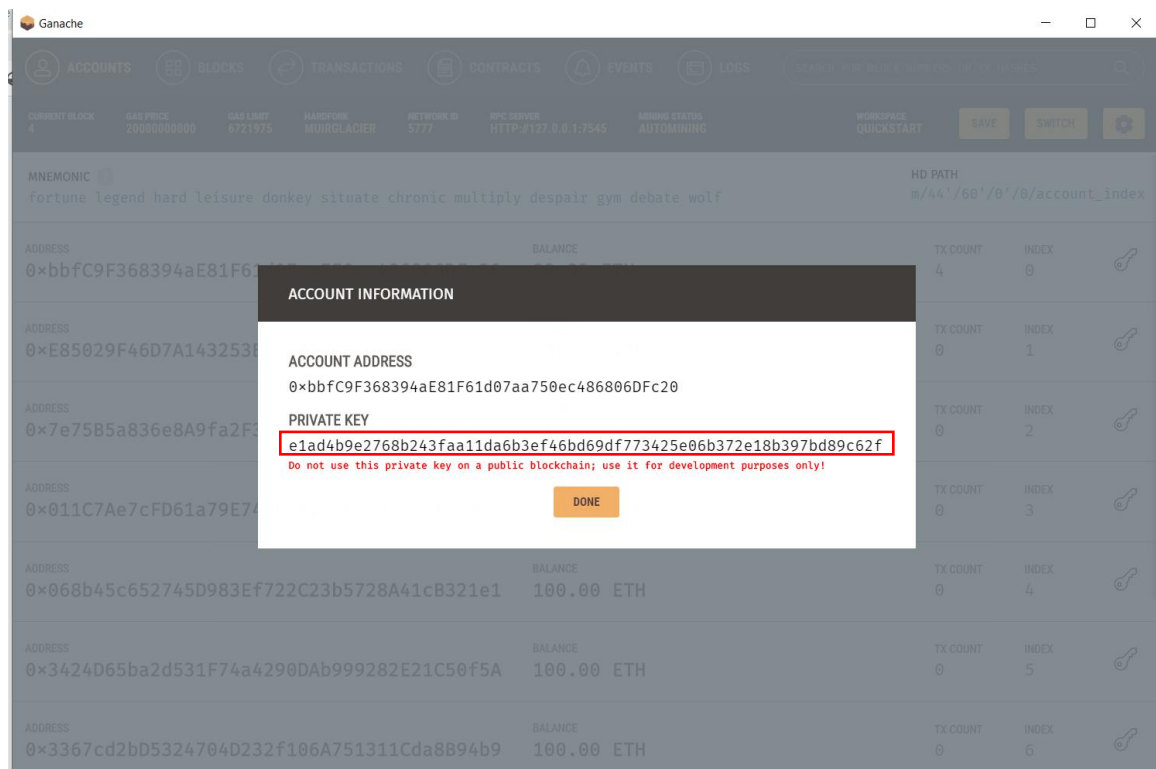
Jika sudah di klik, maka tampilan akan seperti gambar diatas, lalu klik pada “Impor Akun”.



Jika sudah klik pada “Impor Akun” maka tampilan akan seperti ini, pastikan “Jenis” yang dipilih merupakan kunci privat. Untuk mendapatkan kunci privat maka kita buka pada Ganache kita.



Setelah di buka, maka kita klik pada gambar kunci yang ada di setiap address yang ada dalam Ganache atau ikuti cara gambar di atas.



Setelah di klik, maka akan muncul Account Information. Copy Private Key yang ada didalamnya dan kembali ke MetaMask.

Impor Akun

Akun yang diimpor tidak akan dikaitkan dengan Frasa Pemulihan Rahasia akun MetaMask yang asli dibuat. Pelajari selengkapnya tentang akun yang diimpor [di sini](#)

