

**TUGAS AKHIR - EC234801**

# **INTEROPERABILITAS NFT BERBASIS BLOCKCHAIN MENGUNAKAN SMART CONTRACT PADA WEB3.0**

**Arya Abdul Azis**

NRP 5024201069

Dosen Pembimbing

**Mochamad Hariadi, S.T., M.Sc., Ph.D**

NIP 199691209199703 1 002

**Reza Fuad Rachmadi, S.T., M.T., Ph.D**

NIP 19850403201212 1 001

**Program Studi Strata 1 (S1) Teknik Komputer**

Departemen Teknik Komputer

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Surabaya

2024



**TUGAS AKHIR - EC234801**

# **INTEROPERABILITAS NFT BERBASIS BLOCKCHAIN MENGUNAKAN SMART CONTRACT PADA WEB3.0**

**Arya Abdul Azis**

NRP 5024201069

Dosen Pembimbing

**Mochamad Hariadi, S.T., M.Sc., Ph.D**

NIP 199691209199703 1 002

**Reza Fuad Rachmadi, S.T., M.T., Ph.D**

NIP 19850403201212 1 001

**Program Studi Strata 1 (S1) Teknik Komputer**

Departemen Teknik Komputer

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Surabaya

2024

*[Halaman ini sengaja dikosongkan]*



**FINAL PROJECT - EC234801**

***BLOCKCHAIN-BASED NFT INTEROPERABILITY  
USING SMART CONTRACTS IN WEB3.0***

**Arya Abdul Azis**

NRP 5024201069

Advisor

**Mochamad Hariadi, S.T., M.Sc., Ph.D**

NIP 199691209199703 1 002

**Reza Fuad Rachmadi, S.T., M.T., Ph.D**

NIP 19850403201212 1 001

**Undergraduate Study Program of Computer Engineering**

Department of Computer Engineering

Faculty of Intelligent Electrical and Informatics Technology

Sepuluh Nopember Institute of Technology

Surabaya

2024

*[Halaman ini sengaja dikosongkan]*

## LEMBAR PENGESAHAN

### INTEROPERABILITAS NFT BERBASIS BLOCKCHAIN MENGGUNAKAN SMART CONTRACT PADA WEB3.0

#### TUGAS AKHIR

Diajukan untuk memenuhi salah satu syarat  
memperoleh gelar Sarjana Teknik pada  
Program Studi S-1 Teknik Komputer  
Departemen Teknik Komputer  
Fakultas Teknologi Elektro dan Informatika Cerdas  
Institut Teknologi Sepuluh Nopember

Oleh: **Arya Abdul Azis**  
NRP. 5024201069

Disetujui oleh Tim Penguji Tugas Akhir:

Mochamad Hariadi, S.T., M.Sc., Ph.D  
NIP: 199691209199703 1 002

(Pembimbing I)

.....

Reza Fuad Rachmadi, S.T., M.T., Ph.D  
NIP: 19850403201212 1 001

(Pembimbing II)

.....

Dr. Galileo Galilei, S.T., M.Sc.  
NIP: 18560710 194301 1 001

(Penguji I)

.....

Friedrich Nietzsche, S.T., M.Sc.  
NIP: 18560710 194301 1 001

(Penguji II)

.....

Alan Turing, ST., MT.  
NIP: 18560710 194301 1 001

(Penguji III)

.....

Mengetahui,  
Kepala Departemen Teknik Komputer FTEIC - ITS

Dr. Supeno Mardi Susiki Nugroho, S.T., M.T..  
NIP. 19700313199512 1 001

**SURABAYA**  
**Juni, 2024**

*[Halaman ini sengaja dikosongkan]*

# APPROVAL SHEET

## ***BLOCKCHAIN-BASED NFT INTEROPERABILITY USING SMART CONTRACTS IN WEB3.0***

### **FINAL PROJECT**

Submitted to fulfill one of the requirements  
for obtaining a degree Bachelor of Engineering at  
Undergraduate Study Program of Computer Engineering  
Department of Computer Engineering  
Faculty of Intelligent Electrical and Informatics Technology  
Sepuluh Nopember Institute of Technology

By: **Arya Abdul Azis**  
NRP. 5024201069

Approved by Final Project Examiner Team:

Mochamad Hariadi, S.T., M.Sc., Ph.D  
NIP: 199691209199703 1 002

(Advisor I)

.....

Reza Fuad Rachmadi, S.T., M.T., Ph.D  
NIP: 19850403201212 1 001

(Co-Advisor II)

.....

Dr. Galileo Galilei, S.T., M.Sc.  
NIP: 18560710 194301 1 001

(Examiner I)

.....

Friedrich Nietzsche, S.T., M.Sc.  
NIP: 18560710 194301 1 001

(Examiner II)

.....

Alan Turing, ST., MT.  
NIP: 18560710 194301 1 001

(Examiner III)

.....

Acknowledged,  
Head of Computer Engineering Department FTEIC - ITS

Dr. Supeno Mardi Susiki Nugroho, S.T., M.T..  
NIP. 19700313199512 1 001

**SURABAYA**  
**June, 2024**



*[Halaman ini sengaja dikosongkan]*

## PERNYATAAN ORISINALITAS

Yang bertanda tangan dibawah ini:

Nama Mahasiswa / NRP : Arya Abdul Azis / 5024201069  
Departemen : Teknik Komputer  
Dosen Pembimbing / NIP : Mochamad Hariadi, S.T., M.Sc., Ph.D / 199691209199703 1  
002

Dengan ini menyatakan bahwa Tugas Akhir dengan judul "INTEROPERABILITAS NFT BERBASIS BLOCKCHAIN MENGGUNAKAN SMART CONTRACT PADA WEB3.0" adalah hasil karya sendiri, bersifat orisinal, dan ditulis dengan mengikuti kaidah penulisan ilmiah.

Bilamana di kemudian hari ditemukan ketidaksesuaian dengan pernyataan ini, maka saya bersedia menerima sanksi sesuai dengan ketentuan yang berlaku di Institut Teknologi Sepuluh Nopember.

Surabaya, June 2024

Mengetahui  
Dosen Pembimbing

Mahasiswa

Mochamad Hariadi, S.T., M.Sc., Ph.D  
NIP. 199691209199703 1 002

Arya Abdul Azis  
NRP. 5024201069

*[Halaman ini sengaja dikosongkan]*

## STATEMENT OF ORIGINALITY

The undersigned below:

Name of student / NRP : Arya Abdul Azis / 5024201069  
Department : Computer Engineering  
Advisor / NIP : Mochamad Hariadi, S.T., M.Sc., Ph.D / 199691209199703 1 002

Hereby declared that the Final Project with the title of "*BLOCKCHAIN-BASED NFT INTEROPERABILITY USING SMART CONTRACTS IN WEB3.0*" is the result of my own work, is original, and is written by following the rules of scientific writing.

If in future there is a discrepancy with this statement, then I am willing to accept sanctions in accordance with provisions that apply at Sepuluh Nopember Institute of Technology.

Surabaya, June 2024

Acknowledged  
Advisor

Student

Mochamad Hariadi, S.T., M.Sc., Ph.D  
NIP. 199691209199703 1 002

Arya Abdul Azis  
NRP. 5024201069

*[Halaman ini sengaja dikosongkan]*

## ABSTRAK

Nama Mahasiswa : Arya Abdul Azis  
Judul Tugas Akhir : INTEROPERABILITAS NFT BERBASIS BLOCKCHAIN MEN-  
GUNAKAN SMART CONTRACT PADA WEB3.0  
Pembimbing : 1. Mochamad Hariadi, S.T., M.Sc., Ph.D  
2. Reza Fuad Rachmadi, S.T., M.T., Ph.D

Pada penelitian ini kami mengajukan Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Kata Kunci: Roket, *Anti-gravitasi*, Energi, Angkasa.

*[Halaman ini sengaja dikosongkan]*

## ABSTRACT

*Name* : Arya Abdul Azis  
*Title* : *BLOCKCHAIN-BASED NFT INTEROPERABILITY USING SMART CONTRACTS IN WEB3.0*  
*Advisors* : 1. Mochamad Hariadi, S.T., M.Sc., Ph.D  
2. Reza Fuad Rachmadi, S.T., M.T., Ph.D

*In this research, we proposed Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.*

*Keywords: Rocket, Anti-gravity, Energy, Space.*



*[Halaman ini sengaja dikosongkan]*

# KATA PENGANTAR

Puji dan syukur kehadiran Allah SWT. yang telah melimpahkan rahmat dan hidayah-Nya sehingga penyusun dapat menyelesaikan laporan skripsi berjudul Interoperabilitas NFT Berbasis Blockchain Menggunakan Smart Contract Pada Metaverse dan Web3.0.

Penelitian ini disusun dalam rangka sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik Komputer Program Studi Teknik Komputer Institut Teknologi Sepuluh Nopember. Laporan ini berisi hasil penelitian yang telah dilakukan oleh penyusun selama kurang lebih enam bulan. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Keluarga, Ibu, Bapak dan Saudara tercinta yang telah memberikan dukungan dalam penyelesaian tugas akhir ini.
2. Bapak Mochamad Hariadi, S.T., M.Sc., Ph.D, dan bapak Reza Fuad Rachmadi, S.T., M.T., Ph.D selaku dosen pembimbing satu dan kedua yang telah membimbing saya dalam menyelesaikan tugas akhir ini.
3. Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl.

Akhir kata, penyusun menyadari bahwa laporan skripsi ini masih jauh dari kata sempurna. Dengan demikian, penyusun mengharapkan saran dan kritik yang membangun dari pembaca. Semoga laporan ini dapat bermanfaat bagi para pendidik dan mahasiswa dalam meningkatkan literasi dan pengetahuan mengenai Smart Contract, Blockchain, serta NFT.

Surabaya, Juni 2024

Arya Abdul Azis

*[Halaman ini sengaja dikosongkan]*

# DAFTAR ISI

<b>ABSTRAK</b>	<b>i</b>
<b>ABSTRACT</b>	<b>iii</b>
<b>KATA PENGANTAR</b>	<b>v</b>
<b>DAFTAR ISI</b>	<b>vii</b>
<b>DAFTAR GAMBAR</b>	<b>xi</b>
<b>DAFTAR TABEL</b>	<b>xiii</b>
<b>1 PENDAHULUAN</b>	<b>1</b>
1.1 Latar Belakang . . . . .	1
1.2 Permasalahan . . . . .	2
1.3 Batasan Masalah atau Ruang Lingkup . . . . .	2
1.4 Tujuan . . . . .	2
1.5 Batasan Masalah . . . . .	2
1.6 Sistematika Penulisan . . . . .	2
<b>2 TINJAUAN PUSTAKA</b>	<b>5</b>
2.0.1 Non-Fungible Token (NFT): Overview, Evaluation, Opportunities and Challenges . . . . .	5
2.0.2 Blockchain technology for creative industries: Current state and research opportunities . . . . .	5
2.0.3 An Overview on Smart Contracts: Challenges, Advances and Platforms . . . . .	5
2.0.4 Blockchain for the metaverse: A Review . . . . .	6
2.1 Dasar Teori . . . . .	6
2.1.1 <i>Blockchain</i> . . . . .	6
2.1.2 <i>Ethereum</i> . . . . .	6
2.1.3 <i>Ethereum Virtual Machine (EVM)</i> . . . . .	7
2.1.4 <i>Token</i> . . . . .	7
2.1.5 <i>Non-Fungible Token (NFT)</i> . . . . .	7
2.1.6 <i>Minting</i> . . . . .	8

2.1.7	<i>Smart Contracts</i>	8
2.1.8	<i>InterPlanetary File System (IPFS)</i>	9
2.1.9	Web 3.0	9
2.1.10	<i>Cross-chain Interoperability Protocol</i>	10
<b>3</b>	<b>METODOLOGI</b>	<b>11</b>
3.1	Deskripsi Sistem	11
3.1.1	Arsitektur Sistem	11
3.1.2	Flow Pembelian NFT	12
3.1.3	Metadata	12
3.1.4	Minting	13
3.2	Tools yang digunakan	13
3.2.1	Solidity	13
3.2.2	Metamask	13
3.2.3	Hardhat	14
3.2.4	Ganache	14
3.2.5	InterPlanetary File System	14
3.2.6	React.js	14
3.2.7	Web3.js	14
3.2.8	Ethers.js	15
3.3	Metode Yang Digunakan	15
3.3.1	Smart Contract dan Interoperabilitas	15
3.3.2	Pengembangan Frontend	19
3.3.3	Integrasi Smart Contract dengan Frontend	19
3.3.4	Pengujian	20
<b>4</b>	<b>HASIL DAN PEMBAHASAN</b>	<b>21</b>
4.1	Hasil	21
4.1.1	Web Interface	21
4.1.2	<i>Smart Contract</i>	23
4.1.3	<i>Non-Fungible Token (NFT)</i>	24
4.2	Pengujian	25
4.2.1	Pengujian Fitur <i>Smart Contract</i>	25
4.2.2	Pengujian Integrasi <i>Smart Contract</i> Dengan Web3.0	30
<b>5</b>	<b>PENUTUP</b>	<b>41</b>

5.1	Kesimpulan . . . . .	41
5.2	Saran . . . . .	41
<b>DAFTAR PUSTAKA</b>		<b>43</b>
<b>BIOGRAFI PENULIS</b>		<b>45</b>

*[Halaman ini sengaja dikosongkan]*

## DAFTAR GAMBAR

3.1	Arsitektur Sistem . . . . .	11
3.2	Flow Pembelian NFT . . . . .	12
3.3	Proses Minting . . . . .	13
3.4	Metodologi Penelitian Yang Digunakan . . . . .	15
4.1	Tampilan interface . . . . .	21
4.2	Koneksi <i>interface</i> dengan Metamask Wallet . . . . .	22
4.3	<i>Payment</i> dan <i>address</i> . . . . .	22
4.4	Detail <i>payment</i> menggunakan Metamask Wallet . . . . .	23
4.5	<i>address</i> pemilik baru dari NFT . . . . .	23
4.6	Detail transaksi pada etherscan . . . . .	24
4.7	NFT terunggah pada Opensea . . . . .	25
4.8	Riwayat transaksi minting pada Metamask Wallet . . . . .	26
4.9	Detail transaksi pada Etherscan . . . . .	26
4.10	Detail NFT pada Etherscan . . . . .	27
4.11	NFT pada OpenSea . . . . .	27
4.12	Detail fungsi <i>Lock Token</i> pada Etherscan . . . . .	28
4.13	Parameter <i>bridge transfer</i> . . . . .	28
4.14	Pembayaran pada Metamask Wallet . . . . .	29
4.15	Detail pada Etherscan . . . . .	29
4.16	Detail NFT pada Etherscan setelah <i>bridge transfer</i> . . . . .	30
4.17	NFT pada OpenSea <i>address B</i> . . . . .	30
4.18	Tampilan awal dari web . . . . .	31
4.19	Koneksi Metamask dengan Web . . . . .	32
4.20	Tampilan awal dari web . . . . .	32
4.21	Tampilan halaman <i>create</i> . . . . .	33
4.22	Tampilan halaman <i>my listed items</i> . . . . .	33
4.23	Tampilan halaman <i>my purchases</i> . . . . .	34
4.24	Melakukan pengunggahan NFT pada halaman <i>create</i> . . . . .	35
4.25	Pembayaran gas IPFS menggunakan Metamask Wallet . . . . .	35
4.26	Data dari NFT ter- <i>upload</i> pada platform Pinata . . . . .	36



4.27 Melakukan konfirmasi <i>minting</i> NFT pada Metamask Wallet . . . . .	37
4.28 NFT yang telah di- <i>minting</i> terlihat pada halaman <i>My listed items</i> . . . . .	38
4.29 NFT yang telah di- <i>minting</i> terlihat pada halaman <i>home</i> . . . . .	38

## **DAFTAR TABEL**

4.1	Informasi Akun . . . . .	25
-----	--------------------------	----

*[Halaman ini sengaja dikosongkan]*

# BAB I

## PENDAHULUAN

Penelitian ini dilatarbelakangi oleh Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque.

### 1.1 Latar Belakang

Dalam beberapa tahun terakhir, NFT telah menjadi sorotan utama bagi kalangan industri dan akademisi. Data menunjukkan bahwa dalam sehari, volume transaksi pasar NFT mencapai sekitar 4,592 miliar USD, sedangkan total transaksi pasar kripto dalam sehari mencapai sekitar 341,017 miliar USD (Wang et al., 2021). Non-Fungible Tokens (NFTs) adalah aset digital yang mewakili objek seperti seni, barang koleksi, dan item dalam game. Aset ini diperjualbelikan di internet, kebanyakan dengan mata uang kripto, dan umumnya tertanam dalam *smart contract* di sebuah *blockchain*. NFT juga bersifat khas sehingga tidak dapat dipertukarkan dengan objek serupa, hal ini menjadikannya tepat untuk mengenal pasti sesuatu atau seseorang dengan cara unik. Walaupun NFT menjanjikan dampak signifikan pada pasar yang terdesentralisasi sekarang dan kesempatan bisnis yang akan datang, teknologi NFT sejatinya masih di tahap permulaan. Ada beberapa hambatan yang perlu diatasi dengan seksama, dan sejumlah peluang besar yang perlu diperhatikan (Khan et al., 2023).

