



TUGAS AKHIR - EC234801

INTEROPERABILITAS NFT BERBASIS BLOCKCHAIN MENGGUNAKAN 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 MENGGUNAKAN 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
Juli, 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
July, 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, berfsifat 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, July 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, July 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 MENG-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

Emergensi *Non-Fungible Token* (NFT) sebagai komponen kunci dalam ekonomi digital telah memicu minat signifikan terhadap potensinya untuk merevolusi berbagai industri, mulai dari hiburan hingga manajemen identitas digital. Aspek fundamental dalam memaksimalkan kegunaan NFT melibatkan kepastian operasionalnya lintas platform *blockchain*, yang diatasi dengan konsep interoperabilitas. Tesis ini menyajikan studi komprehensif dan implementasi interoperabilitas NFT berbasis *blockchain* menggunakan *smart contract* dalam kerangka Web3.0. Riset ini terutama fokus pada pengembangan *smart contract* yang tidak hanya mendukung fitur standar dari NFT tetapi juga memfasilitasi interaksi lintas rantai mereka. Dengan memanfaatkan *blockchain* Ethereum dan menggunakan standar ERC-721, pekerjaan ini mendirikan protokol yang kuat untuk pembuatan, transaksi, dan manajemen NFT yang menjamin interoperabilitas yang aman, transparan, dan aspek kunci termasuk desain aplikasi terdesentralisasi (DApp) yang berinteraksi dengan *smart contract* untuk mencetak, mengelola, dan mentransfer NFT melintasi batas *blockchain*. Implementasi menunjukkan praktikalitas dan efisiensi sistem yang dalam lingkungan testnet terkontrol. Pekerjaan masa depan mungkin mengeksplorasi solusi penskalaan, fitur keamanan yang ditingkatkan, dan integrasi platform *blockchain* tambahan untuk memperluas jangkauan dan aplikabilitas NFT yang interoperabel di alam semesta yang berkembang dari Web3.0.

Kata Kunci: *Non-Fungible Token* (NFTs), *Blockchain*, *Interoperability*, *Smart Contracts*, Web3.0, Ethereum

[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

The emergence of Non-Fungible Tokens (NFTs) as a key component of the digital economy has sparked significant interest in their potential to revolutionize various industries, from art and entertainment to digital identity management. A fundamental aspect of maximizing the utility of NFTs involves ensuring their seamless operation across different blockchain platforms, which is addressed by the concept of interoperability. This thesis presents a comprehensive study and implementation of blockchain-based NFT interoperability using smart contracts within the framework of Web3.0. The research primarily focuses on developing a smart contract architecture that not only supports the standard features of NFTs but also facilitates their cross-chain interactions. By leveraging the Ethereum blockchain and utilizing the ERC-721 standard, this work establishes a robust protocol for NFT creation, transaction, and management that ensures secure, transparent, and efficient interoperability. Key aspects include the design of a decentralized application (DApp) that interacts with smart contracts to mint, manage, and transfer NFTs across blockchain boundaries. The implementation demonstrates the practicality and efficiency of the proposed system in a controlled testnet environment. Future work might explore scaling solutions, enhanced security features, and the integration of additional blockchain platforms to extend the reach and applicability of interoperable NFTs in the expanding universe of Web3.0.

Keywords: Non-Fungible Tokens (NFTs), Blockchain Interoperability, Smart Contracts, Web3.0, Ethereum

[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, Juli 2024

Arya Abdul Azis

[Halaman ini sengaja dikosongkan]

DAFTAR ISI

ABSTRAK	i
ABSTRACT	iii
KATA PENGANTAR	v
DAFTAR ISI	vii
DAFTAR GAMBAR	xii
DAFTAR TABEL	xiii
1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Permasalahan	2
1.3 Batasan Masalah atau Ruang Lingkup	2
1.4 Tujuan	2
1.5 Manfaat	2
2 TINJAUAN PUSTAKA	5
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 Ethereum	6
2.1.3 <i>Ethereum Virtual Machine (EVM)</i>	7
2.1.4 Token	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
3	METODOLOGI	11
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	14
3.2.3	Hardhat	14
3.2.4	Ganache	14
3.2.5	InterPlanetary File System	14
3.2.6	React.js	15
3.2.7	Web3.js	15
3.2.8	Ethers.js	15
3.2.9	Etherscan	15
3.2.10	Token ERC-721	16
3.3	Metode Yang Digunakan	16
3.3.1	Smart Contract dan Interoperabilitas	16
3.3.2	Pengembangan Frontend	20
3.3.3	Integrasi Smart Contract dengan Frontend	20
3.3.4	Pengujian	21
4	HASIL DAN PEMBAHASAN	23
4.1	Hasil	23
4.1.1	Web Interface	23
4.1.2	<i>Smart Contract</i>	26
4.1.3	<i>Non-Fungible Token (NFT)</i>	27
4.2	Pengujian	27
4.2.1	Pengujian Fitur <i>Smart Contract</i>	27
4.2.2	Pengujian Integrasi <i>Smart Contract</i> Dengan Web3.0	33