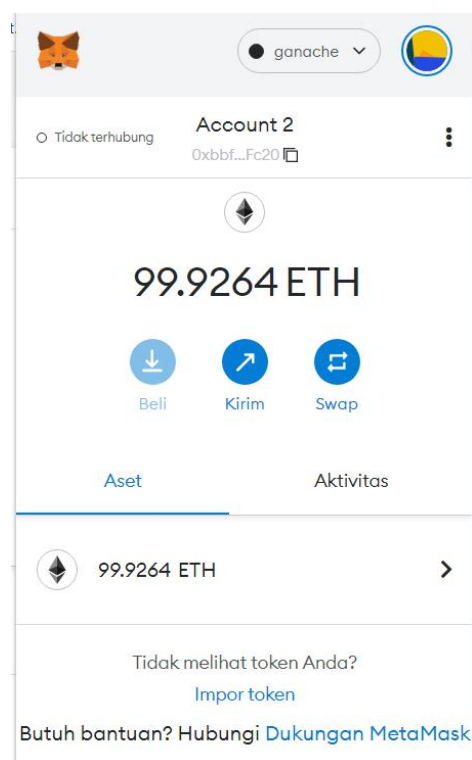
Pilih Jenis Kunci Privat

Tempel string kunci privat Anda di sini:

.....

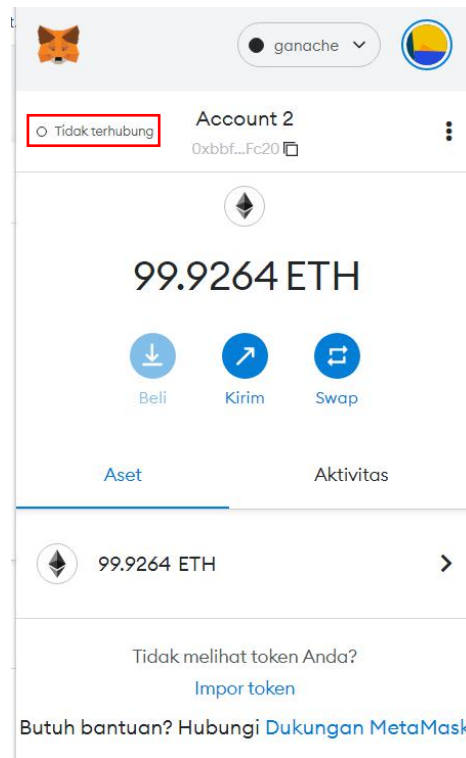
Batal Impor

Jika sudah di copy, masukkan Private Key yang didapatkan dari Ganache ke dalam kolom seperti gambar diatas. Apabila sudah selesai, klik “Impor” dan akun dari Ganache sudah berhasil di tambahkan.

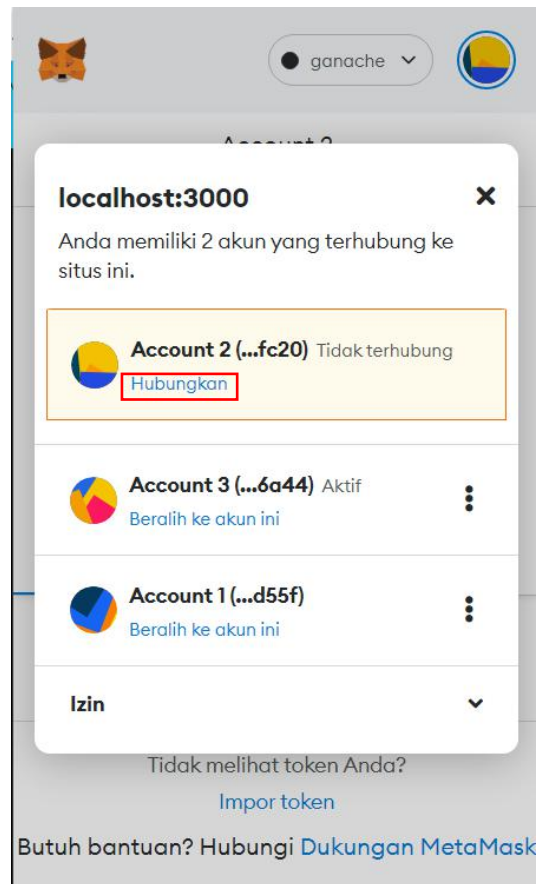


Cara berinteraksi dengan website yang dibuat:

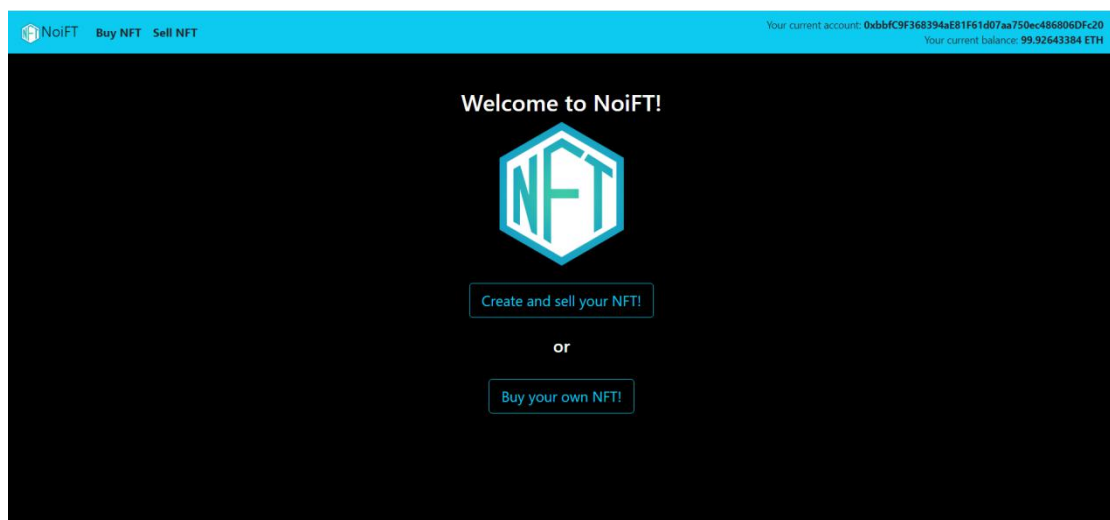
Sebelum memasuki cara berinteraksi dengan websitenya, pastikan sudah menjalankan beberapa perintah yang ada di atas dan sudah menambahkan jaringan baru serta akun dari Ganache. Akun yang dibutuhkan untuk di-impor dari Ganache minimal 2 akun.



Pertama pastikan akun sudah terhubung pada localhost:3000. Jika belum terhubung maka klik tulisan “Tidak terhubung” yang pada sebelah kiri address account atau ikuti seperti gambar di atas.



Jika sudah klik pada “Tidak terhubung” maka tampilan akan seperti berikut, agar akun terhubung lakukan klik “Hubungkan” atau seperti pada gambar di atas. Jika sudah di klik, maka akun tersebut sudah terhubung di localhost:3000.



Jika sudah terhubung, tungguilah sebentar atau refresh halaman sehingga tampilan akan seperti gambar diatas.

NoiFT

Buy NFT

Sell NFT

Your current account: 0xbbfC9F368394aE81F61d07aa750ec486806DFc20

Your current balance: 99.92643384 ETH

Mint and List your NFT

Product Name

Product Description

Product Price

Product Image URL

Submit

Your Products

#	Name	Description	Price	Image
You not have any products yet.				

Jika kita memilih button pertama yaitu “Create and sell your NFT!” maka tampilan akan seperti gambar diatas apabila belum ada barang NFT satupun yang ditambahkan.

NoiFT

Buy NFT

Sell NFT


Your current account: 0xbbfC9F368394aE81F61d07aa750ec486806DFc20

Your current balance: 99.92643384 ETH

Get your own NFT now!

No products for sale yet.

Jika kita memilih button kedua yaitu “Buy your own NFT!” maka tampilan akan seperti gambar diatas apabila belum ada barang NFT satupun yang dijual.


Buy NFT Sell NFT
Your current account: 0xbbfC9f368394aE81F61d07aa750ec486806DFc20
Your current balance: 99.92643384 ETH

Mint and List your NFT

Product Name

Product Description

Product Price

Product Image URL

Your Products

#	Name	Description	Price	Image
You not have any products yet.				


Pertama apabila kita ingin menjual sebuah gambar Icon yang merupakan Icon Video Game Royalty Free dengan harga 1 ETH dan link gambar seperti gambar diatas. Jika kita klik pada tombol Submit maka kita akan membayar gas pada MetaMask terlebih dahulu lalu data tersebut akan disimpan dalam blockchain.

