Pemanfaatan *Non Fungible Token* (NFT) pada media video sebagai alternatif *Digital Rights Management* (DRM) tradisional

Cathleen Lauretta 18221157
Program Studi Sistem dan Teknologi Informasi
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail (gmail): cathlauretta@gmail.com

***Abstract*—Jaman yang serba online membuat masyarakat cenderung lebih aktif dalam dunia virtual dengan menggunakan internet. Terjadi banyak sekali transaksi terkait pengiriman media digital. Adapun permasalahan muncul karena media digital tersebut cenderung sulit dilindungi hak ciptanya dan keamanannya. Oleh karena itu, digunakan *Digital Rights Management* (DRM) sebagai teknik untuk mengatasi permasalahan hak cipta, dimana DRM dapat membatasi akses terhadap sebuah media digital dan hanya memberikan otoritas kepada pihak yang berwenang. Namun, di sisi lain penggunaan DRM masih tergolong sederhana dan tidak bisa dimanfaatkan sepenuhnya karena terbatas pada perangkat dan penggunaan sumber daya komputasi yang rendah. Di sisi lain, perkembangan *Non Fungible Token* (NFT) menjadi semakin marak di kalangan *creator* digital. NFT menawarkan sistem transaksi yang cukup berbeda dari yang lain karena NFT bersifat unik dan setiap transaksinya tercatat dalam sistem *blockchain*. Pada penelitian kali ini, NFT akan digunakan sebagai penyedia akses bagi para pengguna yang ingin menggunakan layanan *streaming*.**

Keywords—Non Fungible Token, Blockchain, Streaming App, Media Video, Digital Rights Management

I. PENDAHULUAN

Seiring dengan perkembangan era digital di masa ini, hak cipta dan manajemen hak digital menjadi hal yang penting untuk diperhatikan bagi seseorang saat ingin melakukan distribusi konten digital di media internet. Beberapa perusahaan pada industri hiburan atau media kreatif dapat menggunakan *Digital Rights Management* (DRM) untuk melindungi hak cipta mereka dan mencegah pihak yang tidak berwenang untuk melakukan distribusi tanpa izin dari pemiliknya. Namun, cara tersebut masih tergolong sederhana dan memiliki keterbatasan saat ingin digunakan lintas *platform* [1].

Salah satu perusahaan yang menerapkan DRM adalah Vidio. Vidio menawarkan layanan *streaming* online kepada para penggunanya yang sudah berlangganan untuk menggunakan jasa mereka. Namun, terdapat beberapa perangkat yang tidak kompatibel untuk memutar konten yang dilindungi oleh DRM [1].

Sebagai alternatif dari penggunaan DRM, muncul *Non Fungible Token* (NFT) sebagai salah satu solusi dalam memperkuat hak cipta dan manajemen hak digital. NFT merupakan aset digital berbasis blockchain yang dapat mewakili projek tertentu [2]. NFT bersifat unik sehingga cocok untuk diimplementasi pada *platform* yang menawarkan produk terbatas kepada pelanggannya, seperti item koleksi, tiket lotre, atau tempat duduk bernomor pada suatu konser atau liga. Penggunaan NFT sendiri sudah memiliki standar tertentu, yaitu ERC-721, sehingga saat sebuah aplikasi ingin menggunakan NFT, NFT tersebut dapat digunakan secara umum.

Pada makalah ini, akan dijelaskan simulasi dari penerapan NFT pada sebuah aplikasi layanan *streaming* yang menyediakan konten video.

II. KAJIAN TEORI

A. *Digital Rights Management* (DRM)

DRM merupakan suatu teknik untuk mengelola perlindungan hak digital. DRM bekerja dengan cara mengontrol akses pengguna terhadap konten yang bisa mereka lihat serta melakukan autentikasi kepada pengguna yang ingin masuk ke dalam aplikasi. DRM juga dapat melacak histori dari suatu konten atau *file*, seperti melihat siapa saja yang sudah mengakses *file* tersebut, apa perubahan yang sudah terjadi pada *file* tersebut, dan kapan hal tersebut dilakukan oleh seorang pengguna. Apabila setelah diperiksa, ternyata pengguna bukanlah seseorang yang memiliki hak terhadap akses konten tersebut, DRM dapat mencabut akses pengguna kapan saja untuk memastikan integritas dari konten tersebut. [4]