NFT yang didukung oleh teknologi *blockchain* dan *smart contract* menjanjikan peluang besar bagi industri kreatif, meskipun kehadiran mereka telah mengguncang pasar. Namun, NFT juga memiliki keterbatasan yang berasal dari teknologi *blockchain* dasar. Blockchain sendiri memastikan kesepakatan yang terpercaya dalam sistemnya yang terdistribusi dengan mengandalkan komputer-komputer (sering disebut sebagai "miners") untuk menyelesaikan masalah matematika yang kompleks. Salah satu hambatan besar dalam evolusi teknologi *blockchain* adalah interoperabilitas. Walaupun *blockchain* memberikan solusi yang kokoh dan dapat diandalkan, beragam varian dan tipe *blockchain* yang ada kini kerap kali mengalami kesulitan dalam berinteraksi dan berkomunikasi dengan baik. Hal ini mengimplikasikan bahwa *smart contract* atau NFT yang dikembangkan pada suatu *blockchain* mungkin tidak kompatibel atau dikenali oleh blockchain tipe lain. Kendala ini membatasi kemampuan *blockchain* untuk berkembang dan berintegrasi dengan sistem industri yang lebih besar, seperti sektor keuangan, medis, atau bisnis internasional (Malik et al., 2023).

Blockchain dan Smart Contract menjadi fondasi penting dalam mewujudkan visi Metaverse dan WEB3.0. Blockchain menyediakan transparansi, keamanan, dan kepercayaan dengan menyimpan data dalam jaringan yang terdesentralisasi, sementara Smart Contract memungkinkan otomatisasi transaksi dan perjanjian di dunia digital tanpa perlu perantara. Kombinasi dari teknologi-teknologi ini dapat membentuk fondasi untuk era baru di dunia digital, di mana interaksi lebih aman, transparan, dan tanpa hambatan (Gadekallu et al., 2022).

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non

justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

## 1.2 Permasalahan

Berdasarkan latar belakang di atas, maka dapat dirumuskan masalah pada Tugas Akhir ini adalah kekurangan interoperabilitas NFT di antara berbagai platform.

## 1.3 Batasan Masalah atau Ruang Lingkup

Penelitian ini akan menggunakan platform blockchain Ethereum, mengingat SOLIDITY merupakan bahasa pemrograman khusus untuk Ethereum. Lalu penelitian akan difokuskan pada NFT dalam bentuk properti virtual. Penelitian tidak akan mencakup aspek-aspek hukum atau regulasi terkait transaksi NFT. Evaluasi kinerja akan berfokus pada interoperabilitas dan keandalan *smart contract*, dan tidak mencakup aspek-aspek lain seperti efisiensi atau skalabilitas. Pengujian dan validasi akan dilakukan dalam lingkungan *testnet*, bukan pada jaringan *blockchain* utama atau *mainnet*.

## 1.4 Tujuan

Tujuan utama dari penelitian ini adalah

1. Mengembangkan framework blockchain yang dapat berinteraksi dan berkomunikasi dengan blockchain lain tanpa hambatan
2. Mengembangkan smart contract yang dirancang khusus untuk memastikan bahwa NFT dari platform lain dapat dikenali, diperdagangkan, dan diintegrasikan dengan mudah

## 1.5 Batasan Masalah

Batasan-batasan dari penelitian tugas akhir ini terdapat beberapa, yakni adalah:

1. Pembuatan sistem dikerjakan dalam lingkup *testnet*.
2. *smart contract* dibuat dengan menggunakan bahasa pemrograman Solidity.
3. Menggunakan Hardhat sebagai *ethereum development environment* dan Metamask sebagai virtual wallet.

## 1.6 Sistematika Penulisan

Laporan penelitian tugas akhir ini terbagi menjadi Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. yaitu:

### 1. BAB I Pendahuluan

Bab ini berisi Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis.

## 2. **BAB II Tinjauan Pustaka**

Bab ini berisi Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit.

## 3. **BAB III Desain dan Implementasi Sistem**

Bab ini berisi Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst.

## 4. **BAB IV Pengujian dan Analisa**

Bab ini berisi Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis.

## 5. **BAB V Penutup**

Bab ini berisi Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna.

*[Halaman ini sengaja dikosongkan]*

## **BAB II**

### **TINJAUAN PUSTAKA**

Demi mendukung penelitian ini, terdapat beberapa jurnal penelitian terdahulu yang dijadikan sebagai pacuan dan referensi

#### **2.0.1 Non-Fungible Token (NFT): Overview, Evaluation, Opportunities and Challenges**

Qin Wang, bersama dengan dua rekan lainnya melakukan penelitian terhadap NFT yang dapat merubah pasar digital atau aset virtual. Dari hasil penelitian tersebut memberikan gambaran mendalam tentang teknologi NFT, potensinya, serta tantangan yang dihadapinya. Akan tetapi dalam jurnal penelitian tersebut tidak ada sangkut pautnya dengan pengimplementasian sistem blockchain dan smart contract yang sesuai judul dari penelitian, yaitu interoperabilitas dengan platform penyedia NFT lain.

#### **2.0.2 Blockchain technology for creative industries: Current state and research opportunities**

Nikhil Malik dan tiga rekan lainnya menuliskan jurnal penelitian terhadap teknologi blockchain yang dapat memiliki dampak kepada industri kreatif, seperti musik, desain grafis, permainan, dan software. Dalam jurnal tersebut, mereka menyoroti bagaimana NFT (non-fungible tokens) dan *smart contracts* memberikan peluang menarik bagi industri kreatif. Meskipun teknologi ini telah menciptakan kegembiraan besar di pasar, di tengah-tengah kegemparan tersebut muncul nilai nyata bagi industri tersebut. Secara tradisional, para pencipta di industri kreatif seringkali harus bergantung pada perantara yang kuat untuk mendistribusikan dan mendapatkan keuntungan dari kreasi mereka. Namun, dengan adanya NFT dan *smart contracts*, para pencipta kini dapat lebih dekat dengan konsumen/pembeli konten mereka. Selain itu, jurnal ini juga menggali fraksi pasar dan "biaya transaksi" yang dihadapi para pencipta ketika mendistribusikan konten kreatif mereka serta bagaimana *smart contracts* dan NFT dapat mengubah dinamika pasar dengan mengurangi biaya-biaya tersebut. Namun, mereka juga menunjukkan keterbatasan dan tantangan yang mungkin dihadapi oleh para pencipta, pembeli, dan pasar dalam adopsi NFT dan *smart contracts*.

#### **2.0.3 An Overview on Smart Contracts: Challenges, Advances and Platforms**

Penelitian tentang smart contract yang dilakukan oleh Zibin Zheng dan enam rekan lainnya ini menghasilkan pengetahuan berupa tinjauan mengenai teknologi smart contract yang mutakhir, tantangan dalam berbagai aspek pembuatan, penyebaran, eksekusi, dan penyelesaian *smart contracts*, perbandingan beberapa platform smart contract utama, serta ulasan tentang teknologi smart contract dan blockchain. Tetapi dalam jurnal ini tentu saja terdapat beberapa kekurangan yang disampaikan oleh penulis yaitu meskipun smart contract berkembang pesat, masih ada banyak tantangan yang perlu diatasi.



## 2.0.4 Blockchain for the metaverse: A Review

Penelitian ini menyelidiki secara komprehensif peran dan dampak *blockchain* untuk dasar dan pengembangan aplikasi dan layanan di metaverse. Jurnal yang ditulis oleh Thien Gadekallu dan 7 peneliti lain ini memberikan gambaran beserta dengan pengetahuan mengenai konsep dasar *blockchain* dan metaverse. Dalam jurnal ini pembaca dapat mengetahui tantangan yang perlu ditangani, seperti algoritma konsensus, manajemen jaringan, dan interoperabilitas *blockchain*.

## 2.1 Dasar Teori

### 2.1.1 Blockchain

Blockchain merupakan teknologi basis data terdistribusi yang mengizinkan data disimpan dalam serangkaian blok yang terkoneksi. Tiap blok mengandung rangkaian transaksi yang sudah divalidasi, dan setiap penambahan blok baru ke dalam rantai menjamin keabadian dan ketidakmampuan modifikasi dari transaksi yang tercatat. Fitur keamanan dan keterbukaan menjadi ciri khas *blockchain* karena setiap transaksi divalidasi oleh jaringan partisipan dan tercatat dalam sebuah rantai yang tak bisa dimodifikasi. Berkat struktur terdesentralisasinya, *blockchain* menawarkan ketahanan terhadap modifikasi atau upaya peretasan, sehingga memastikan keutuhan dan keotentikan data. (Tjokrosetio, 2022)

Blockchain menyediakan catatan keuangan yang aman dan disimpan secara terdesentralisasi. Fungsi ini memiliki dampak penting bagi pemasaran. Berdasarkan fungsi ini, muncul dua teknologi yang relevan bagi industri kreatif, yaitu smart contract dan non-fungible tokens (NFT). NFT mengidentifikasi karya seni unik dan mencatat kepemilikannya di *blockchain*. Smart contract adalah program yang disimpan di *blockchain* yang otomatis menjalankan perjanjian saat kondisi yang telah ditentukan terpenuhi. Smart contract dapat digunakan untuk menetapkan aturan penjualan, penggunaan, dan lisensi NFT (Malik et al., 2023).

### 2.1.2 Ethereum

Ethereum, sebuah platform *blockchain* terdesentralisasi yang diperkenalkan oleh Vitalik Buterin pada tahun 2013, memfasilitasi pembuatan dan pelaksanaan aplikasi terdistribusi, dikenal sebagai kontrak pintar (*smart contract*). Platform ini, yang memajukan ide *blockchain*, mengizinkan eksekusi kode pemrograman tingkat lanjut langsung dalam sistem *blockchain*-nya, menggunakan mata uang digital bernama Ether (ETH) untuk melangsungkan transaksi dan mengoperasikan *smart contract*. Ether (ETH), mata uang digital dalam jaringan Ethereum, tidak hanya berfungsi sebagai alat pembayaran tetapi juga diperlukan untuk menjalankan *smart contract*. Setiap kali *smart contract* dijalankan, pemiliknya harus membayar biaya dalam Ether, dikenal sebagai biaya *gas*, yang mencegah spam dan memastikan penggunaan sumber daya jaringan yang efisien. Fitur penting Ethereum adalah kemampuan Turing-completeness, yang memungkinkannya menjalankan berbagai aplikasi terdistribusi. Ini memberi pengembang kebebasan untuk menciptakan aplikasi yang lebih kompleks dan beragam, mulai dari permainan hingga pasar digital, sistem keuangan terdesentralisasi, identitas digital, dan banyak lagi. Selain itu, Ethereum memungkinkan penciptaan token ERC-20, memberikan pengguna kemampuan untuk membuat dan mengelola token mereka sendiri di ekosistem Ethereum. Token ini digunakan untuk berbagai tujuan, termasuk dalam Initial Coin Offering (ICO) untuk penggalangan dana proyek baru, atau sebagai aset digital yang dapat ditukar di bursa kripto (Antonopoulos et al., 2018).

### 2.1.3 *Ethereum Virtual Machine (EVM)*

Ethereum Virtual Machine (EVM) adalah komponen kunci yang memungkinkan Ethereum untuk berfungsi sebagai platform bagi *smart contracts*. Setiap komputer yang terhubung ke jaringan Ethereum memiliki EVM yang memungkinkannya untuk menjalankan kode yang sama, memastikan bahwa setiap smart contract menghasilkan output yang konsisten di seluruh jaringan. EVM bersifat Turing lengkap, yang berarti ia mampu mengeksekusi berbagai instruksi yang diberikan kepadanya, asalkan instruksi tersebut disampaikan dalam format bytecode yang ia mengerti. Pengembang biasanya menulis smart contract dalam bahasa yang lebih aksesibel seperti Solidity atau Vyper sebelum mengompilasinya menjadi bytecode untuk EVM. EVM tidak hanya menjalankan kode ini, tetapi juga mengawasi dan memelihara keadaan semua informasi di *blockchain* Ethereum, termasuk saldo akun dan data yang disimpan oleh *smart contracts*. Untuk menjaga kestabilan dan mencegah penyalahgunaan sistem, EVM menggunakan sistem biaya yang dikenal sebagai gas, yang membatasi jumlah sumber daya yang dapat dikonsumsi oleh setiap smart contract. Dengan demikian, EVM adalah mesin yang kuat dan fleksibel yang memungkinkan berjalannya aplikasi terdesentralisasi yang aman dan tidak dapat diubah dalam lingkup Ethereum. (Dannen, 2017)

### 2.1.4 *Token*

Token secara umum didefinisikan sebagai unit nilai digital yang tercatat pada *blockchain*. Dalam ekosistem *blockchain*, token dapat diklasifikasikan ke dalam beragam kategori berdasarkan ciri khasnya. Dua klasifikasi utama token dalam *blockchain* adalah token berbasis klaim dan token berbasis objek. Token berbasis objek memiliki nilai intrinsik, biasanya ditentukan oleh dinamika permintaan dan penawaran. Sebaliknya, *token* berbasis klaim, yang sering dikenal sebagai *stablecoin*, dirancang untuk mempertahankan nilai stabilnya, seringkali dengan mengaitkannya dengan nilai aset likuid (Freni et al., 2022). Di dalam *blockchain* Ethereum, terdapat standar tertentu yang harus diikuti agar token dapat berinteraksi dengan berbagai aplikasi desentralisasi yang ada di dalam jaringan. Beberapa standar yang telah ditetapkan untuk pengembangan di *blockchain* Ethereum meliputi:

- ERC-20, *interface* standar untuk fungible token, di mana setiap unit token memiliki nilai yang sama, memungkinkan pertukaran dalam jumlah yang setara.
- ERC-721, *interface* standar untuk non-fungible token, di mana setiap token bersifat unik dan memiliki nilai yang berbeda.
- ERC-1155, *interface* standar yang berlaku untuk baik fungible maupun non-fungible token.
- ERC-1633, *interface* standar untuk memungkinkan pembagian kepemilikan NFT, sehingga satu NFT bisa dimiliki oleh lebih dari satu pihak.
- ERC-4907, *interface* standar yang memperkenalkan peran baru dalam NFT, yaitu pemilik (owner) dan pengguna (user) yang memiliki akses terbatas terhadap NFT tersebut selama jangka waktu tertentu.

### 2.1.5 *Non-Fungible Token (NFT)*

Non-Fungible Token atau NFT merupakan varian mata uang kripto yang berasal dari *smart contract* di Ethereum. NFT pertama kali diperkenalkan melalui Ethereum Improvement Pro-

posals (EIP)-721 dan diperluas melalui EIP-1155. NFT membedakan diri dari mata uang kripto konvensional, misalnya Bitcoin. Jika Bitcoin memiliki koin-koin yang sama dan identik, NFT justru dikenal dengan keunikannya yang tidak bisa digantikan atau ditukar secara setara, menjadikannya alat identifikasi yang unik. Menggunakan NFT di dalam *smart contract* Ethereum memungkinkan pencipta untuk memvalidasi keberadaan dan kepemilikan aset digital seperti video, gambar, karya seni, tiket, dan sebagainya. Pencipta bahkan dapat mendapatkan royalti setiap kali aset mereka diperdagangkan di pasar NFT atau melalui transaksi peer-to-peer. Berkat fitur transaksinya yang transparan, likuiditas tinggi, serta kemudahan integrasi, NFT menjadi alat yang efektif dalam melindungi hak kekayaan intelektual. Walaupun esensinya NFT hanyalah kode, namun kode tersebut memiliki nilai tersendiri bagi pembeli karena kelangkaannya, menjadikannya alat penting dalam menetapkan harga bagi produk dengan hak kekayaan intelektual di dunia digital (Wang et al., 2021).

### 2.1.6 Minting

Dalam konteks *smart contracts*, "minting token" mengacu pada proses pembuatan dan pelepasan *token* baru dalam jaringan *blockchain*. Proses ini krusial dalam mengembangkan ekosistem *token* untuk aplikasi dan platform berbasis *blockchain*. Minting token umumnya dilakukan melalui *smart contracts* yang mengadopsi standar *token* seperti ERC-20 atau ERC-721 di Ethereum. *Smart contracts* tersebut memiliki fungsi khusus untuk menghasilkan *token* baru dan menetapkan karakteristiknya, biasanya dilakukan oleh pemilik kontrak atau entitas berwenang lainnya. Selama minting, catatan transaksi yang merekam pembuatan dan penerbitan *token* baru juga direkam secara permanen dalam *blockchain*, menjamin keaslian dan riwayat kepemilikan *token*. Informasi ini dapat diakses dan diverifikasi oleh semua peserta jaringan, menawarkan transparansi dan kepercayaan dalam ekosistem *token*.