4.2.3 Pengujian Fitur Transfer <i>Ownership</i> Secara <i>Localhost</i> Dan Secara Lin-	45
tas <i>Blockchain</i>	
5 PENUTUP	57
5.1 Kesimpulan	57
5.2 Saran	57
DAFTAR PUSTAKA	59
BIOGRAFI PENULIS	61

[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	16
4.1	Tampilan interface	23
4.2	Tampilan awal dari web	24
4.3	Tampilan halaman <i>create</i>	24
4.4	Tampilan halaman <i>my listed items</i>	25
4.5	Tampilan halaman <i>my purchases</i>	25
4.6	Detail transaksi pada etherscan	26
4.7	NFT terunggah pada Opensea	27
4.8	Riwayat transaksi minting pada Metamask Wallet	28
4.9	Detail transaksi pada Etherscan	29
4.10	Detail NFT pada Etherscan	30
4.11	NFT pada OpenSea	30
4.12	Detail fungsi <i>Lock Token</i> pada Etherscan	31
4.13	Parameter <i>bridge transfer</i>	31
4.14	Pembayaran pada Metamask Wallet	32
4.15	Detail pada Etherscan	32
4.16	Detail NFT pada Etherscan setelah <i>bridge transfer</i>	33
4.17	NFT pada OpenSea <i>address</i> B	33
4.18	Tampilan awal dari web	34
4.19	Koneksi Metamask dengan Web	35
4.20	Melakukan pengunggahan NFT pada halaman <i>create</i>	36
4.21	Pembayaran gas IPFS menggunakan Metamask Wallet	36
4.22	Data dari NFT ter- <i>upload</i> pada platform Pinata	37
4.23	Melakukan konfirmasi <i>minting</i> NFT pada Metamask Wallet	38
4.24	NFT yang telah di- <i>minting</i> terlihat pada halaman <i>My listed items</i>	39
4.25	NFT yang telah di- <i>minting</i> terlihat pada halaman <i>home</i>	40
4.26	Akun Metamask kedua	41

4.27	Proses pembelian NFT	42
4.28	Tampilan <i>home</i> setelah NFT dibeli	42
4.29	Tampilan <i>Listed items</i> ketika melakukan pembelian	43
4.30	Tampilan halaman <i>my purchases</i> ketika sudah melakukan pembelian	44
4.31	Tampilan halaman <i>my purchases</i> ketika sudah melakukan pembelian	45
4.32	Detail akun ketiga pada Metamask Wallet	46
4.33	Tampilan NFT yang siap dilakukan pindah kepemilikan	47
4.34	Tampilan NFT yang siap dilakukan pindah kepemilikan	48
4.35	<i>Alert</i> "NFT transferred successfully!" ketika NFT berhasil dipindahkan	48
4.36	<i>Alert</i> "NFT transferred successfully!" ketika NFT berhasil dipindahkan	49
4.37	Console Transaksi Pada <i>Hardhat Node</i>	49
4.38	Tampilan halaman <i>my purchases</i> pada akun ketiga	50
4.39	Detail Akun Untuk Pengujian Lintas Rantai	50
4.40	Memasukkan <i>address</i> tujuan untuk <i>bridge transfer</i>	51
4.41	Konfirmasi pembayaran gas dari fungsi <i>lock token</i> di Metamask Wallet	52
4.42	Konfirmasi pembayaran gas dari fungsi <i>bridge transfer</i> di Metamask Wallet	52
4.43	Kepemilikan NFT "C" Berganti Menjadi Akun <i>Network BNB Smart Chain</i>	53
4.44	Konfirmasi Pembayaran Gas <i>Unlock Token</i>	53
4.45	Hasil Akhir Dari Transfer Lintas Rantai	54
4.46	<i>Block Hash</i> dari setiap transaksi	54

DAFTAR TABEL

4.1	Informasi Akun Testing Beda <i>Network</i>	28
4.2	Informasi Akun Testing Sesama <i>Network</i>	28

[Halaman ini sengaja dikosongkan]

BAB I

PENDAHULUAN

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 [1]. 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 [2].

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 mengan-dalkan 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 dian-dalkan, 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 [3].