B. *Blockchain*

Blockchain merupakan teknologi yang digunakan dalam penyimpanan data transaksi menggunakan teknik keamanan

kriptografi. Penyimpanan dilakukan dengan cara menyimpan data dalam bentuk blok (*block*), kemudian menghubungkannya menjadi satu rantai yang terhubung (*chain*) dimana isi atau nilainya tidak bisa diubah sembarangan. *Blockchain* menjadi suatu teknik yang penting karena *blockchain* dapat menciptakan sebuah buku besar yang konsisten dan disetujui oleh kedua pihak dalam bertransaksi. Salah satu pihak tidak bisa berbohong untuk mendapatkan keuntungan karena sudah ada sistem yang melakukan validasi terhadap transaksi yang sudah terjadi [6].

Komponen-komponen utama dari arsitektur *blockchain* adalah :

1. Kontrak

   Kontrak merupakan program yang disimpan di dalam *blockchain* untuk mengelola kontrak bisnis saat terdapat sebuah kondisi yang sudah terpenuhi. Implementasi dari kontrak dapat dilihat pada penggunaan NFT.

2. Kriptografi kunci-publik (asimetri)

   Kriptografi kunci-publik merupakan algoritma kriptografi yang menggunakan dua buah kunci dalam melakukan enkripsi dan dekripsi, selanjutnya disebut sebagai kunci publik dan kunci privat. Nantinya, kunci publik akan disebarkan ke umum untuk diketahui oleh semua orang, sedangkan kunci privat akan disimpan oleh pemilik kunci sehingga hanya diketahui oleh pemilik tersebut saja.

C. *Non Fungible Token* (NFT)

Seperti namanya, NFT merupakan sebuah token digital yang nilainya tidak dapat digantikan atau ditukarkan. NFT biasanya digunakan sebagai aset digital yang mewakili suatu barang berharga sehingga koleksi NFT bisa mencapai harga jutaan dollar Amerika Serikat. Setiap NFT memiliki catatan transaksi yang tersimpan di dalam *blockchain*. Biasanya terdiri dari nama penciptanya, harga, serta histori kepemilikannya (sudah dipindah tangan berapa kali dan ke siapa saja).

Mengapa NFT tidak bisa dipertukarkan? Masing-masing NFT hanya diciptakan sekali (seperti ibu jari yang unik pada setiap manusia) sehingga nilai dari satu NFT dengan NFT lain belum tentu sama (meskipun mungkin nama dan aset digital yang di-*sharing* itu sama). Sifat NFT yang unik menjadikan NFT sebagai alat yang bersifat mutlak, artinya siapapun pemilik sebuah NFT memiliki hak penuh terhadap hak cipta dari aset digital asli yang dipegang oleh NFT tersebut. Oleh karena itu, NFT disukai oleh *creator* karena NFT menjadi solusi terbaru dalam penjualan karya tanpa melalui pihak perantara. [2]

III. Implementasi

Aplikasi simulasi layanan *streaming* ini akan dibuat menggunakan beberapa komponen pembangun yang mendukung teknologi *blockchain* dan NFT. Berikut link github dari implementasi tersebut : https://github.com/cathlauretta/II4031-Newflix

A. Truffle dan OpenZeppelin

Truffle merupakan sebuah *framework* yang didasarkan pada Ethereum Blockchain [5]. Framework ini digunakan untuk membantu pengembangan dan *deployment* dari *smart contract* NFT. Truffle dapat digunakan dengan menginstal Truffle Ethereum pada *Node Package Manager* (npm).

```
> npm install -g truffle
```

Setelah melakukan instalasi Truffle, selanjutnya Anda harus menginisialisasi Truffle dan *library* OpenZeppelin di dalam folder yang sudah tercipta dari instalasi Truffle. OpenZeppelin merupakan *platform open-source* yang berisi kumpulan *library* untuk *smart contract*.

```
> truffle init

> npm install @openzeppelin/contracts
```

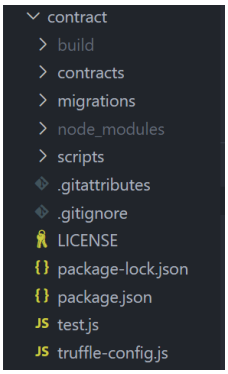

Fig. 1. Struktur folder dari kontrak truffle

Selanjutnya, akan dibuat kontrak NFT dengan standar ERC-721 [3] di dalam folder `contracts`. Berikut merupakan kode program dari `NFTCollection.sol`