Untuk *smart contracts* berstandar ERC-20, fungsi mint atau fungsi serupa digunakan untuk menghasilkan *token* baru. Fungsi ini memerlukan parameter seperti jumlah *token* yang akan dibuat dan alamat pemilik *token* pertama. Setelah dipanggil, kontrak akan menghasilkan *token* baru dan menetapkan ciri-cirinya seperti simbol, nama, total pasokan, dan pemilik awal. Dalam kontrak pintar ERC-721, proses minting melibatkan fungsi yang mirip. Namun, dalam konteks *token* non-fungible (NFT), tiap *token* memiliki identitas dan karakteristik unik. Proses minting di ERC-721 melibatkan penetapan atribut spesifik untuk setiap *token*, termasuk metadata yang menggambarkan aset yang direpresentasikan oleh *token*.

Proses minting *token* memiliki dampak luas dalam ekosistem *blockchain*, memungkinkan penciptaan *token* baru untuk berbagai aplikasi seperti keuangan, pasar NFT, game *blockchain*, identitas digital, dan lainnya. Minting *token* membantu pengembang menciptakan ekonomi digital yang dinamis dan inovatif di atas teknologi *blockchain*.

### 2.1.7 Smart Contracts

Smart Contract adalah protokol transaksi yang diotomatisasi, yang melaksanakan syarat-syarat dalam suatu perjanjian. Ketika suatu kondisi tertentu dipenuhi, klausul dalam *smart contracts* akan otomatis dieksekusi. Smart contract dibuat di atas *blockchain*, di mana setiap eksekusi pernyataan kontrak dicatat sebagai transaksi yang tidak dapat diubah. Dalam pengembangannya, pengembang dapat menentukan izin akses untuk setiap fungsi dalam kontrak. Misalnya, jika seseorang melanggar kontrak, sanksi yang ditentukan dalam kontrak akan otomatis dikenakan. Smart Contract merupakan kemajuan penting dalam teknologi *blockchain* (Zheng et al., 2020).

Smart Contract dapat dikembangkan dan diterapkan di berbagai platform *blockchain* (misalnya, NXT, Ethereum, dan Hyperledger Fabric). Beberapa platform menawarkan fitur khusus untuk mengembangkan *smart contracts*, termasuk bahasa pemrograman, eksekusi kode, dan tingkat keamanan. Beberapa platform mendukung bahasa pemrograman tingkat tinggi untuk mengembangkan *smart contracts* (Khan et al., 2023).

### **2.1.8 InterPlanetary File System (IPFS)**

*InterPlanetary File System* (IPFS) adalah platform berbagi file yang desentralisasi, dirancang untuk menyimpan dan mengakses file, website, aplikasi, dan data dengan cara yang lebih efisien dan tidak bergantung pada server tunggal. IPFS menggunakan teknologi hash kriptografi untuk mengidentifikasi setiap potongan data secara unik, sehingga memudahkan verifikasi dan pengambilan data tanpa perlu mengandalkan lokasi pusat. Selain itu, IPFS menggabungkan file menjadi unit yang lebih besar menggunakan grafik yang diarahkan dan tidak berulang, yang dikenal sebagai Merkle DAGs. Ini memungkinkan distribusi dan penyimpanan data secara efisien di banyak komputer.

IPFS mengurangi redundansi dan meningkatkan kecepatan dengan menyimpan data hanya sekali namun membuatnya dapat diakses dari banyak komputer yang memiliki salinan atau bagian dari data tersebut. Ketika file diunggah ke IPFS, file tersebut dipecah menjadi blok-blok kecil yang masing-masing diidentifikasi melalui hash kriptografi. Jika dua file atau bagian dari file mengandung data yang sama, mereka akan berbagi blok yang sama di IPFS, menghemat ruang dan mengurangi duplikasi.

Keunggulan terbesar IPFS adalah kemampuannya untuk membuat web lebih tahan terhadap kegagalan, karena menghilangkan titik kegagalan tunggal dan memberikan distribusi yang lebih luas dari konten. Hal ini sangat berharga dalam kasus sensor atau infrastruktur internet yang terbatas. Dengan menyimpan data di banyak tempat, akses ke informasi bisa menjadi lebih cepat dan lebih andal, meningkatkan pengalaman pengguna global terhadap internet yang lebih terdesentralisasi. (Steichen et al., 2018).

### **2.1.9 Web 3.0**

Web 3.0 merupakan era baru dalam perkembangan internet, dikenal sebagai internet desentralisasi atau semantic web. Konsep ini muncul sebagai respons terhadap kekuasaan terpusat yang dimiliki oleh raksasa teknologi dalam mengelola data di Web 2.0, yang sering kali menimbulkan masalah privasi dan monopoli data. Web 3.0 memanfaatkan teknologi blockchain untuk memberikan kontrol lebih besar kepada pengguna atas informasi pribadi mereka, memungkinkan interaksi yang lebih transparan dan terpercaya tanpa perantara. Dalam Web 3.0, setiap individu bisa berinteraksi secara langsung melalui aplikasi terdesentralisasi (DApps) yang berjalan di atas blockchain, memfasilitasi transaksi keuangan, pertukaran data, dan komunikasi yang aman. (Ray, 2023)

Web 3.0 tidak hanya mengubah cara kita berinteraksi dengan web melalui penggunaan teknologi blockchain, tetapi juga melalui penggabungan teknologi kecerdasan buatan dan machine learning untuk menciptakan pengalaman pengguna yang lebih intuitif dan responsif. Teknologi ini memungkinkan mesin untuk memahami makna atau semantik dari data di web, sehingga dapat menyajikan konten yang lebih relevan dan personalisasi kepada pengguna. Dengan semantik yang lebih kaya, Web 3.0 diharapkan dapat menyajikan pencarian yang lebih efisien, merekomendasikan konten yang lebih akurat berdasarkan kebutuhan pengguna, dan bahkan mengotomatisasi tugas-tugas yang kompleks. (Chohan, 2022)

Adopsi Web 3.0 juga menjanjikan potensi besar dalam revolusi sektor keuangan dan bisnis. Teknologi blockchain yang menjadi tulang punggung Web 3.0 menawarkan kemungkinan untuk merevolusi kontrak dan layanan keuangan dengan kontrak pintar yang dapat dieksekusi secara otomatis tanpa perlu intervensi manusia. Ini membuka peluang bagi pengembangan ekonomi digital yang benar-benar baru, di mana transparansi, keamanan, dan efisiensi adalah prinsip utama. Dengan demikian, Web 3.0 tidak hanya membawa perubahan pada tata cara kita menggunakan internet tetapi juga mendefinisikan ulang struktur ekonomi global, memungkinkan kolaborasi dan inovasi tanpa batas dalam semua sektor. (Bambacht & Pouwelse, 2022)

### **2.1.10 Cross-chain Interoperability Protocol**

Konsep *cross-chain interoperability* menjadi semakin penting karena memungkinkan berbagai jaringan *blockchain* untuk berkomunikasi dan berbagi informasi serta nilai dengan lancar. Proses ini, yang sangat penting untuk meningkatkan kolaborasi dan efisiensi di antara berbagai sistem *blockchain*, dijelaskan secara mendetail dalam sebuah studi oleh (Lu et al., 2023). Mereka mengeksplorasi pendekatan baru interoperabilitas lintas rantai yang disebut CCIP, dirancang untuk *blockchain* konsorsium menggunakan teknologi *blockchain oracle*. Pendekatan ini dirancang untuk memfasilitasi interaksi dua arah yang kuat dan aman di antara jaringan *blockchain* yang spesifik untuk industri, mengatasi tantangan signifikan seperti efisiensi transmisi data dan keamanan di luar *blockchain konsorsium*.

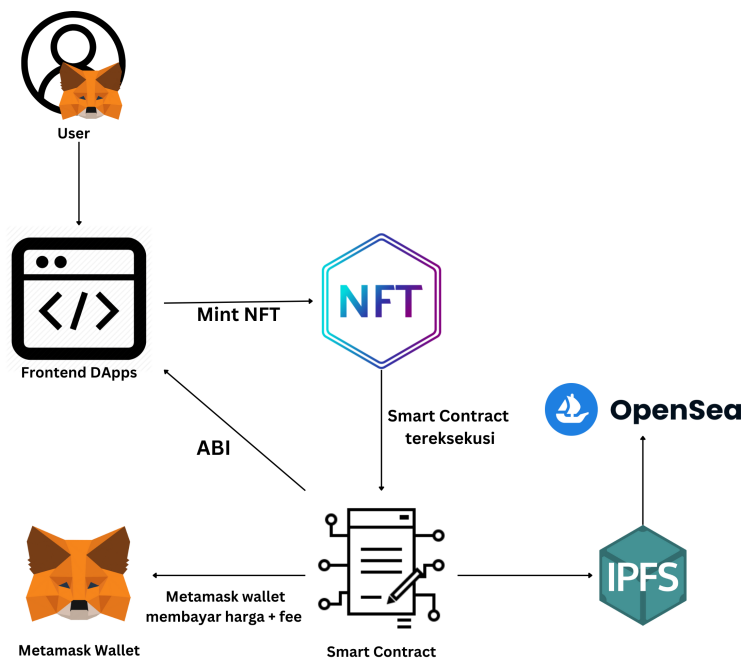
Kerangka CCIP meningkatkan sistem *oracle* tradisional dengan mengintegrasikan metode enkripsi simetris dan asimetris untuk mengamankan data di lintas rantai. Ini memastikan bahwa data sensitif yang ditransmisikan melintasi jaringan *blockchain* mempertahankan integritas dan kerahasiaan. Efektivitas CCIP dibuktikan oleh hasil eksperimental yang menunjukkan efisiensi tinggi dalam interaksi data lintas rantai, yang sangat sesuai dengan kebutuhan *blockchain* konsorsium skala besar. Kerangka kerja ini tidak hanya mendukung tuntutan operasional ekosistem blockchain modern tetapi juga selaras dengan tujuan luas interoperabilitas teknologi di era digital (Pillai et al., 2020).

## BAB III

### METODOLOGI

#### 3.1 Deskripsi Sistem

##### 3.1.1 Arsitektur Sistem

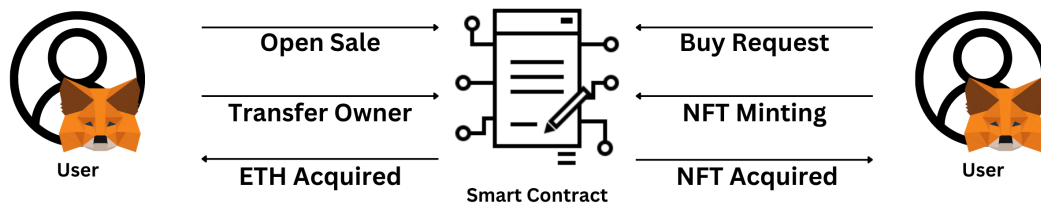


Gambar 3.1: Arsitektur Sistem

Dalam pengembangan sistem *smart contract* ini agar berjalan sesuai kehendak penulis untuk memenuhi tugas akhir ini, penulis membuat suatu arsitektur sistem. Pada arsitektur sistem ini, terdapat role *user* yang dapat menjadi *owner* dari suatu *token* dan *user* merupakan pihak yang memiliki akses terbatas terhadap suatu token dalam waktu yang telah ditentukan. Kemudian juga terdapat *front end* untuk melakukan proses minting dari token NFT. Untuk dapat mengakses *front end* baik owner ataupun user harus terhubung menggunakan wallet dari Metamask dalam berinteraksi.

Pada *front end* akan terhubung dengan *smart contract* ketika pengguna melakukan Minting dimana prosesnya pengguna mengunggah aset digital yang dimiliki dan diunggah ke IPFS melakukan aplikasi backend dan akan mengembalikan Content Identifier (CID). CID merupakan sebuah address file dalam IPFS yang digunakan untuk mengakses file tersebut. CID yang diperoleh kemudian akan diunggah ke jaringan *blockchain* dan menjadi suatu token. Data-data mengenai NFT yang tersedia dapat langsung diperoleh *frontend* melalui *smart contract* pada *blockchain*.

### 3.1.2 Flow Pembelian NFT



Gambar 3.2: Flow Pembelian NFT

Pada sistem ini terdapat juga fitur untuk melakukan pembelian NFT. Pembelian pada *smart contract* yang akan dikembangkan menggunakan fungsi dari *interface* token basis ERC-721. Interface ini memungkinkan token non-fungible diperjual belikan dalam satu kontrak. Kontrak ini akan tereksekusi ketika pengguna kedua (pembeli) melakukan pembelian token milik pengguna pertama (penjual). Kemudian pembeli melakukan minting NFT pada *smart contract* dan juga penjual melakukan pemindahan kepemilikan dari NFT. Setelah proses itu selesai tereksekusi, pembeli mendapatkan NFT dan juga penjual mendapatkan mata uang kripto dari hasil penjualannya.

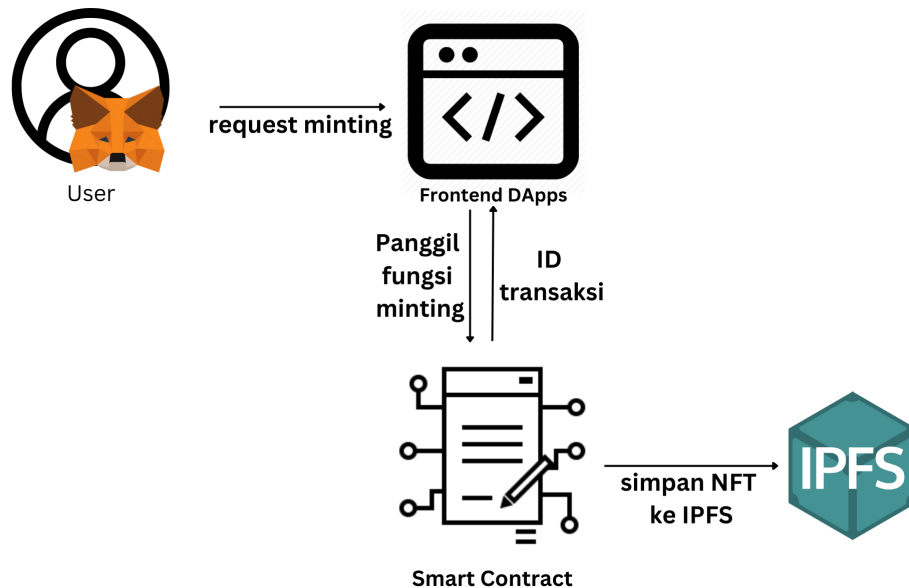
### 3.1.3 Metadata

Listing 3.1: Metadata pada NFT

```
1 {
2   "name": "Letter A",
3   "description": "A bold and modern depiction of the letter 'A'.",
4   "image": "ipfs://QmahP9RgWHTqj25biqqvN91EPkb9fachig4txzRz2AFzSL"
5 }
```

Dalam standar ERC-721, setiap NFT diwakili oleh metadata, yang merupakan kumpulan data mendetail mengenai konten Natternya dalam format JSON (JavaScript Object Notation). Metadata ini biasanya berisi informasi seperti nama, deskripsi, dan sebuah tautan ke gambar, yang semua dapat disesuaikan sesuai dengan kebutuhan aplikasi desentralisasi yang sedang dikembangkan. Struktur data ini disimpan dalam bentuk *array* yang berisi objek-objek dalam format JSON, memungkinkan fleksibilitas tinggi dalam menyesuaikan data yang terkait dengan masing-masing NFT.

### 3.1.4 Minting



Gambar 3.3: Proses Minting

Proses *minting* adalah langkah awal di mana sebuah NFT baru dibuat di dalam sebuah *platform*. Selama proses ini, berbagai informasi penting tentang NFT tersebut harus ditentukan, termasuk nama, deskripsi, kategori, jumlah suplai yang tersedia, dan aset visual yang mewakili NFT, yang bisa berupa gambar 2D, model 3D, atau video. Harga awal, koleksi yang termasuk NFT, serta atribut lainnya juga perlu ditetapkan. Setelah proses minting selesai, data dari NFT yang baru dibuat ini akan disimpan dalam InterPlanetary File System (IPFS), yang memungkinkan penyimpanan dan akses data yang terdesentralisasi. Penyimpanan ini memastikan bahwa semua informasi terkait, seperti kepemilikan yang tercatat dalam atribut NFT, dapat diakses secara permanen dan aman.

## 3.2 Tools yang digunakan

### 3.2.1 Solidity

Solidity adalah bahasa pemrograman tingkat tinggi yang dirancang khusus untuk menulis *smart contracts* di dalam platform *blockchain* Ethereum. Bahasa pemrograman ini dirancang khusus untuk mengatasi *smart contracts*, menyediakan sintaks yang mendukung variabel, fungsi, serta pernyataan kontrol yang diperlukan untuk kontrak yang andal. Solidity juga menyertakan fitur-fitur keamanan yang dapat membantu pengembang menghindari kerentanannya seperti serangan *reentrancy* dan *overflows*. Solidity dirancang untuk berjalan di atas Ethereum Virtual Machine (EVM), yang merupakan lingkungan komputasi Ethereum.