Product Description

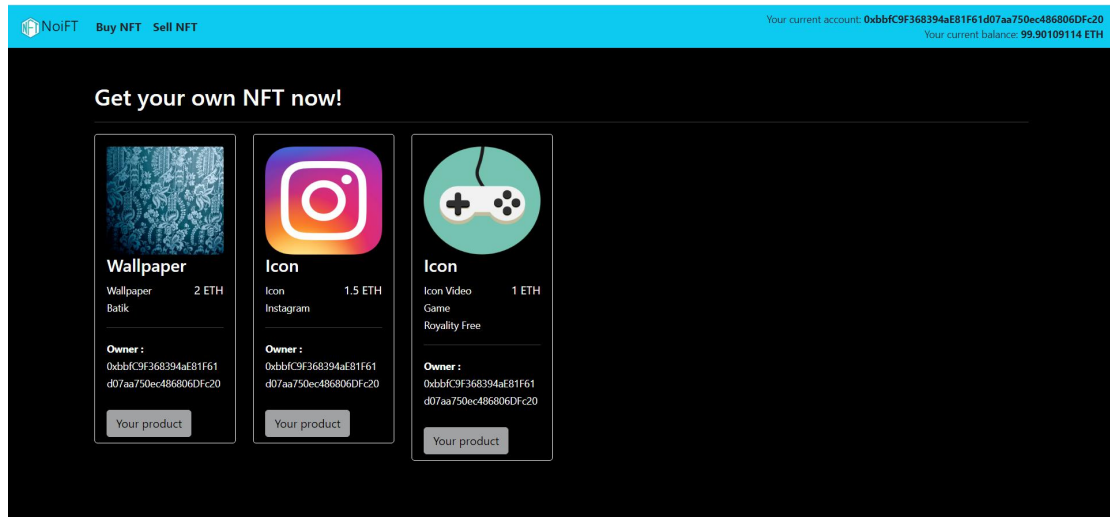
Product Price

Product Image URL

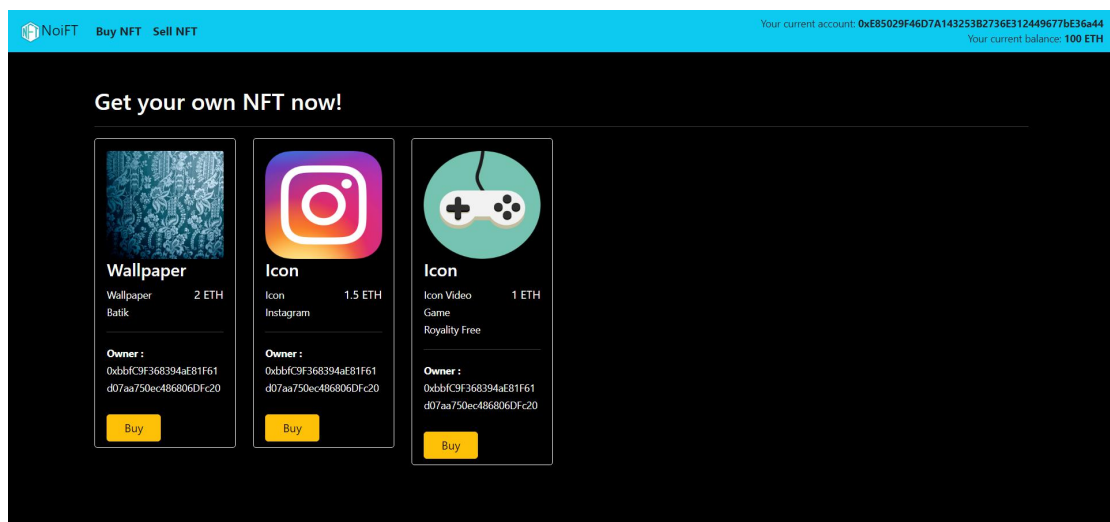
Your Products

#	Name	Description	Price	Image
1	Icon	Icon Video Game Royalty Free	1 ETH	

Dan tampilan pada Your Products akan menampilkan produk NFT yang baru saja ditambahkan seperti gambar diatas. Lalu kita coba masukkan beberapa produk NFT yang ingin di jual.