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.13;

import "@openzeppelin/contracts/access/Ownable.sol";
import "@openzeppelin/contracts/utils/math/Math.sol";
import
"@openzeppelin/contracts/token/ERC721/extensions/ERC721En
umerable.sol";

contract NFTCollection is ERC721Enumerable, Ownable {
    using Math for uint256;

    uint private _tokenIds;
    // The max number of NFTs in the collection
    uint public constant MAX_SUPPLY = 1000;
    // The mint price for the collection
    uint public constant PRICE = 0 ether;
    // The max number of mints per wallet
    uint public constant MAX_PER_MINT = 10;

    string public baseTokenURI;

    constructor(string memory baseURI, string memory
name, string memory symbol, address owner) ERC721(name,
symbol) Ownable(owner) {
        setBaseURI(baseURI);
    }
```

```
    function _baseURI() internal view virtual override
returns (string memory) {
        return baseTokenURI;
    }

    function setBaseURI(string memory _baseTokenURI)
public onlyOwner {
        baseTokenURI = _baseTokenURI;
    }

    function mintNFTs(uint _count) public payable {
        uint totalMinted = _tokenIds;

        require(totalMinted + _count <= MAX_SUPPLY, "This
collection is sold out!");
        require(_count > 0 && _count <= MAX_PER_MINT,
"You have received the maximum amount of NFTs allowed.");
        require(msg.value >= PRICE * _count, "Not enough
ether to purchase NFTs.");

        for (uint i = 0; i < _count; i++) {
            _mintSingleNFT();
        }
    }

    function mintForAddress(address recipient, uint
_count) public onlyOwner {
        uint totalMinted = _tokenIds;
        require(totalMinted + _count <= MAX_SUPPLY, "Not
enough NFTs left to mint");

        for (uint i = 0; i < _count; i++) {
            _mintSingleNFT(recipient);
        }
    }

    function _mintSingleNFT() private {
        uint newTokenID = _tokenIds;
        _safeMint(msg.sender, newTokenID);
        _tokenIds++;
    }

    function _mintSingleNFT(address recipient) private {
        uint newTokenID = _tokenIds;
        _safeMint(recipient, newTokenID);
        _tokenIds++;
    }

    // Returns the ids of the NFTs owned by the wallet
address
    function tokensOfOwner(address _owner) external view
returns (uint[] memory) {
        uint tokenCount = balanceOf(_owner);
        uint[] memory tokensId = new
uint256[](tokenCount);

        for (uint i = 0; i < tokenCount; i++) {
            tokensId[i] = tokenOfOwnerByIndex(_owner, i);
        }
        return tokensId;
    }

    // Withdraw the ether in the contract
    function withdraw() public payable onlyOwner {
        uint balance = address(this).balance;
        require(balance > 0, "No ether left to
withdraw");

        (bool success, ) = (msg.sender).call{value:
balance}("");
        require(success, "Transfer failed.");
    }

    // Reserve NFTs only for owner to mint for free
    function reserveNFTs(uint _count) public onlyOwner {
        uint totalMinted = _tokenIds;

        require(totalMinted + _count < MAX_SUPPLY, "Not
enough NFTs left to reserve");

        for (uint i = 0; i < _count; i++) {
            _mintSingleNFT();
        }
    }
}
```

B. Next.js

Next.js merupakan *framework* berbasis React yang digunakan untuk mempermudah pembuatan *website* yang siap diproduksi. Next.js akan digunakan dalam implementasi antarmuka dari layanan *streaming*.

Simulasi layanan *streaming* akan dilakukan dengan scenario berikut.

1. Terdapat dua akun yang masing-masing memiliki NFT.
2. Akun pertama akan memiliki seluruh token yang terdiri dari token [0, 1, 2, 3, 4, 5, 6, 7] sedangkan akun kedua belum memiliki token apapun.
3. Masing-masing token memiliki akses terhadap 3 video yang berbeda-beda.
4. Akun kedua belum memiliki token sama sekali sehingga akun pertama akan melakukan transfer token ke akun kedua supaya akun kedua memiliki token untuk mengakses video.

NFT yang digunakan pada simulasi kali ini belum pernah ada sebelumnya sehingga diperlukan kode untuk generate token baru. Berikut merupakan kode pembuatan NFT untuk digunakan pada simulasi ini.