### 3.2.2 Metamask

Metamask adalah perangkat lunak yang memungkinkan pengguna untuk mengelola cryptocurrency, terutama Ether (ETH), serta berinteraksi dengan aplikasi dan layanan terdesentralisasi di *blockchain* Ethereum. Metamask hadir sebagai sebuah Extension pada Browser.



### 3.2.3 Hardhat

Hardhat adalah framework pengembangan *smart contracts* yang didesain khusus untuk mempermudah pengembangan, pengujian, dan distribusi *smart contracts*, Hardhat menyediakan berbagai alat dan fungsi yang kuat bagi pengembang blockchain.

### 3.2.4 Ganache

Ganache adalah alat simulasi *blockchain* Ethereum yang digunakan secara luas dalam pengembangan dan pengujian *smart contracts* dan aplikasi berbasis Ethereum. Ini memungkinkan pengembang untuk membuat jaringan *blockchain* lokal yang dapat dijalankan di komputer mereka sendiri, mirip dengan jaringan Ethereum nyata, tetapi tanpa memerlukan sumber daya yang besar atau biaya transaksi yang sebenarnya.

### 3.2.5 InterPlanetary File System

InterPlanetary File System (IPFS) adalah protokol *hypermedia* terdistribusi yang bertujuan untuk membuat web lebih cepat, aman, dan terbuka. IPFS telah menjadi populer di kalangan pengembang, terutama dalam pengembangan *smart contract* untuk Non-Fungible Tokens (NFTs). Pada dasarnya, IPFS memungkinkan penyimpanan dan berbagi data dalam jaringan terdistribusi yang mirip dengan sistem peer-to-peer. Setiap file dan semua blok dalamnya diberi *hash kriptografis* yang unik, yang bertindak sebagai alamat unik di jaringan.

Dalam konteks NFT, IPFS sangat berguna karena menyediakan solusi untuk tidak tergantung pada server tunggal atau lokasi penyimpanan yang bisa menjadi titik kegagalan. Misalnya, jika *metadata* atau aset digital NFT seperti gambar, video, atau file audio disimpan hanya di server terpusat, ada risiko bahwa data tersebut bisa hilang atau dihapus. Menggunakan IPFS, file tersebut disimpan di beberapa node dalam jaringan, meningkatkan ketahanan dan ketersediaan data. Setiap kali file diminta, IPFS mengambil beberapa potongan dari banyak node, bukan mengambil seluruh file dari satu lokasi, mempercepat waktu pemuatan dan mengurangi beban pada jaringan.

### 3.2.6 React.js

React.js adalah sebuah pustaka JavaScript yang populer dan kuat untuk membangun antarmuka pengguna, atau lebih dikenal dengan istilah UI (*User Interface*). Keunggulan React.js terletak pada kemampuannya untuk membangun aplikasi web yang dinamis dengan cara yang efisien dan mudah. React.js memungkinkan pengembang untuk membuat komponen UI yang besar dan kompleks, yang dapat dikelola dengan baik melalui penggunaan state dan props yang memberikan cara untuk *data flow* dan *re-rendering* yang efisien.

Menggunakan React.js, pengembang dapat memanfaatkan library seperti Web3.js atau Ethers.js untuk berinteraksi langsung dengan *blockchain Ethereum*, tempat *smart contract* di-deploy. Hal ini memungkinkan aplikasi untuk melakukan *query* dan memperbarui *blockchain* dengan transparan, menampilkan informasi seperti kepemilikan NFT, sejarah transaksi, dan *metadata* NFT secara *real-time*.

### 3.2.7 Web3.js

Web3.js adalah sebuah pustaka JavaScript yang sangat penting untuk pengembangan aplikasi web yang berinteraksi dengan *blockchain Ethereum*, khususnya dalam ekosistem Web 3.0. Library ini memungkinkan pengembang *frontend* untuk berkomunikasi langsung dengan

*node* Ethereum, memanfaatkan fungsionalitas *blockchain* seperti *smart contracts*, transaksi, dan manajemen akun langsung dari browser pengguna.

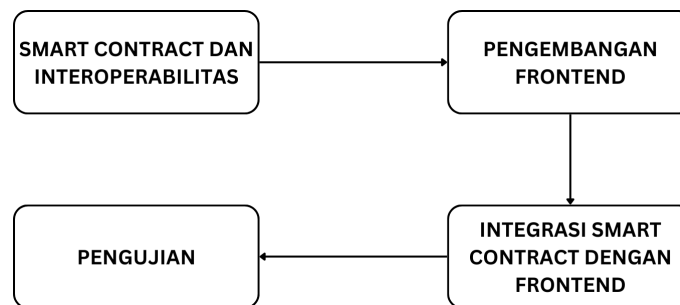
Integrasi Web3.js dalam aplikasi React.js memungkinkan pengembang untuk memanfaatkan *hook* dan komponen React untuk merespons perubahan *state* pada *blockchain* secara *real-time*. Ini termasuk pembaruan saldo, perubahan status transaksi, dan pembaruan pada data *smart contract*. Fungsi ini memperkuat pengalaman pengguna dengan memberikan tampilan yang konsisten dan *up-to-date* dari status *blockchain*, yang sangat penting dalam aplikasi yang menuntut keandalan dan transparansi tinggi.

### 3.2.8 Ethers.js

Ethers.js adalah sebuah *library* JavaScript ringan yang dirancang untuk berinteraksi dengan Ethereum *blockchain*. *Library* ini sangat cocok untuk digunakan dalam pengembangan aplikasi *frontend* dengan React.js karena menawarkan API yang sederhana dan modular, memudahkan pengembangan aplikasi desentralisasi (DApps) yang efisien. Ethers.js menyediakan fungsi-fungsi penting seperti penyambungan dengan *wallet* pengguna, pembuatan dan pengiriman transaksi, serta interaksi dengan *smart contract*.

Dalam pengembangan dengan React.js, Ethers.js membantu mengintegrasikan fungsi *blockchain* ke dalam komponen React dengan cara yang bersih dan efektif. Misalnya, dengan menggunakan React *hooks* seperti *useState* dan *useEffect*, pengembang dapat dengan mudah mengintegrasikan status dari *Ethereum blockchain* ke dalam UI aplikasi, memperbarui UI secara *real-time* ketika ada perubahan pada *blockchain* seperti konfirmasi transaksi atau perubahan pada data *contract*.

## 3.3 Metode Yang Digunakan



Gambar 3.4: Metodologi Penelitian Yang Digunakan

### 3.3.1 Smart Contract dan Interoperabilitas

Pada tahapan ini, penulis mengembangkan sistem *smart contract* yang digunakan untuk pembuatan token yang berinteroperabilitas menggunakan bahasa pemrograman Solidity. Sistem *smart contract* ini digunakan sebagai jembatan untuk memproses request dari yang dihasilkan dari interaksi pengguna pada *frontend* ke jaringan *blockchain*. Pada *smart contract* yang telah di-*compile* akan muncul ABI (*Application Binary Interface*) yang dapat digunakan pada *frontend* sebagai encoding dan decoding data yang memungkinkan *frontend* aplikasi untuk mengirim instruksi yang tepat ke *smart contract* dan memahami data yang dikembalikan olehnya. Dalam *smart contract* yang dibuat oleh penulis terdapat beberapa fungsi yang dibuat sebagai inti pada token yang dikembangkan.

## Fungsi Mint

Fungsi Mint dalam konteks *smart contract* ERC-721 yang digunakan untuk *minting* token NFT merupakan komponen krusial dalam mengelola penerbitan token baru. Ini diakses secara publik dan dapat menerima Ether, yang memungkinkan pembayaran langsung selama proses *minting*.

Listing 3.2: Fungsi Mint

---

```
1 FUNSI mint(_tokenURI: STRING) MENGEMBALIKAN BILANGAN BULAT:
2     TAMBAHKAN tokenCount SEBANYAK 1
3     JALANKAN _safeMint DENGAN PARAMETER (msg.sender, tokenCount)
4     JALANKAN _setTokenURI DENGAN PARAMETER (tokenCount, _tokenURI)
5     KEMBALIKAN tokenCount
6 AKHIR FUNGSI
```

---

- *Increment tokenCount*

Pada awalnya, fungsi ini meningkatkan variabel *tokenCount*. Variabel ini bertindak sebagai pengidentifikasi unik untuk setiap token baru; ini memastikan bahwa setiap token yang dicetak memiliki ID yang unik.

- *Mint Token*

Selanjutnya, fungsi memanggil metode *\_safeMint*. Metode ini bertanggung jawab atas penciptaan token sebenarnya. Ini menetapkan kepemilikan token yang baru dicetak kepada pemanggil fungsi (*msg.sender*), yang biasanya adalah pengguna yang berinteraksi dengan kontrak. Penggunaan *\_safeMint* (dibandingkan *\_mint*) memastikan pemeriksaan tambahan ada untuk mencegah token dikirim ke kontrak yang tidak siap untuk menanganinya, yang dapat mencegah kehilangan token secara tidak sengaja.

- *Set Token URI*

Setelah mencetak token, fungsi memperbarui metadata token dengan memanggil *\_tokenURI*. Metode ini mengaitkan URI yang disediakan (*\_tokenURI*) dengan ID token. URI biasanya mengarah ke file JSON yang di-host secara eksternal yang berisi *metadata* tentang token, seperti nama, deskripsi, dan URL gambar. Metadata ini memperkaya token dan dapat digunakan untuk memberikan informasi lebih rinci tentang aset digital.

- *Return tokenCount*

Terakhir, fungsi mengembalikan *tokenCount*, yang merupakan pengidentifikasi unik dari token yang baru dicetak. ID ini dapat digunakan oleh aplikasi eksternal atau fungsi kontrak lain untuk merujuk token spesifik ini.

Secara keseluruhan, fungsi *mint* ini sangat penting untuk menciptakan aset digital baru di *blockchain*, memberikan pengguna kemampuan untuk menghasilkan NFT unik yang dapat mewakili segala sesuatu mulai dari seni digital hingga *real estate virtual*, memastikan setiap aset dapat diidentifikasi secara jelas melalui ID token yang unik.

## Fungsi *lockToken*

Listing 3.3: Fungsi *lockToken*

---

```
1 FUNGSI lockToken(tokenId: BILANGAN BULAT):  
2     PASTIKAN _exists(tokenId) = TRUE dengan pesan "Token does not exist."  
3     ATUR _locked[tokenId] = TRUE  
4     KIRIMKAN EVENT TokenLocked dengan (tokenId, msg.sender)  
5 AKHIR FUNGSI
```

---

- Pemeriksaan Keberadaan Token

Fungsi ini memulai dengan memeriksa apakah token dengan ID yang diberikan benar-benar ada. Ini dilakukan melalui pemanggilan fungsi *\_exists* yang memeriksa dalam basis data smart contract apakah token tersebut sudah dicetak dan masih ada. Jika token tidak ditemukan, fungsi akan memberikan error dan berhenti, dengan pesan "Token does not exist."

- Mengunci Token

Jika token ditemukan, fungsi kemudian melanjutkan untuk mengunci token tersebut dengan mengatur nilai dari mapping *\_locked* untuk tokenId yang bersangkutan menjadi true. Ini efektif membatasi fungsi smart contract yang dapat berinteraksi dengan token tersebut, khususnya mencegah transfer atau transaksi lain yang mungkin mengubah kepemilikan atau status token.

- Emit Event

Setelah token berhasil dikunci, fungsi kemudian memicu event *TokenLocked* yang menandakan bahwa token telah dikunci. Event ini mencatat tokenId dan alamat yang memicu fungsi (biasanya pemilik atau pengontrol kontrak, sebagaimana ditandai oleh modifier *onlyOwner*). Event ini bisa digunakan untuk memberitahukan pengguna atau aplikasi lain yang berinteraksi dengan blockchain bahwa token tersebut sekarang berada dalam status terkunci dan tidak dapat dioperasikan atau dipindahkan hingga dibuka kunci.

Secara keseluruhan, fungsi *lockToken* berperan penting dalam menjamin keamanan dan integritas transaksi yang melibatkan NFT, khususnya dalam skenario yang melibatkan operasi lintas rantai atau ketika sebuah token sedang menunggu konfirmasi transaksi yang kritis. Fungsi ini membantu memastikan bahwa tidak ada pihak yang tidak berwenang atau proses otomatis lainnya yang dapat mengganggu proses hukum atau komersial yang sedang berlangsung.

## Fungsi *unlockToken*

Fungsi *unlockToken* dalam *smart contract* bertujuan untuk membuka kunci token yang sebelumnya telah dikunci. Fungsi ini sangat penting dalam manajemen aset dalam *smart contract*, terutama dalam kasus penggunaan seperti transaksi lintas rantai atau ketika token harus dikunci untuk alasan keamanan atau administratif.

Listing 3.4: Fungsi *unlockToken*

---

```
1 FUNGSI unlockToken(tokenId: BILANGAN BULAT)  
2     PASTIKAN _locked[tokenId] = TRUE dengan pesan "Token is not locked."  
3     ATUR _locked[tokenId] = FALSE  
4     KIRIMKAN EVENT TokenUnlocked dengan (tokenId, msg.sender)  
5 AKHIR FUNGSI
```

---

- Verifikasi Status Kunci

Langkah pertama dalam fungsi ini adalah memastikan bahwa token yang ditentukan benar-benar dalam keadaan terkunci. Ini dilakukan dengan memeriksa mapping `_locked` untuk `tokenId` yang diberikan. Jika nilai dari mapping ini adalah `false`, yang berarti token tidak terkunci, maka fungsi akan menghasilkan error dan menghentikan eksekusi lebih lanjut dengan pesan "Token is not locked." Ini mencegah upaya untuk membuka kunci token yang tidak perlu atau yang mungkin telah secara tidak sengaja terbuka.

- Membuka Kunci Token

Jika token memang terkunci, fungsi kemudian mengatur nilai dalam mapping `_locked` untuk `tokenId` tersebut menjadi `false`, secara efektif membuka kunci token tersebut. Ini mengizinkan token untuk berpartisipasi dalam transaksi dan interaksi lainnya sesuai dengan aturan smart contract lainnya.

- Emit Event

Setelah token berhasil dibuka kunci, fungsi memicu event `TokenUnlocked` yang menandakan bahwa token tersebut telah dibuka kunci. Event ini mencatat `tokenId` dan alamat pengguna yang menjalankan fungsi (dijamin adalah pemilik atau pengontrol kontrak melalui penggunaan modifier `onlyOwner`). Event ini berguna untuk audit, pemantauan keamanan, dan sebagai bukti dalam aplikasi yang berinteraksi dengan smart contract bahwa token telah kembali ke status normal dan siap untuk dipindahkan atau digunakan.

Secara keseluruhan, fungsi `unlockToken` memainkan peran krusial dalam memastikan fleksibilitas dan keamanan dalam pengelolaan NFT dan aset digital lainnya. Fungsi ini memungkinkan pemilik atau pengontrol kontrak untuk secara efektif mengelola akses dan kontrol atas aset digital, memastikan bahwa aset tersebut hanya terkunci ketika benar-benar diperlukan dan bisa dibuka kembali ketika kondisi memungkinkan.

## Fungsi *bridgeTransfer*

Fungsi *bridgeTransfer* dirancang untuk memungkinkan pemindahan token yang aman antara berbagai blockchain atau sub-jaringan, sebuah proses yang sering disebut sebagai "bridging". Fungsi ini memastikan bahwa NFT (Non-Fungible Token) hanya dapat ditransfer setelah memenuhi kriteria keamanan tertentu.

Listing 3.5: Fungsi `bridgeTransfer`

---

```
1 FUNGSI bridgeTransfer(_to: ALAMAT, _tokenId: BILANGAN BULAT)
2     PASTIKAN _locked[_tokenId] = TRUE dengan pesan "Token must be locked ←
      before bridging"
3     TRANSFER_TOKEN dari owner() ke _to dengan _tokenId
4     KIRIMKAN EVENT TokenBridged dengan (_tokenId, owner(), _to)
5 AKHIR FUNGSI
```

---

- Verifikasi Kondisi Kunci

Langkah awal dalam fungsi ini adalah memastikan bahwa token yang ingin ditransfer sudah terkunci. Ini dilakukan dengan memeriksa nilai dari mapping `_locked` untuk `tokenId` yang diberikan. Jika token ini tidak terkunci, fungsi akan mengeluarkan kesalahan

dan menghentikan proses lebih lanjut dengan pesan *"Token must be locked before bridging"*. Kondisi ini memastikan bahwa token hanya dipindahkan setelah melalui tahapan penguncian yang mencegah penggunaan atau transfer yang tidak sah sebelumnya.

- **Eksekusi Transfer**

Setelah dipastikan bahwa token terkunci, fungsi selanjutnya melakukan transfer NFT dari pemilik saat ini ke alamat tujuan yang ditentukan (*\_to*). Transfer ini dilakukan menggunakan fungsi *\_transfer* yang merupakan bagian dari standar *ERC-721*, memungkinkan perubahan kepemilikan NFT dalam *blockchain*.