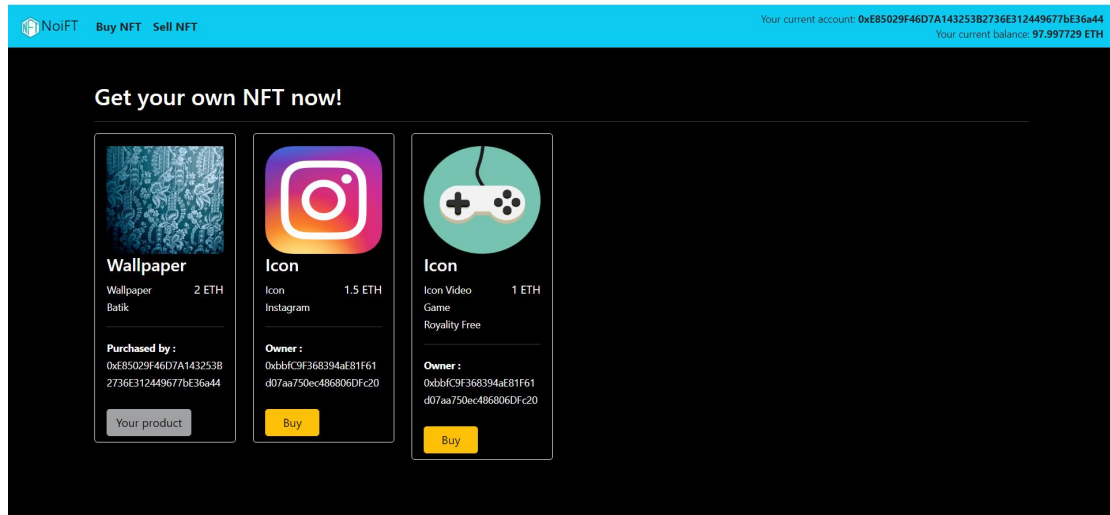
Jika kita sudah berhasil menambahkan beberapa produk NFT yang ingin dijual dan kita klik pada tulisan “Buy NFT” yang ada pada navbar diatas. Maka kita akan kembali ke halaman membeli NFT, jika kita merupakan owner atau pemiliki dari NFT tersebut maka kita tidak bisa membeli produk NFT kita sendiri.



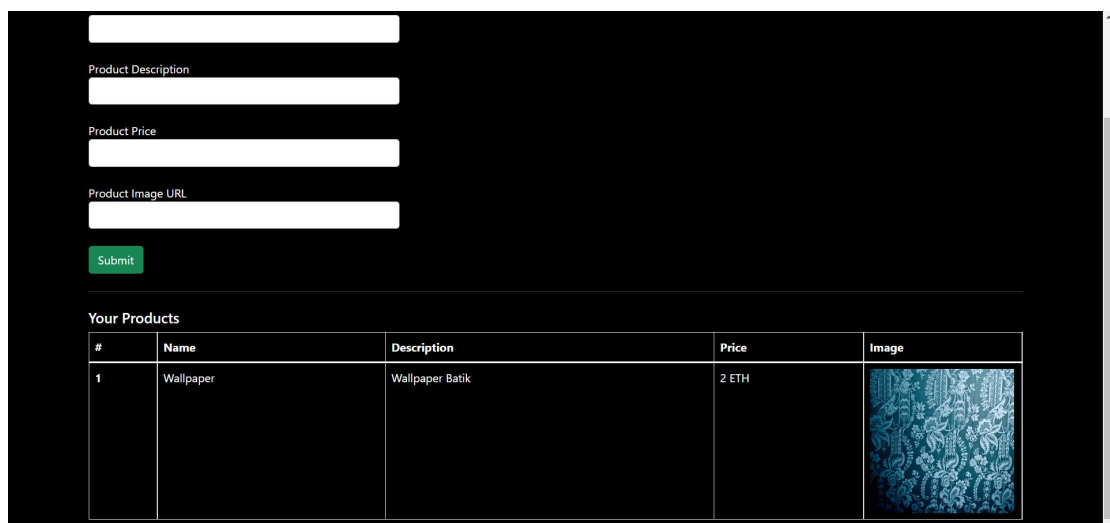
Apabila kita berada di akun pembeli, maka ada muncul tombol “Buy” dimana kita bisa membeli NFT tersebut. Pembeli ingin membeli NFT Wallpaper, maka kita klik pada tombol “Buy” yang ada pada Wallpaper.