```
const { Web3 } = require('web3');

// Initialize Web3
const web3 = new Web3("https://eth-
sepolia.g.alchemy.com/v2/mlbXSJTgFXSeZvKvVnsaTIyPACpFlxhR
");
// Define the function to mint tokens
const mintTokens = async (recipient, count) => {
    const abi = [
        {
            "inputs": [
                {
                    "internalType": "address",
                    "name": "recipient",
                    "type": "address"
                },
                {
                    "internalType": "uint256",
                    "name": "_count",
                    "type": "uint256"
```

```
                }
            ],
            "name": "mintForAddress",
            "outputs": [],
            "stateMutability": "nonpayable",
            "type": "function"
        }
    ];
    try {
        const MetaprivateKey =
'0xe7e575caf52cd9399e08338f78d503e6f030a4e04e95af0bbe9b4e
19347eb4f5';
        const deployedAddress =
"0xb9E1D0D59D24a2f202D7D54cf79fba47F21a8732";
        const myContract = new web3.eth.Contract(abi,
deployedAddress);
        myContract.handleRevert = true;

        // Get account from the private key
        const account =
web3.eth.accounts.wallet.add(MetaprivateKey);
        const usedAccount = account[0];

        // Interact with the smart contract
        const receipt = await
myContract.methods.mintForAddress(recipient,
count).send({
            from: usedAccount.address,
            gas: "500000",
            gasPrice: "100000000000",
        });

        // Return a successful response
        return { success: true, transactionHash:
receipt.transactionHash };

    } catch (error) {
        console.error("Error:", error);
        return { success: false, message: "Internal
Server Error" };
    }
};

// Test the function
const testMint = async () => {
    const recipient =
"0xf1a5cba704Be91332fFc26dfD8b87E72Cb5514ee"; // Replace
with a valid recipient address
    const count = 1;

    const result = await mintTokens(recipient, count);
    console.log(result);
    process.exit(); // Add this line to exit the script
};

testMint();

module.exports = { mintTokens, web3};
```

Berikut merupakan skenario pengiriman token NFT dari satu akun ke akun lainnya. Untuk memperoleh *address* dan *private key* dari pengguna NFT, akan digunakan MetaMask, yaitu aplikasi *crypto wallet* yang terhubung dengan *web browser* (berupa *extension* dari Google Chrome).

```
const { Web3 } = require('web3');

// Initialize Web3
const web3 = new Web3("https://eth-
sepolia.g.alchemy.com/v2/mlbXSJTgFXSeZvKvVnsaTIyPACpFlxhR
");

// Define the function to send NFT
const sendNFT = async (from, to, tokenId) => {
    const abi = [
        {
            "inputs": [
                {
                    "internalType": "address",
                    "name": "from",
                    "type": "address"
                },
                {
                    "internalType": "address",
                    "name": "to",
                    "type": "address"
                },
                {
                    "internalType": "uint256",
                    "name": "tokenId",
                    "type": "uint256"
                }
            ],
            "name": "safeTransferFrom",
            "outputs": [],
            "stateMutability": "nonpayable",
            "type": "function"
        }
    ];

    try {
        const privateKey =
'0x1cb43f07a510e0533cb468f11cdbc21434615d79c0fc79d72517a9
587f75e531';
        const deployedAddress =
"0xb9E1D0D59D24a2f202D7D54cf79fba47F21a8732"; // Replace
with your contract address
        const myContract = new web3.eth.Contract(abi,
deployedAddress);
        myContract.handleRevert = true;

        // Get account from the private key
        const account =
web3.eth.accounts.wallet.add(privateKey);
        const usedAccount = account[0];

        // Interact with the smart contract
        const receipt = await
myContract.methods.safeTransferFrom(from, to,
tokenId).send({
            from: usedAccount.address,
            gas: "500000",
            gasPrice: "100000000000",
        });

        // Return a successful response
        return { success: true, transactionHash:
receipt.transactionHash };

    } catch (error) {
        console.error("Error:", error);
        return { success: false, message: "Internal
Server Error" };
    }
};