- **Pencatatan Event**

Setelah transfer berhasil, fungsi mengirim event *TokenBridged*. Event ini mencatat tokenId, alamat pemilik sebelum transfer (*owner()*), dan alamat tujuan (*\_to*). Event ini penting untuk audit dan pemantauan, memungkinkan aplikasi dan layanan yang terhubung untuk merespons atau mengakui transfer NFT lintas rantai.

Dengan cara ini, *bridgeTransfer* menyediakan mekanisme yang efisien dan aman untuk mengintegrasikan fungsi interoperabilitas dalam *smart contract* NFT, memungkinkan aset digital untuk berpindah antar ekosistem *blockchain* sambil mempertahankan tingkat keamanan dan verifikasi yang ketat. Fungsi ini sangat penting dalam ekosistem *blockchain* yang semakin terhubung, di mana NFT dan aset digital lainnya sering perlu beroperasi di berbagai platform dan jaringan.

### 3.3.2 Pengembangan Frontend

Pada tahap pengembangan frontend dalam proyek ini, kami memilih React.js sebagai kerangka kerja utama karena fleksibilitasnya dan efisiensinya dalam membangun antarmuka pengguna yang dinamis dan responsif. React.js memungkinkan pembuatan komponen yang dapat digunakan kembali, yang sangat meningkatkan efisiensi pengembangan dengan memungkinkan kami untuk membangun UI yang kompleks dari komponen yang lebih kecil dan termodularisasi.

Untuk mengintegrasikan aplikasi React dengan blockchain Ethereum, kami menggunakan dua pustaka utama: ethers.js dan web3.js. Kedua pustaka ini menyediakan fungsionalitas yang diperlukan untuk berinteraksi dengan Ethereum blockchain, tetapi dengan pendekatan yang sedikit berbeda. Ethers.js dikenal dengan API-nya yang minimalis dan mudah digunakan, yang sangat cocok untuk proyek-proyek dengan kebutuhan yang lebih sederhana dan lebih fokus pada pembacaan serta penulisan data ke blockchain. Pustaka ini menyediakan fungsi yang kuat untuk berinteraksi dengan smart contracts, seperti mengirim transaksi, membaca status kontrak, dan menangani notifikasi event.

Di sisi lain, web3.js adalah pustaka yang lebih luas yang sering digunakan untuk proyek yang memerlukan integrasi yang lebih kompleks dengan Ethereum. Web3.js menyediakan modul yang lebih komprehensif dan mendukung interaksi yang lebih beragam dengan blockchain, termasuk pengelolaan akun, komputasi gas, dan langganan event yang lebih canggih. Kedua pustaka ini, ketika digunakan bersama-sama atau secara terpisah, memberikan fleksibilitas yang sangat dibutuhkan dalam pengembangan frontend untuk aplikasi berbasis blockchain.

### 3.3.3 Integrasi Smart Contract dengan Frontend

Pada tahap ini memungkinkan aplikasi yang dibangun dengan React.js untuk berinteraksi secara langsung dengan smart contract yang telah dideploy di jaringan Ethereum. Untuk men-

capai integrasi ini, kami menggunakan pustaka ethers.js, yang menyediakan antarmuka yang bersih dan mudah digunakan untuk berkomunikasi dengan Ethereum.

Setelah smart contract dikembangkan dan dideploy, ABI (Application Binary Interface) dari contract tersebut digunakan untuk membangun sebuah instance contract dalam aplikasi React. ABI memungkinkan frontend untuk mengetahui fungsi-fungsi apa saja yang tersedia dalam smart contract, termasuk variabel dan tipe data yang digunakan. Dengan informasi ini, ethers.js dapat memanggil fungsi-fungsi tersebut seperti fungsi safeMint atau transferOwnership, sesuai dengan logika yang didefinisikan dalam contract.

Dalam aplikasi, kami mengonfigurasi ethers.js untuk terhubung dengan provider Ethereum, yang bisa berupa MetaMask atau node Ethereum lainnya. Ini memungkinkan aplikasi untuk mengirim transaksi dan memantau event yang diterbitkan oleh smart contract. Setiap kali pengguna berinteraksi dengan UI, seperti mengklik tombol untuk memint NFT atau mentransfer kepemilikan, permintaan tersebut diterjemahkan oleh ethers.js menjadi transaksi blockchain yang sesuai.

Selain itu, untuk meningkatkan keamanan dan keandalan aplikasi, kami juga mengimplementasikan penanganan error yang robust untuk mengatasi masalah yang mungkin terjadi selama interaksi dengan blockchain, seperti kegagalan transaksi atau masalah konektivitas. Dengan demikian, pengguna dapat menerima feedback yang tepat waktu dan akurat jika ada masalah yang terjadi selama proses transaksi.

### 3.3.4 Pengujian

Pada tahap ini terdapat beberapa langkah pengujian *smart contract* yaitu pengujian unit dan juga pengujian integrasi.

- **Pengujian Unit** Pada pengujian ini adalah fokus dalam memvalidasi setiap komponen secara individual. Dalam konteks frontend React.js, ini berarti menguji komponen-komponen secara terpisah untuk memastikan bahwa mereka berperilaku sesuai dengan ekspektasi. Pengujian unit juga dilakukan pada fungsi-fungsi smart contract untuk memverifikasi logika bisnisnya, seperti fungsi minting atau transfer token. Pengujian ini biasanya dilakukan dengan bantuan framework React untuk frontend dan Hardhat untuk smart contracts.
- **Pengujian Integrasi** Setelah pengujian unit, langkah berikutnya adalah pengujian integrasi, yang memastikan bahwa semua komponen dalam aplikasi bekerja dengan baik saat digabungkan. Dalam konteks integrasi smart contract, ini melibatkan menguji interaksi antara frontend React.js dan smart contract melalui ethers.js atau web3.js. Pengujian integrasi membantu mendeteksi masalah pada alur data antara frontend dan blockchain, termasuk validasi transaksi dan pembaruan state yang benar.

## BAB IV

### HASIL DAN PEMBAHASAN

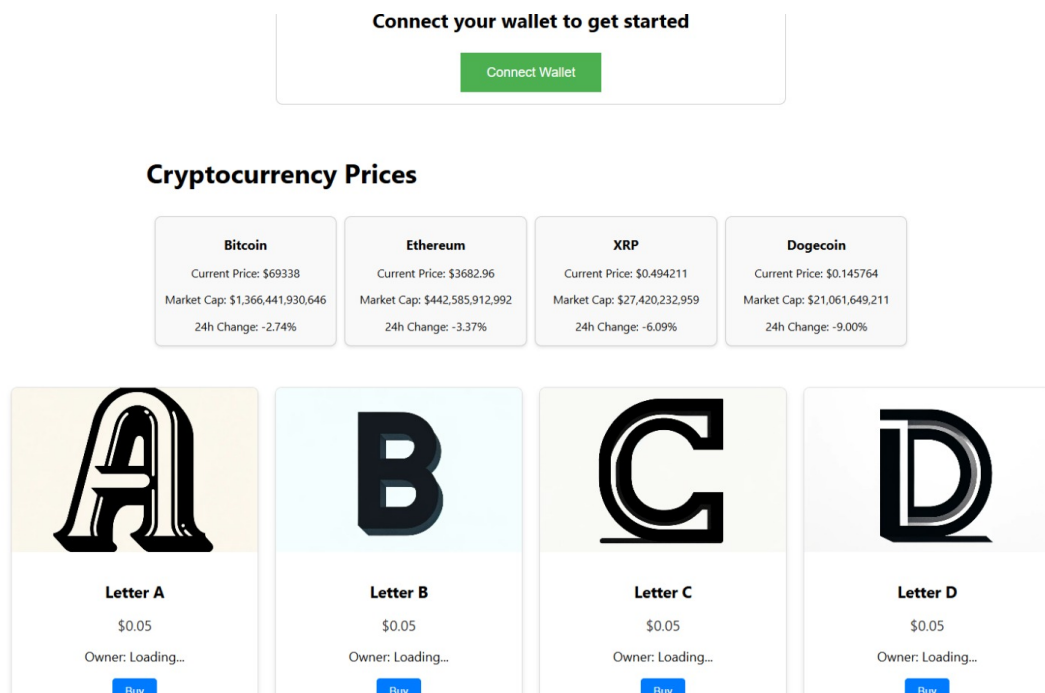
Pada subbab pertama pada bab ini akan dipaparkan hasil akhir dari pengerjaan tugas akhir Interoperabilitas NFT Berbasis *Blockchain* Menggunakan *Smart Contract* pada WEB3.0. Pada subbab selanjutnya akan dipaparkan mengenai hasil pengujian dari hasil tugas akhir untuk memastikan bahwa sistem yang dikembangkan berjalan dengan sesuai.

#### 4.1 Hasil

Hasil keseluruhan dari sistem yang dikembangkan adalah *interface* berupa web, *Non-Fungible Token* (NFT) dan juga *smart contract* yang berinteroperabilitas. Berikut dipaparkan masing-masing hasil dari sistem yang dikembangkan.

##### 4.1.1 Web Interface

Tampilan awal web *interface*, menampilkan daftar koleksi NFT yang dapat di-*minting*. Nantinya NFT tersebut dapat dimiliki oleh *address* yang melakukan pertama kali *minting* dan juga kepemilikan dari NFT tersebut dapat diberikan ke *address* lain.



Gambar 4.1: Tampilan interface

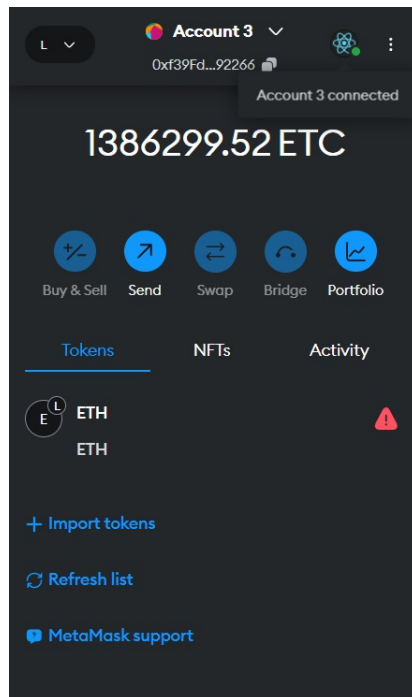
Pada dashboard awalnya pengguna harus menekan tombol *Connect Wallet* yang kemudian akan terhubung dengan akun Metamask Wallet. Dari koneksi tersebut pengguna mendapatkan *address* yang dapat digunakan untuk melakukan *minting* token NFT.



## Welcome to Your Dashboard

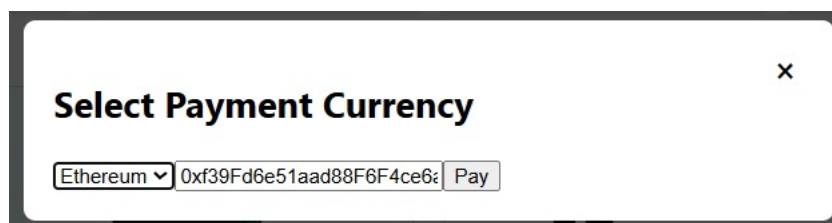
0xf39Fd6e51aad88F6F4ce6aB8827279cFfFb92266

Connect Wallet



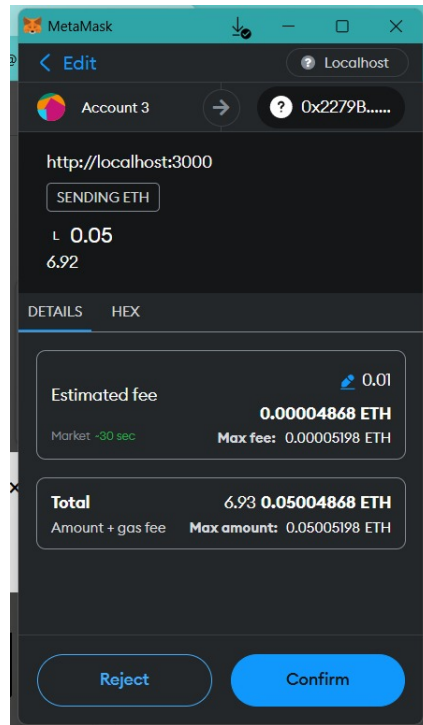
Gambar 4.2: Koneksi *interface* dengan Metamask Wallet

Setelah melakukan proses koneksi dengan Metamask, pengguna dapat melakukan *minting* pada NFT yang tersedia pada *interface* dengan menekan tombol *pay*.



Gambar 4.3: *Payment* dan *address*

Kemudian pada saat pengguna melakukan proses pembelian atau *minting*, pengguna dapat memilih pembayaran dengan *cryptocurrency* mana dan juga memasukkan *address* untuk dapat mencetak kepemilikan NFT tersebut.



Gambar 4.4: Detail *payment* menggunakan Metamask Wallet

Setelah melakukan proses pembayaran dan transaksi berhasil, maka pada *interface* bagian *owner* dari produk NFT tersebut akan diperbarui menjadi *address* dari pemilik yang baru.



Gambar 4.5: *address* pemilik baru dari NFT

#### 4.1.2 Smart Contract

Dalam pengembangan aplikasi berbasis *blockchain*, pengintegrasian *smart contract* dengan antarmuka pengguna seperti React.js menjadi krusial. *Smart contract* yang dibangun menggunakan Solidity dapat di-*deploy* di *testnet* Ethereum, seperti Sepolia Testnet, yang menyedi-

akan lingkungan yang hampir mirip dengan mainnet Ethereum namun tanpa memerlukan biaya transaksi yang besar.

ABI atau *Application Binary Interface*, adalah cara yang memungkinkan fungsi dalam *smart contract* Ethereum untuk berkomunikasi dengan aplikasi eksternal, termasuk antarmuka pengguna yang dibangun dengan kerangka kerja seperti React.js. ABI berperan sebagai lapisan translasi yang menguraikan cara memanggil fungsi dalam smart contract, jenis parameter yang diterima, jenis keluaran yang diharapkan, serta sifat state dari fungsi tersebut. ABI biasanya dihasilkan secara otomatis oleh *compiler* Solidity sebagai bagian dari proses kompilasi *smart contract* dan disimpan dalam format JSON. Setiap kali aplikasi *frontend* mengirimkan transaksi atau *query* ke *blockchain*, ia menggunakan ABI untuk mengkodekan data panggilan ke dalam format yang dapat dipahami oleh *Ethereum Virtual Machine* (EVM). Kemudian, saat data dikembalikan dari *smart contract*, ABI digunakan untuk mendekode balasan sehingga aplikasi React.js dapat memahami dan memprosesnya.

Proses *deploy* dilakukan *smart contract* akan di-*deploy* ke beberapa jaringan *blockchain* namun dikarenakan dalam pengujian ini yang digunakan adalah sepolia *testnet* maka yang digunakan hanyalah dari jaringan sepolia *testnet*. Ketika melakukan proses *deployment* akan dikenakan *gas* atau fee yang dapat dibayar menggunakan *ethereum* yang terdapat pada *wallet* Metamask. Yang kemudian detail dari transaksi tersebut dapat dilihat pada Etherscan.

[ This is a Sepolia **Testnet** transaction only ]

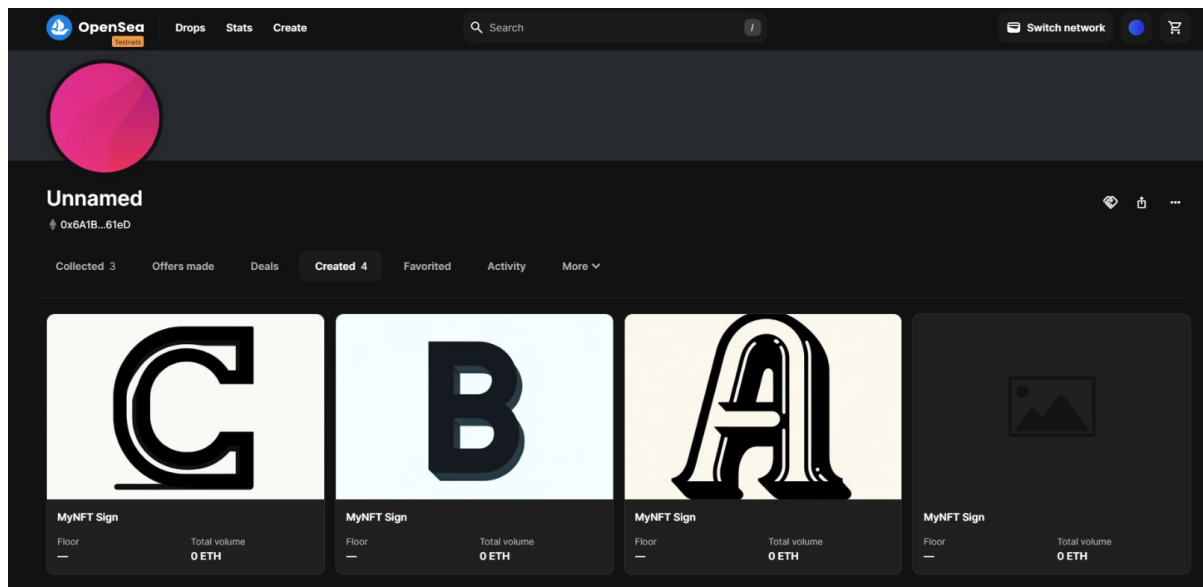
Transaction Hash:	0xf0573d90b5124a32e8947f37de9925fcb6cca6fb9031132c87e3aa357a2042e8
Status:	Success
Block:	6062201 3560 Block Confirmations
Timestamp:	12 hrs ago (Jun-08-2024 03:04:36 AM +UTC)
Transaction Action:	Call 0x60806040 Method by 0x6A1B77e8...5EBf361eD
From:	0x6A1B77e82b61D54C4F1A27cd00A27325EBf361eD
To:	[ 0x8d01bb70bc8bb35e9c39d7f76b370144ca235a5a Created ]
Value:	0 ETH (\$0.00)
Transaction Fee:	0.003570899471483106 ETH (\$0.00)
Gas Price:	1.501265442 Gwei (0.000000001501265442 ETH)

Gambar 4.6: Detail transaksi pada etherscan

### 4.1.3 Non-Fungible Token (NFT)

*Non-Fungible Token* (NFT) sendiri juga memiliki beberapa tahapan agar dapat dapat disimpan dalam *The Interplanetary File System* (IPFS) dan kemudian dapat di-*publish* pada OpenSea *test net*. Pada pengerjaan tugas akhir ini, Pinata digunakan sebagai *platform* untuk mengunggah foto NFT dan juga data *JavaScript Object Notation* (JSON) agar bisa di-*minting* sebagai

*Uniform Resource Identifier (URI)* pada *smart contract*. Hal ini dibutuhkan agar NFT yang di-minting memiliki gambar yang dapat dilihat pada OpenSea.