#CATATAN :

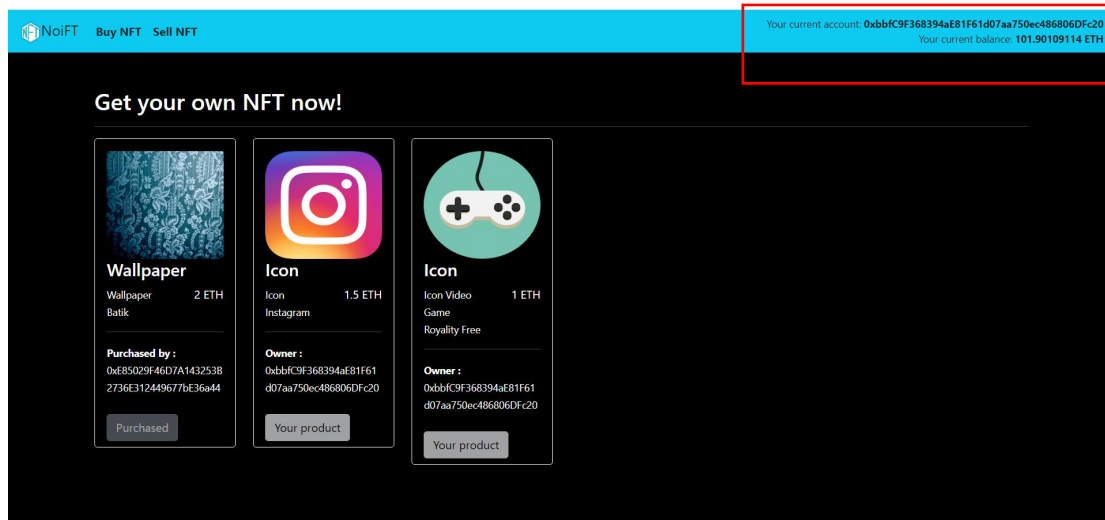
Pada saat berpindah akun, pastikan akun tersebut sudah terhubung pada localhost:3000 kemudian refresh halaman tersebut.



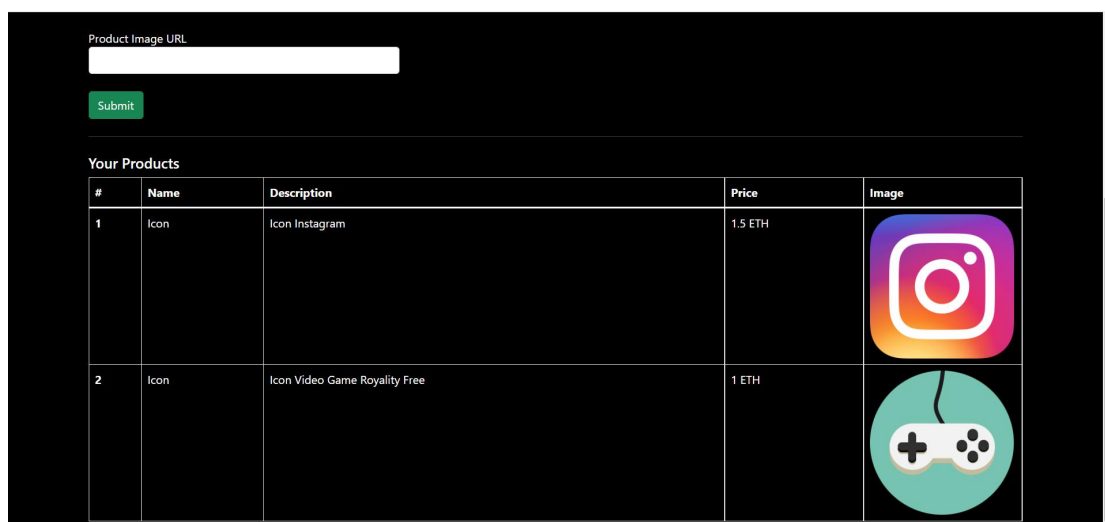
Pada saat proses membeli, pembeli harus membayar sedemikian harga gas serta harga barang tersebut. Jika pembeli sudah OK dengan harga tersebut maka klik Konfirmasi pada MetaMask lalu pembelian tersebut di proses dan kita akan dikembalikan pada halaman Buy NFT. Dan seperti gambar diatas, kita bisa melihat bahwa pembeli sudah berhasil membeli NFT Wallpaper dan NFT tersebut menjadi milik pembeli.



Jika kita kembali ke halaman Sell NFT, kita bisa lihat bahwa NFT Wallpaper sudah menjadi milik pembeli.



Apabila kita beralih ke akun penjual, maka pada halaman Buy NFT kita bisa melihat bahwa NFT Wallpaper sudah dibeli dan uang kita bertambah sebanyak 2 ETH atau kita bisa lihat pada navbar sebelah kanan.



Dan pada halaman Sell NFT, kita bisa melihat bahwa NFT Wallpaper sudah hilang karena sudah menjadi milik pembeli.