// Test the function
const testSendNFT = async () => {
```

```
    const from =
"0xf1a5cba704Be91332fFc26dfD8b87E72Cb5514ee"; // Replace
with the sender's address
    const to =
"0x6eC9669aAb5e98F96FAAA7dC6bA716b190Ef0aa6"; // Replace
with the recipient's address
    const tokenId = 1; // Replace with the token ID you
want to transfer

    const result = await sendNFT(from, to, tokenId);
    console.log(result);
    process.exit(); // Add this line to exit the script
};

testSendNFT();

module.exports = { sendNFT, web3 };
```

Setelah seluruh rangkaian skenario berhasil dijalankan, hasilnya kurang lebih akan menjadi seperti ini.

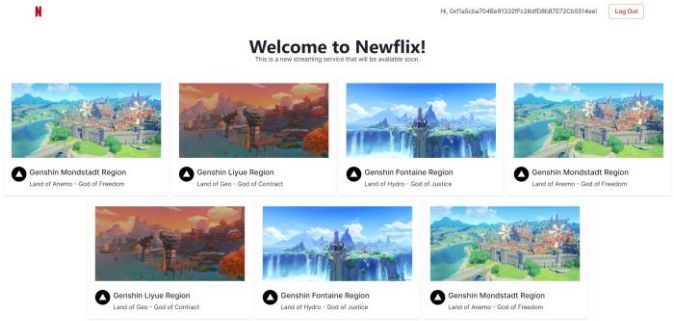

Fig. 2. Antarmuka dari akun pengguna pertama

Gambar di atas merupakan tampilan saat pengguna pertama melakukan login ke dalam aplikasi layanan *streaming*. Dapat dilihat, seharusnya pengguna tersebut memiliki akses kepada 8 video. Namun karena satu NFT sudah diberikan ke akun lain, maka pengguna pertama sudah kehilangan akses terhadap satu video tersebut.

Fig. 3. Antarmuka dari akun pengguna kedua

Gambar di atas merupakan tampilan saat pengguna kedua melakukan login ke dalam aplikasi layanan *streaming*. Karena pengguna sudah menerima 1 NFT dari pengguna pertama, maka saat ini pengguna kedua sudah memiliki akses terhadap 1 video yang sesuai dengan NFT tersebut.

IV. Penutup

Simulasi dari pemanfaatan NFT pada media video dapat mengatasi permasalahan yang berkaitan dengan hak cipta dan manajemen hak digital. NFT dapat dijadikan alternatif dari penggunaan DRM tradisional yang pada beberapa kesempatan terkadang mengalami masalah terkait keamanan. Dengan penggunaan NFT sebagai aset digital yang unik pada setiap video, hal ini dapat memperkuat hak cipta dari *creator* masing-masing konten dan mengelola manajemen hak digital dengan lebih efektif. Namun, karena penelitian kali ini masih bersifat simulasi dan belum diterapkan secara penuh, masih diperlukan penelitian lebih lanjut untuk memahami implementasi dari NFT itu sendiri dalam perindustrian yang ada di dunia realita.

Daftar Pustaka

[1] Kumparan. (2022, November 22). APA ITU DRM (Digital Rights Management)? Ini Pengertian Dan Contohnya. kumparan. https://kumparan.com/berita-hari-ini/apa-itu-drm-digital-rights-management-ini-pengertian-dan-contohnya-1zIEe9wICLA/full

[2] Pintu. (2024, June 10). Apa Itu Nft? Ghozali Dapat Miliaran Darisini, Kamu Selanjutnya?. Apa Itu NFT? https://pintu.co.id/academy/post/nft-adalah

[3] Ethereum. (n.d.). ERC-721 non-fungible token standard. ethereum.org. https://ethereum.org/en/developers/docs/standards/tokens/erc-721/

[4] Phintraco. (2022, June 2). Digital Rights Management (DRM) Lindungi Penyebaran Data Digital. Phintraco Group. https://phintraco.com/digital-rights-management-drm-lindungi-penyebaran-data-digital/ J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68-73.

[5] Priyandani, D. (2020, October 8). APA ITU Framework Truffle Suite di Dalam Blockchain ?. Cryptoiz Research. https://news.cryptoizresearch.com/apa-itu-framework-truffle-suite-di-dalam-blockchain/

[6] AWS. (n.d.). What is blockchain? - blockchain technology explained - AWS. What is blockchain? https://aws.amazon.com/what-is/blockchain/

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 12 Juni 2024

Cathleen Lauretta 18221157