Gambar 4.7: NFT terunggah pada Opendea

## 4.2 Pengujian

Untuk memastikan bahwa sistem interoperabilitas *Smart Contract* berjalan sesuai dengan yang direncanakan maka perlu dilakukan pengujian lebih lanjut. Pengujian yang dilakukan secara garis besar dibagi menjadi dua yaitu Pengujian fitur-fitur utama dari *Smart Contract* yang meliputi *minting*, *lock*, dan *bridge transfer* kepemilikan NFT dan yang kedua adalah pengujian integrasi ke *interface* yang telah dibuat. Berikut merupakan paparan dari pengujian.

### 4.2.1 Pengujian Fitur *Smart Contract*

Ekspektasi pengujian dari sistem *smart contract* antara lain adalah sebagai berikut:

- *User* dapat melakukan minting token NFT yang kemudian kepemilikan NFT tersebut dapat dilihat pada *test net platform* OpenSea.
- Antar *user* yang berbeda *Network* dapat saling berkomunikasi dan kepemilikan *token* NFT dapat berpindah dari *user A* yang berada pada *network A* ke *user B* yang berada pada *network B*.

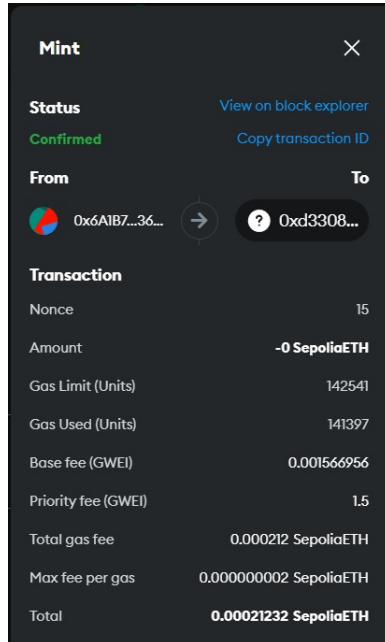
Berikut merupakan data mengenai *wallet* yang digunakan dalam pengujian:

Tabel 4.1: Informasi Akun

Akun	Address	Network
A	0x6A1B77e82b61D54C4F1A27cd00A27325EBf361eD	Sepolia Ethereum Testnet
B	0xD066d6576D9485Eb2c2a41BB8B52EcE17a0557d6	BNB Chain Testnet

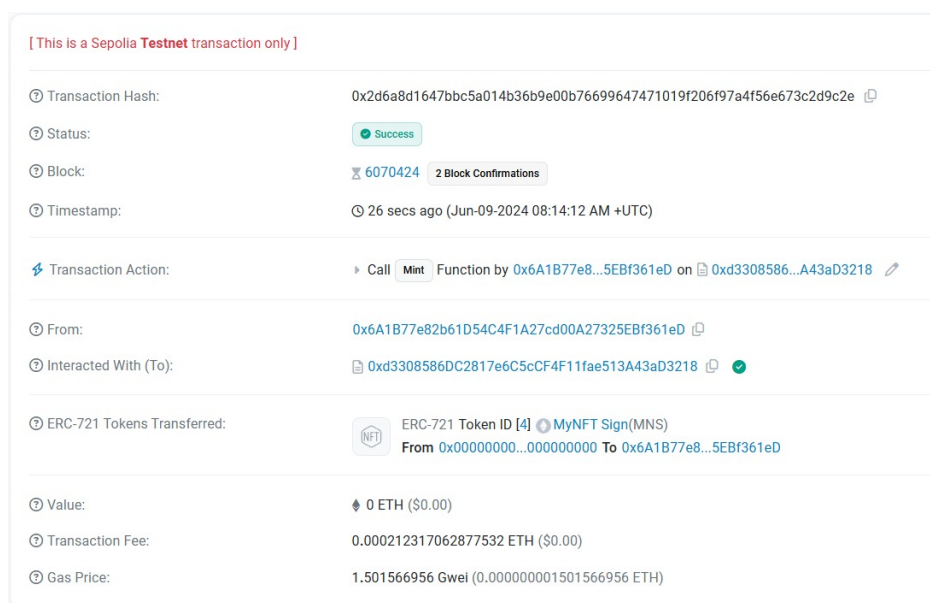
Berikut merupakan langkah-langkah dari pengujian sistem *smart contract* yang akan dilakukan:

- *User A* akan melakukan *minting* pada NFT terlebih dahulu



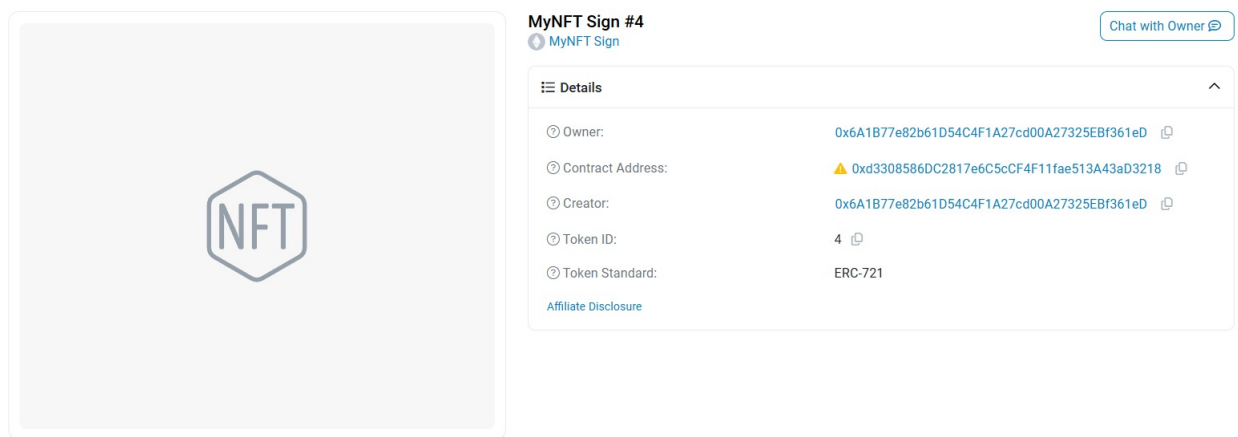
Gambar 4.8: Riwayat transaksi minting pada Metamask Wallet

- dapat dilihat juga pada gambar di bawah terdapat detail pada etherscan yang menunjukkan detail seperti *transaction hash*, *block*, dan tipe dari token yaitu *ERC-721*.



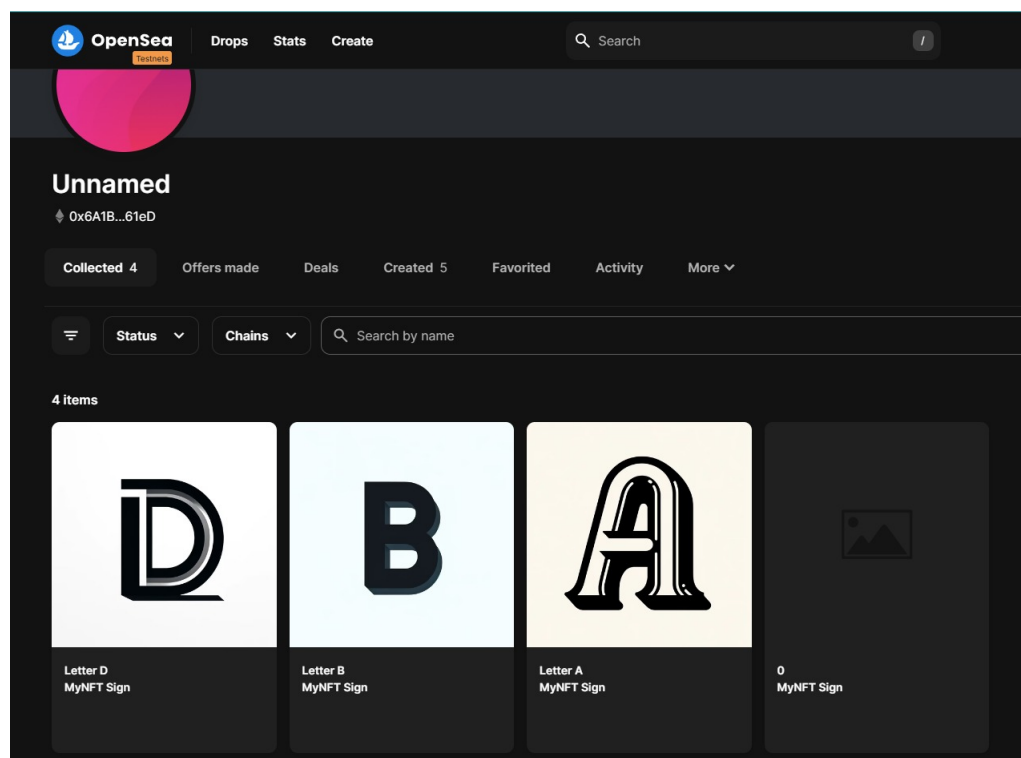
Gambar 4.9: Detail transaksi pada Etherscan

- kemudian pada etherscan juga terdapat detail dari NFT yang telah di-*minting*, yang dapat dilihat seperti pada gambar berikut



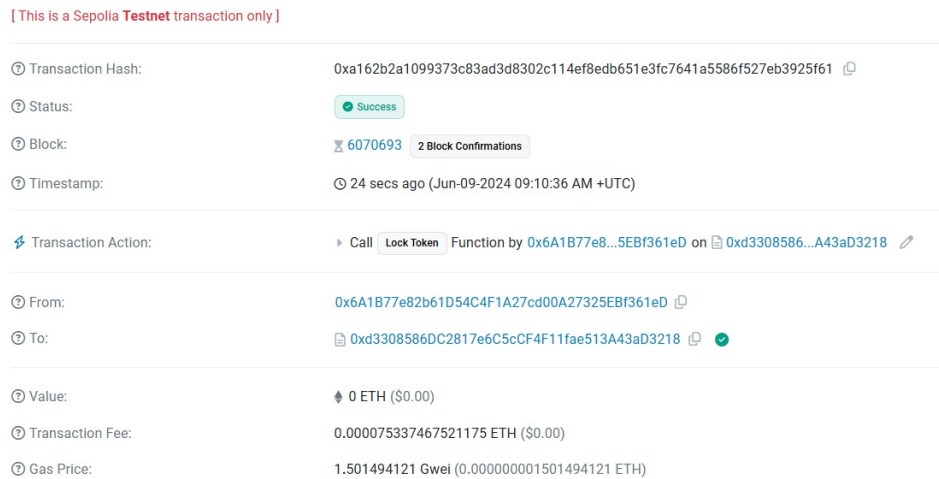
Gambar 4.10: Detail NFT pada Etherscan

- setelah proses tersebut berhasil, maka pada *platform* OpenSea bisa dilihat NFT yang telah di-*minting*. Pada kasus pengujian ini, NFT yang ter-*minting* adalah "Letter D".



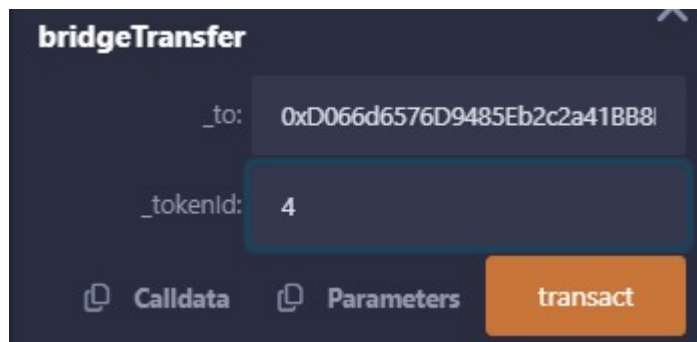
Gambar 4.11: NFT pada OpenSea

- setelah NFT sudah terbuat, maka tahap berikutnya adalah melakukan *lock* pada NFT yang akan dikirimkan ke *user B* pada *network B*. Fungsi dari *lock* ini untuk menjamin bahwa data token tidak diubah selama proses *transfer*, menjaga kepercayaan dan keautentikan data NFT. Kemudian juga berguna untuk memberi sinyal kepada semua pihak terkait (pengguna, *smart contract* di jaringan lain) bahwa token tersebut sedang dalam proses *transfer*, dan operasi pada token harus ditangguhkan hingga proses selesai.



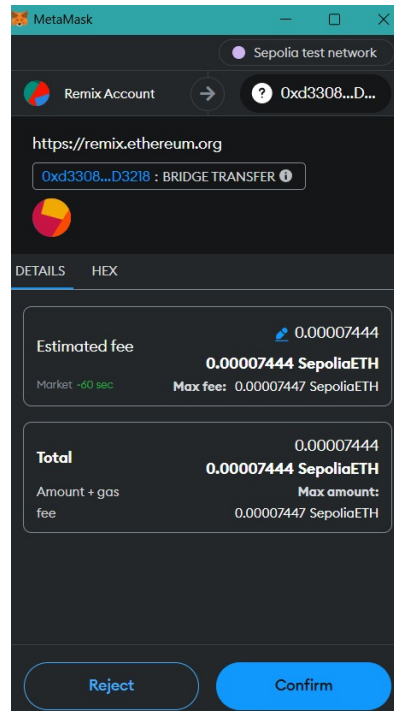
Gambar 4.12: Detail fungsi *Lock Token* pada Etherscan

- selesai melakukan proses *lock token*, lalu fungsi *bridge transfer* dapat dilaksanakan. Fungsi ini memfasilitasi transfer aman NFT antar *blockchain* dengan memastikan bahwa NFT tersebut terkunci selama proses transfer dan memberikan visibilitas transparan tentang kejadian transfer melalui *event* yang dicatat. Ini adalah komponen kunci dalam membangun aplikasi *interoperable* yang memungkinkan aset *digital* bergerak lintas ekosistem *blockchain*.



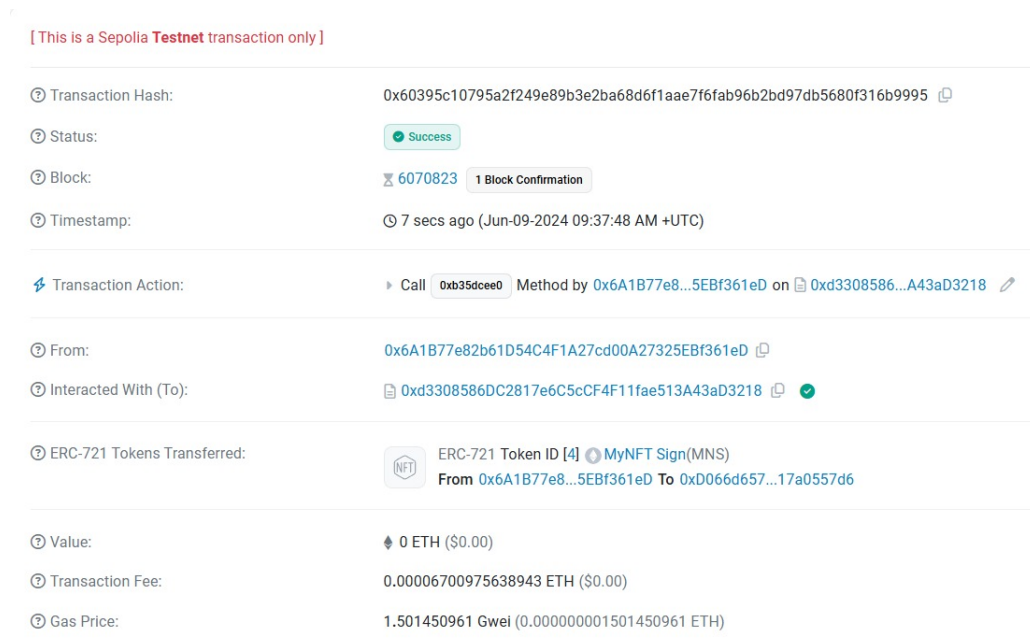
Gambar 4.13: Parameter *bridge transfer*

- pada gambar 4.13 terdapat 2 parameter yaitu "\_to" yang digunakan untuk menerima parameter *address* dari user B dan juga parameter "\_tokenId" untuk menerima argumen dari token NFT. Setelah dilakukan *transact* maka akan dilanjutkan ke pembayaran pada Metamask Wallet.



Gambar 4.14: Pembayaran pada Metamask Wallet

- setelah transaksi berhasil, *record* dari transaksi tersimpan pada *blockchain* yang dapat dilihat pada etherscan seperti pada gambar di bawah ini.

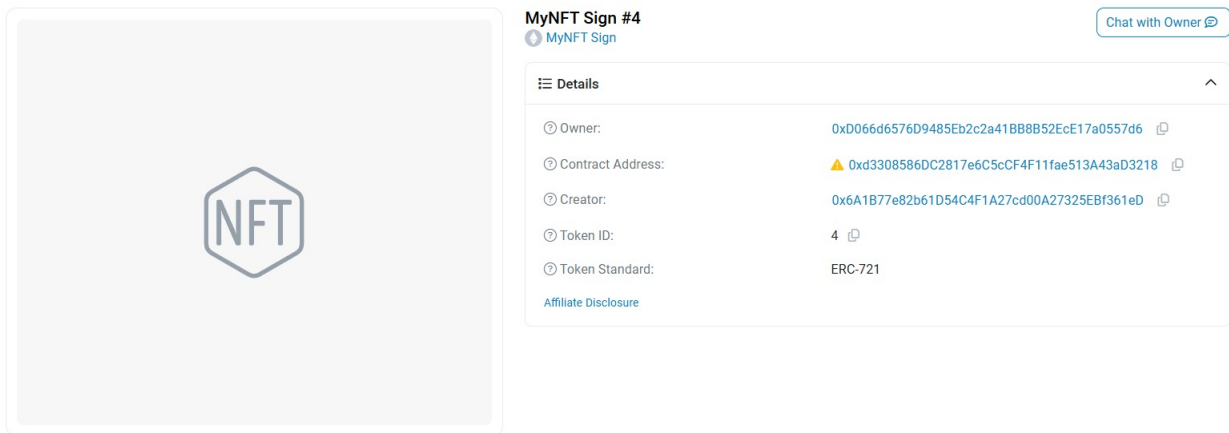


Gambar 4.15: Detail pada Etherscan

- pada etherscan juga terdapat detail pada NFT yang dapat dicek kepemilikannya. Jika dilihat dari gambar 4.10 *owner* dari NFT #4 adalah *address* dari *user A* yang berada pada *network Sepolia Ethereum Testnet*, lalu pada gambar di bawah ini setelah berhasil

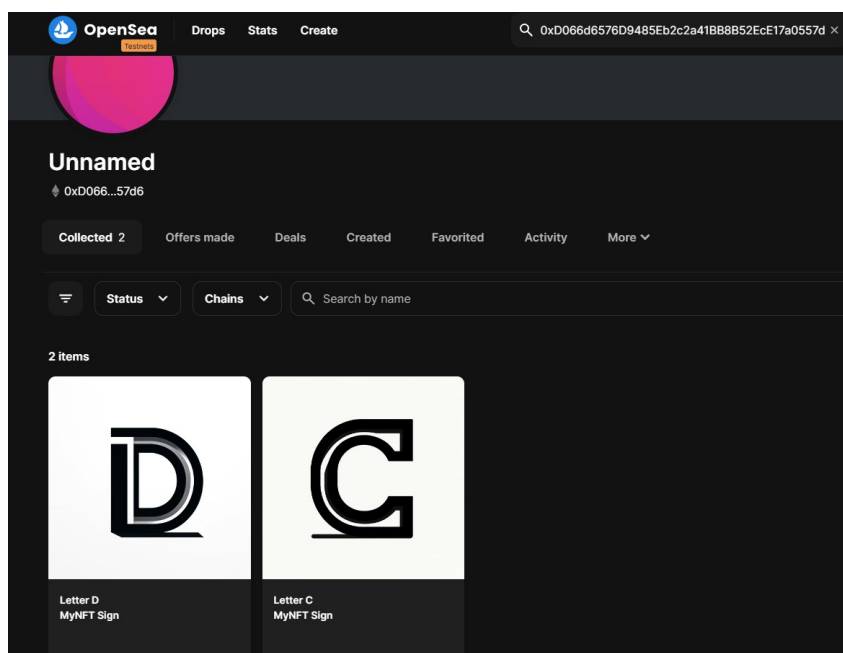


melakukan *bridge transfer* kepemilikannya berganti kepada *user B* yang berada pada *network BNB Chain Testnet*.



Gambar 4.16: Detail NFT pada Etherscan setelah *bridge transfer*

- pada *platform* OpenSea juga dapat dilihat pada akun milik *address user B* maka juga terdapat NFT yang telah terkirim ke alamatnya.



Gambar 4.17: NFT pada OpenSea *address B*

#### 4.2.2 Pengujian Integrasi *Smart Contract* Dengan Web3.0

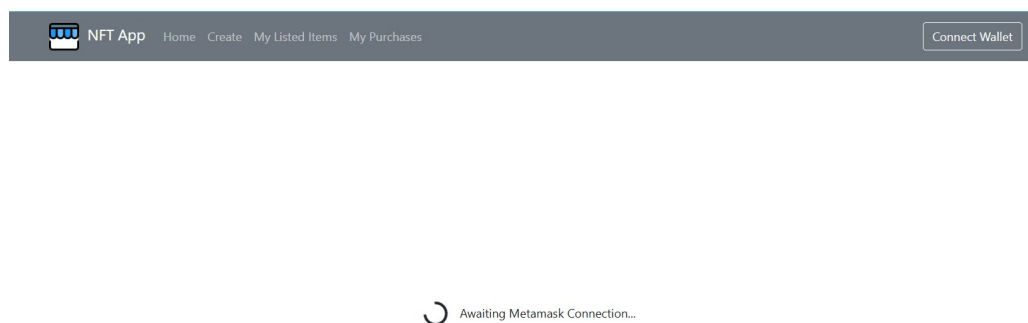
Ekspektasi pengujian dari integrasi sistem *smart contract* dengan web3.0 antara lain adalah sebagai berikut:

- *User* dapat melakukan integrasi akun Metamask Wallet dengan aplikasi web3.0
- *User* dapat melakukan *create listing* NFT untuk mengunggah NFT dari pemilik ke halaman *Home* dan ke halaman *My Listed Items*.

- *User* dapat melakukan minting pada NFT yang telah dibuat dan kemudian terlihat pada halaman *My Purchases*.
- *User* yang memiliki NFT tersebut dapat melakukan *transfer ownership* atau pindah kepemilikan dari alamat pengguna pemilik menuju ke alamat pengguna tujuan.
- *User* yang memiliki NFT tersebut dapat melakukan fungsi seperti tahapan dalam melakukan *bridge transfer* untuk melakukan pengiriman NFT dari alamat pengguna pada *network* lokal ke alamat pengguna tujuan yang berada pada *network binance smart chain* ataupun *sepolia ethereum*.

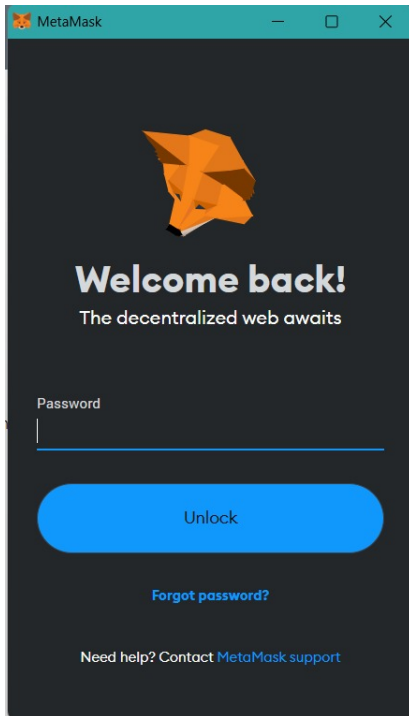
Berikut merupakan langkah-langkah dari pengujian integrasi *smart contract* dengan Web3.0 yang akan dilakukan:

- *User* akan mengakses web dengan menggunakan link yang dimunculkan secara lokal dengan melakukan run program lalu akan memunculkan link berupa *localhost* yang dapat diakses pada platform seperti Google Chrome, Mozilla Firefox, atau Microsoft Edge, dan platform pengakses website lainnya.

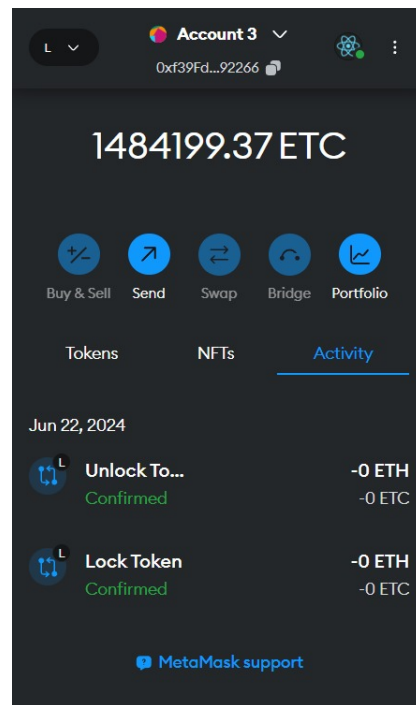


Gambar 4.18: Tampilan awal dari web

- Pada gambar 4.18 merupakan tampilan utama dari web, sebelum dapat melakukan *load* data pengguna harus melakukan koneksi dengan akun Metamask Wallet. Integrasi harus dilakukan agar pengguna dapat mengakses tampilan lanjutan pada web. Integrasi tersebut dapat dilakukan dengan menekan tombol *Connect Wallet* pada *navigation bar* di pojok kanan atas.



(a)



(b)

Gambar 4.19: Koneksi Metamask dengan Web

- Pada gambar 4.25 adalah proses ketika melakukan integrasi dengan Metamask Wallet pada web. Ketika pengguna menekan tombol *Connect Wallet* maka akan diarahkan kepada Metamask. Jika pengguna belum melakukan login pada Metamask maka pengguna akan disuruh login terlebih dahulu pada Metamask seperti pada gambar 4.19a. Jika pengguna telah melakukan login ke akun Metamask, maka tampilannya akan seperti gambar 4.19b yang di mana tampilan dari Metamask Wallet terdapat informasi *address* dari akun dan juga *balance* dari akun tersebut. Dikarenakan pada pengujian ini masih menggunakan *localhost* maka *balance* dari akun tersebut menggunakan milik Hardhat.

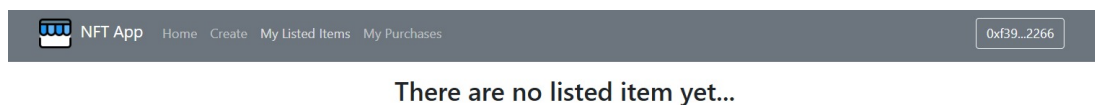


Gambar 4.20: Tampilan awal dari web

- Pada gambar 4.20 adalah tampilan dari *homepage* awal ketika pengguna menekan tombol *home* pada *navigation bar* atas. Pada web ini terdapat beberapa halaman yang dapat diakses dengan menekan tombol pada *navigation bar*, seperti *Home*, *Create*, *My Listed Items*, dan *My Purchase*. Tampilan-tampilan ini dapat diakses ketika pengguna telah melakukan koneksi web dengan Metamask Wallet, hal ini dapat dilihat pada *navigation bar* atas kanan terdapat *address* dari akun Metamask Wallet. Pada tampilan awal ini, pengguna dapat melihat NFT yang telah diunggah untuk dapat dilakukan pembelian atau *minting*. Tetapi karena belum terdapat NFT yang diunggah maka hanya akan menampilkan tulisan "Loading...".

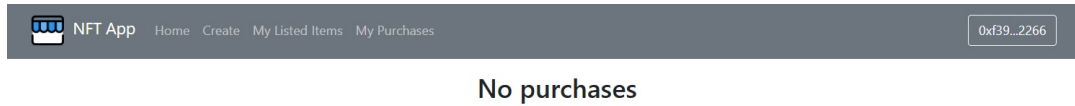
Gambar 4.21: Tampilan halaman *create*

- Pada gambar 4.21 adalah tampilan dari *create page*. Pada halaman ini pengguna dapat mengunggah sebuah NFT berupa gambar ke web yang kemudian jika diunggah akan muncul pada halaman *home* dan juga akan muncul pada halaman *my listed items*.



Gambar 4.22: Tampilan halaman *my listed items*

- Pada gambar 4.22 adalah tampilan dari *my listed items page*. Pada halaman ini pengguna dapat melihat NFT kepemilikannya yang telah diunggah pada ketika dia melakukan *create*. Tetapi dikarenakan belum melakukan pengunggahan NFT, maka halaman ini hanya akan memunculkan tulisan "*There are no listed item yet*".

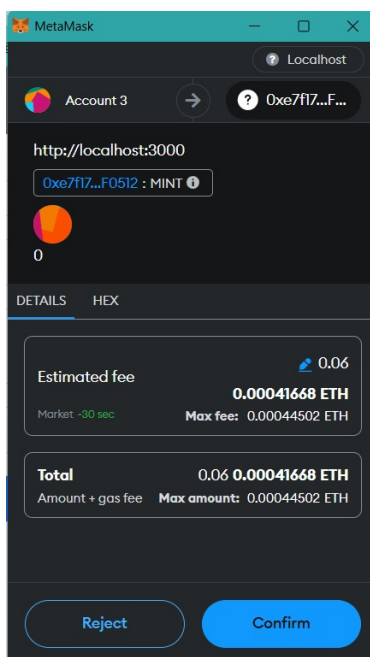


Gambar 4.23: Tampilan halaman *my purchases*

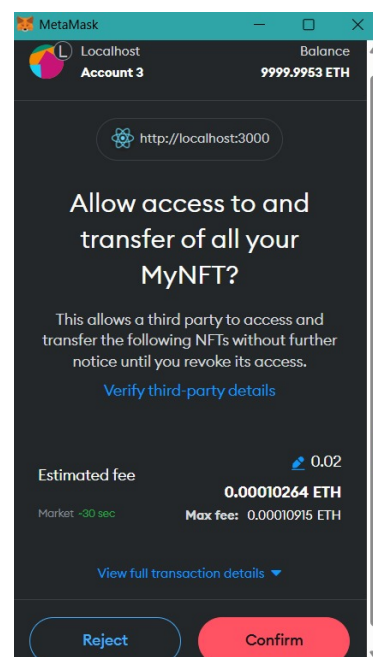
- Pada gambar 4.23, terlihat tampilan dari *my purchase page* yang dirancang khusus untuk memungkinkan pengguna melihat semua NFT yang telah mereka peroleh melalui pembelian di halaman utama platform. Setelah melakukan pembelian, NFT tersebut akan terdaftar di halaman ini, memberikan tampilan visual serta detail dari setiap NFT yang dimiliki. Lebih lanjut, halaman ini juga menyediakan fungsi *transfer ownership*, yang memungkinkan pemilik untuk mengalihkan kepemilikan NFT kepada pengguna lain dengan *address* yang berbeda. Fungsi ini sangat penting untuk mendukung fleksibilitas dan likuiditas dalam perdagangan NFT di pasar. Namun, jika pengguna belum melakukan pembelian apapun, halaman ini akan menampilkan pesan "*No purchases*", yang menandakan bahwa tidak ada NFT yang dapat ditampilkan atau ditransfer.
- Kemudian, kita akan menguji fungsionalitas sistem dengan fokus pada proses pengunggahan NFT, melakukan transaksi pembelian, serta mengirim NFT ke *address* lain. Dalam skenario pengujian ini, seluruh aktivitas akan dilakukan dalam jaringan yang sama, yakni *localhost*. Hal ini bertujuan untuk memastikan bahwa interaksi antar fungsi dalam *smart contract* berjalan dengan lancar dan tanpa adanya gangguan eksternal yang mungkin terjadi dalam jaringan publik. Pengujian di *localhost* memungkinkan kita untuk mengisolasi dan mengidentifikasi masalah fungsi dalam kondisi yang terkontrol sebelum memindahkannya ke jaringan tes yang lebih besar atau ke jaringan Ethereum utama. Selama proses ini, kita akan mengamati bagaimana sistem menangani proses-proses seperti penentuan kepemilikan, transaksi pembayaran, dan transfer kepemilikan antar pengguna, yang semuanya merupakan komponen penting dari aplikasi berbasis NFT. Pengujian ini juga mencakup verifikasi keamanan dan integritas data untuk memastikan bahwa tidak ada celah yang dapat dimanfaatkan oleh pengguna jahat dalam ekosistem.

Gambar 4.24: Melakukan pengunggahan NFT pada halaman *create*

- Pada tahap awal ini pengguna melakukan pengunggahan NFT pada halaman *create*. Pengguna memasukkan detail-detail dari NFT yang ingin diunggah seperti gambar, nama, deskripsi, dan juga harga dalam mata uang *ethereum*. Pada pengujian ini pengguna memasukkan NFT gambar "A" seperti pada pengujian *smart contract* sebelumnya. Setelah pengguna menekan tombol "Create & List NFT!" maka akan muncul *pop up window* dari Metamask Wallet yang digunakan untuk membayar *gas* atau fee dari melakukan eksekusi kode *smart contract*.



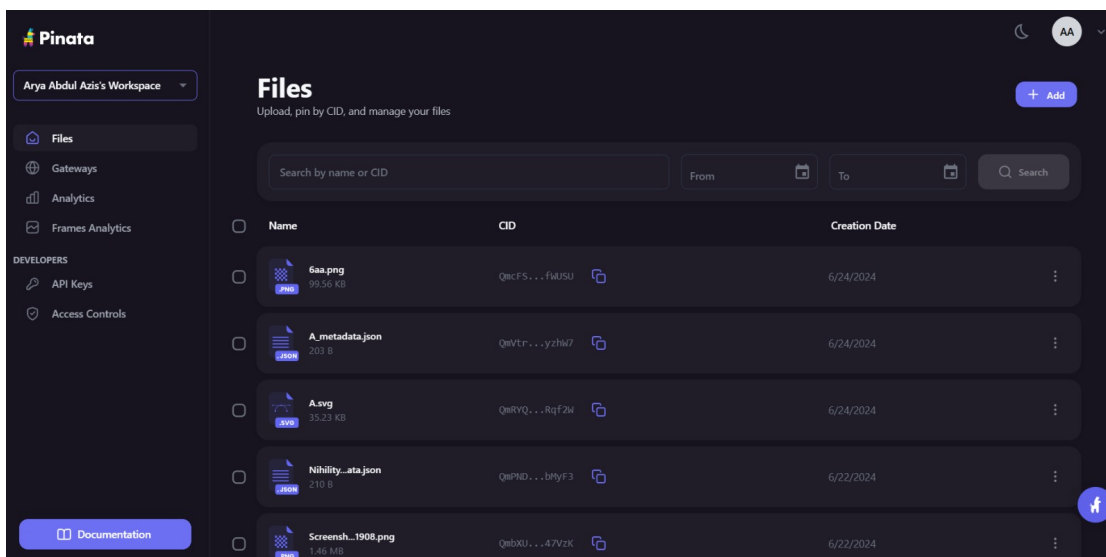
(a)



(b)

Gambar 4.25: Pembayaran gas IPFS menggunakan Metamask Wallet

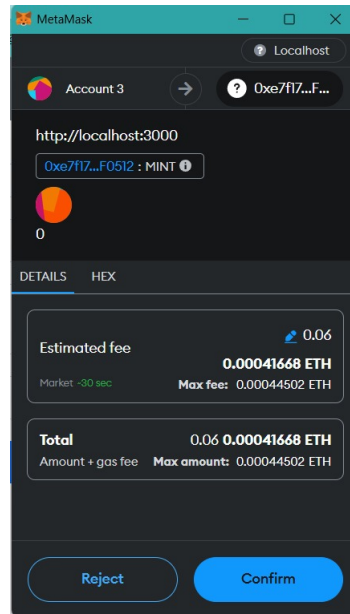
- Gambar 4.25a adalah pembayaran *gas* dari metamask wallet. Pembayaran *gas* tersebut terjadi karena pada *smart contract* terjadi proses pengunggahan gambar NFT ke platform penyedia IPFS. Pada web ini kita mengintegrasikan dengan platform bernama Pinata. Pinata sendiri merupakan platform penyedia servis IPFS. Kemudian juga terdapat konfirmasi pada gambar 4.25b, konfirmasi ini dilakukan karena melakukan integrasi dengan platform Pinata yang kita gunakan sebagai penyedia servis IPFS.
- Setelah berhasil melakukan konfirmasi pembayaran gas melalui MetaMask, NFT yang telah dibuat melalui halaman "Create" pada aplikasi akan diunggah ke platform Pinata. Platform Pinata ini berfungsi sebagai penyedia layanan penyimpanan dan pengelolaan file berbasis teknologi blockchain, yang menggunakan sistem InterPlanetary File System (IPFS) untuk memastikan keamanan dan ketahanan data.



Gambar 4.26: Data dari NFT ter-*upload* pada platform Pinata

- Dalam platform Pinata seperti pada gambar 4.26, setiap file yang diunggah, termasuk gambar untuk NFT, akan diberikan Content Identifier (CID) unik yang memudahkan pelacakan dan akses tanpa perlu mengkhawatirkan perubahan isi file, karena CID ini akan berubah jika konten file berubah. Hal ini sangat penting dalam ekosistem NFT di mana keaslian dan keunikan konten harus terjamin. Di Pinata, pengguna dapat dengan mudah melihat semua file yang telah diunggah, mengatur akses, dan memantau penggunaan dan analisis lalu lintas file. Fitur ini memungkinkan pembuat konten NFT untuk tidak hanya menyimpan aset digital mereka dengan aman tetapi juga mengelola distribusi mereka dengan lebih efektif dalam pasar digital. Keseluruhan proses ini mengintegrasikan teknologi blockchain dengan aplikasi web, memastikan bahwa setiap pembelian dan transfer NFT dapat dilacak dan diverifikasi secara transparan dan aman. Pinata tidak hanya menyediakan solusi penyimpanan yang efisien dan aman tetapi juga menawarkan antarmuka yang ramah pengguna, memudahkan para pengguna, terutama para pembuat NFT, untuk mengunggah dan mengelola aset digital mereka dengan mudah. Fitur pencarian dan pengelolaan file yang intuitif memungkinkan pengguna untuk mengakses dan mengatur koleksi NFT mereka tanpa hambatan, memastikan bahwa aset digital dapat dikelola dan diakses dengan cepat sesuai kebutuhan. Lebih lanjut, Pinata mendukung integrasi dengan

berbagai platform dan marketplace NFT, sehingga memperluas jangkauan dan visibilitas aset digital yang disimpan di dalamnya. Hal ini memberikan nilai tambah bagi para pembuat dan kolektor NFT dalam mendistribusikan dan memperdagangkan karya mereka di berbagai platform dengan mudah dan aman. Melalui Pinata, pembuat NFT dapat memastikan bahwa setiap aset digital tidak hanya aman tetapi juga siap untuk diintegrasikan dan digunakan dalam ekosistem blockchain yang lebih luas.

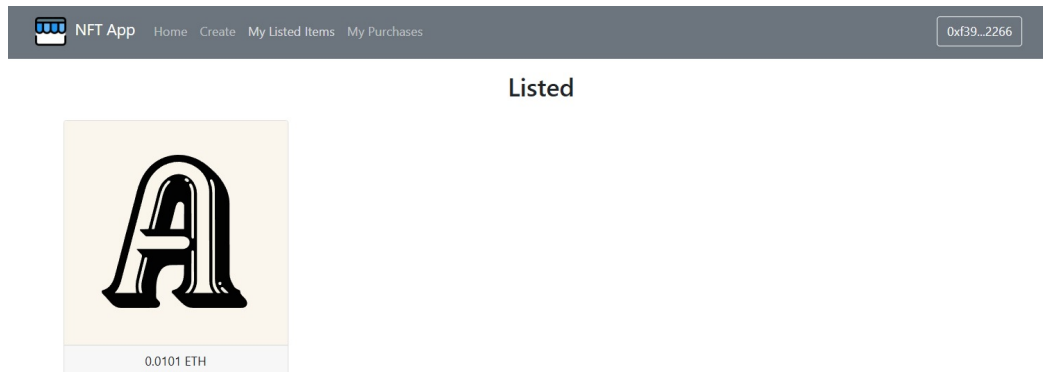


Gambar 4.27: Melakukan konfirmasi *minting* NFT pada Metamask Wallet

- Pada gambar 4.27, ditampilkan jendela konfirmasi MetaMask yang digunakan untuk memberikan persetujuan atas transaksi minting NFT. Dalam jendela ini, pengguna diminta untuk mengkonfirmasi atau menolak transaksi yang sedang diinisiasi dari aplikasi yang dihosting pada localhost:3000. Proses ini memastikan bahwa pengguna memahami dan menyetujui semua detail transaksi sebelum melanjutkan. Di dalam jendela konfirmasi, jumlah ETH yang akan ditransfer adalah 0, yang menandakan bahwa transaksi ini mungkin hanya melibatkan biaya gas, tanpa ada transfer dana tambahan yang terlibat. "Estimated fee" atau perkiraan biaya gas untuk transaksi ini ditampilkan, dengan jumlah minimal yang ditetapkan serta maksimum yang mungkin diperlukan, memberikan transparansi tentang biaya yang dapat dikeluarkan untuk memproses transaksi ini. Selain itu, MetaMask juga menyediakan informasi mengenai durasi estimasi untuk penyelesaian transaksi di pasar yang ditandai dengan "+30 sec", yang berarti transaksi diharapkan terjadi dalam waktu sekitar 30 detik.
- Setelah pengguna menyelesaikan proses minting NFT dan melakukan konfirmasi melalui MetaMask seperti yang dijelaskan sebelumnya, NFT yang baru dibuat akan muncul pada halaman "My Listed Items" dalam aplikasi. Halaman ini berfungsi sebagai galeri pribadi pengguna dimana semua NFT yang telah mereka buat dan daftarkan untuk dijual ditampilkan. Dalam contoh yang ditampilkan pada gambar, NFT dengan desain huruf "A" yang telah berhasil di-list tampak dengan harga yang tertera di bawah gambar yaitu 0.0101 ETH. Hal ini menandakan bahwa NFT ini siap untuk dibeli oleh pengguna lain. Setiap NFT yang terdaftar di halaman ini akan menampilkan gambar yang berkaitan dengan

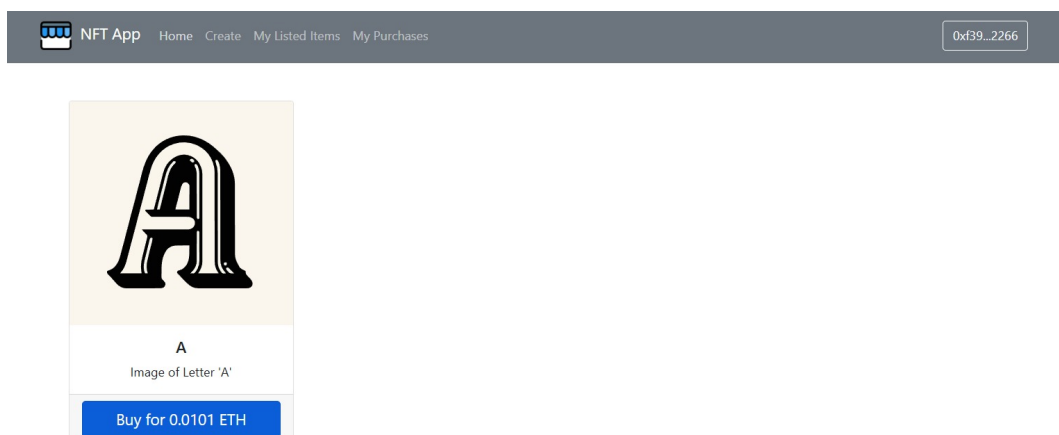


NFT tersebut, yang merupakan hasil unggahan pengguna ke IPFS, dan metadata lainnya seperti nama dan deskripsi juga akan ditampilkan jika pengguna telah menyertakannya saat proses minting.



Gambar 4.28: NFT yang telah di-*minting* terlihat pada halaman *My listed items*

- Setelah menyelesaikan proses pengunggahan dan minting NFT, NFT yang baru dibuat juga dapat dilihat di halaman utama (*home page*) web aplikasi NFT. Gambar yang disajikan menunjukkan NFT berupa gambar huruf "A", yang ditampilkan dengan label dan harga penjualan. Di halaman utama ini, NFT tidak hanya dipamerkan tetapi juga siap untuk dibeli oleh pengguna lain. Dalam tampilan ini, setiap NFT yang di-list akan dilengkapi dengan tombol "Buy" yang memungkinkan pengguna langsung membeli NFT tersebut. Tombol ini memfasilitasi pembelian langsung tanpa perlu navigasi ke halaman lain, memberikan kemudahan bagi pembeli untuk segera mengakuisisi aset digital yang diinginkan.



Gambar 4.29: NFT yang telah di-*minting* terlihat pada halaman *home*

- Pada gambar 4.29 ketika tombol tersebut diklik, transaksi akan diproses melalui MetaMask atau wallet digital lainnya yang telah disinkronkan dengan aplikasi. Halaman ini dirancang untuk memberikan pengalaman pengguna yang intuitif dan efisien, di mana pembeli dapat dengan cepat mengakses NFT yang mereka inginkan dan melihat informasi penting seperti nama, deskripsi, dan harga NFT. Ini tidak hanya memperkuat transparansi dan aksesibilitas dalam pasar NFT tetapi juga mendorong interaksi langsung dan spontan antara pembeli dan aset digital.

*[Halaman ini sengaja dikosongkan]*

## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

Berdasarkan hasil pengujian yang Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. sebagai berikut:

1. Pembuatan Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus.
2. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa.
3. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna.

#### **5.2 Saran**

Untuk pengembangan lebih lanjut pada Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. antara lain:

1. Memperbaiki Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus.
2. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa.
3. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna.

*[Halaman ini sengaja dikosongkan]*

## DAFTAR PUSTAKA

- Antonopoulos, A., Wood, G., & Wood, G. (2018). Mastering ethereum: Building smart contracts and dapps. *O'Reilly Media, Incorporated*.
- Bambacht, J., & Pouwelse, J. (2022). Web3: A decentralized societal infrastructure for identity, trust, money, and data.
- Chohan, U. W. (2022). Web 3.0: The future architecture of the internet?. *Web3.0*.
- Dannen, C. (2017). Introducing ethereum and solidity. *Foundations of Cryptocurrency and Blockchain Programming for Beginners*.
- Freni, P., Ferro, E., & Moncada, R. (2022). Tokenomics and blockchain tokens: A design-oriented morphological framework. *Blockchain: Research and Applications*.
- Gadekallu, T. R., Huynh-The, T., Wang, W., Yenduri, G., Ranaweera, P., Pham, Q.-V., da Costa, D. B., & Liyanage, M. (2022). Blockchain for the metaverse: A review. *Blockchain*.
- Khan, S. N., Loukil, F., Ghedira-Guegan, C., Benkhelifa, E., & Bani-Hani, A. (2023). Blockchain smart contracts: Applications, challenges, and future trends. *Blockchain smart contracts: Applications, challenges, and future trends*.
- Lu, S., Pei, J., Zhao, R., Yu, X., Zhang, X., Li, J., & Yang, G. (2023). Ccio: A cross-chain interoperability approach for consortium blockchains based on oracle. *Sensors*, 23(4). <https://doi.org/10.3390/s23041864>
- Malik, N., Wei, Y. ", Appel, G., & Luo, L. (2023). Blockchain technology for creative industries: Current state and research opportunities. *International Journal of Research in Marketing*.
- Pillai, B., Biswas, K., & Muthukkumarasamy, V. (2020). Cross-chain interoperability among blockchain-based systems using transactions. *The Knowledge Engineering Review*, 35, e23. <https://doi.org/10.1017/S0269888920000314>
- Ray, P. P. (2023). Web3: A comprehensive review on background, technologies, applications, zero-trust architectures, challenges and future directions. *Internet of Things and Cyber-Physical Systems*, 3, 213–248. <https://doi.org/https://doi.org/10.1016/j.iotcps.2023.05.003>
- Steichen, M., Fiz, B., Norvill, R., Shbair, W., & State, R. (2018). Blockchain-based, decentralized access control for ipfs. *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*.
- Tjokrosetio, D. (2022). The complete guide to blockchain. *Tempo Publishing*.
- Wang, Q., Li, R., Wang, Q., & Chen, S. (2021). Non-fungible token (nft): Overview, evaluation, opportunities and challenges. *NFT's*.
- Zheng, Z., Xie, S., Dai, H.-N., Chen, W., hen, X., Weng, J., & Imran, M. (2020). An overview on smart contracts: Challenges, advances and platforms. future generation computer systems. *An overview on smart contracts: Challenges, advances and platforms. Future Generation Computer Systems*.

*[Halaman ini sengaja dikosongkan]*

## BIOGRAFI PENULIS



Arya Abdul Azis, lahir pada Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.



*[Halaman ini sengaja dikosongkan]*