Interoperabilitas yang rendah ini menciptakan silo dalam industri blockchain, yang pada gilirannya mengurangi potensi inovasi dan pertukaran nilai secara efisien. Dalam konteks Web 3.0 — era internet yang lebih terdesentralisasi dan pengguna-berkuasa — interoperabilitas menjadi penting untuk memastikan bahwa teknologi blockchain dapat mendukung sebuah ekosistem digital yang benar-benar terbuka dan terhubung. Oleh karena itu, tugas akhir ini bertujuan untuk mengembangkan solusi yang memungkinkan interoperabilitas NFT lintas blockchain, sehingga memfasilitasi transfer dan pengakuan aset digital di berbagai platform blockchain. Solusi ini di-harapkan tidak hanya akan mengatasi hambatan teknis saat ini tetapi juga akan membuka pintu untuk adopsi yang lebih luas dari teknologi NFT dalam berbagai aplikasi praktis di Web 3.0, dari seni digital hingga real estate dan beyond.

Blockchain dan Smart Contract menjadi fondasi penting dalam mewujudkan visi 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 perantara. Kombinasi dari teknologi-teknologi ini dapat membentuk fondasi untuk era baru di dunia digital, di mana interaksi lebih aman, transparan, dan tanpa hambatan [4].

1.2 Permasalahan

Berdasarkan latar belakang di atas, maka dapat dirumuskan masalah pada tugas akhir ini adalah bagaimana mengembangkan solusi interoperabilitas untuk *Non-Fungible Tokens* (NFT) yang memungkinkan transfer lintas berbagai platform *blockchain* dengan efisien dan aman.

1.3 Batasan Masalah atau Ruang Lingkup

1. Penelitian ini akan menggunakan platform *blockchain* Ethereum, mengingat Solidity merupakan bahasa pemrograman khusus untuk Ethereum.
2. Penelitian akan difokuskan pada NFT dalam bentuk gambar.
3. 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 sistem *smart contract* yang dapat berinteraksi dan berkomunikasi dengan *blockchain* pada *network* lain tanpa hambatan.
2. Mengimplementasikan sistem *smart contract* yang berinteroperabilitas dengan web3.0.

1.5 Manfaat

1. Riset ini memfasilitasi pertukaran dan pemanfaatan token non-fungible (NFT) secara mulus lintas berbagai *network blockchain*. Dengan mengatasi masalah interoperabilitas, ini memungkinkan pemilik NFT menggunakan aset mereka di berbagai lingkungan aplikasi terdesentralisasi tanpa terikat pada satu jaringan *blockchain* saja.
2. Penggunaan *smart contract* dalam mengelola transaksi NFT mengenalkan pendekatan standarisasi yang meningkatkan keamanan, transparansi, dan efisiensi.
3. Dengan mengintegrasikan NFT dengan Web3.0, tugas akhir ini mendorong batasan cara aplikasi terdesentralisasi dapat beroperasi, mengarah pada platform yang lebih otonom, tanpa kepercayaan, dan diatur oleh pengguna.

Sistematika Penulisan

Laporan penelitian tugas akhir ini terbagi menjadi 5 bab yaitu:

1. BAB I Pendahuluan

Bab ini berisi latar belakang, permasalahan, batasan masalah atau ruang lingkup, tujuan, manfaat, dan juga sistematika penulisan dari penelitian ini.

2. BAB II Tinjauan Pustaka

Bab ini berisi penelitian-penelitian terdahulu yang relevan dengan penelitian ini. Dalam

bab ini juga dijelaskan teori-teori yang mendukung penelitian ini.

3. BAB III Metodologi

Bab ini berisi keperluan yang digunakan dalam penelitian ini, termasuk metode, alat yang digunakan serta urutan dalam pelaksanaan penelitian.

4. BAB IV Hasil dan Pembahasan

Bab ini berisi hasil dari pengujian-pengujian yang sudah dilakukan dan hasil yang didapatkan dari setiap pengujian yang sudah dilakukan

5. BAB V Penutup

Bab ini berisi kesimpulan serta hasil akhir dari penelitian ini, dan juga menjawab permasalahan yang sudah dijelaskan di bab pendahuluan.

[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. [5]

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 [3].

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 [6].

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. [7]

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 [8]. 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 ke langkaannya, menjadikannya alat penting dalam menetapkan harga bagi produk dengan hak kekayaan intelektual di dunia digital [1].

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* [9].

Smart Contract dapat dikembangkan dan diterapkan di berbagai platform *blockchain* (mis-

alnya, 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* [2].

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. [10].

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. [11]

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. [12]

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. [13]

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 [14]. 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*.

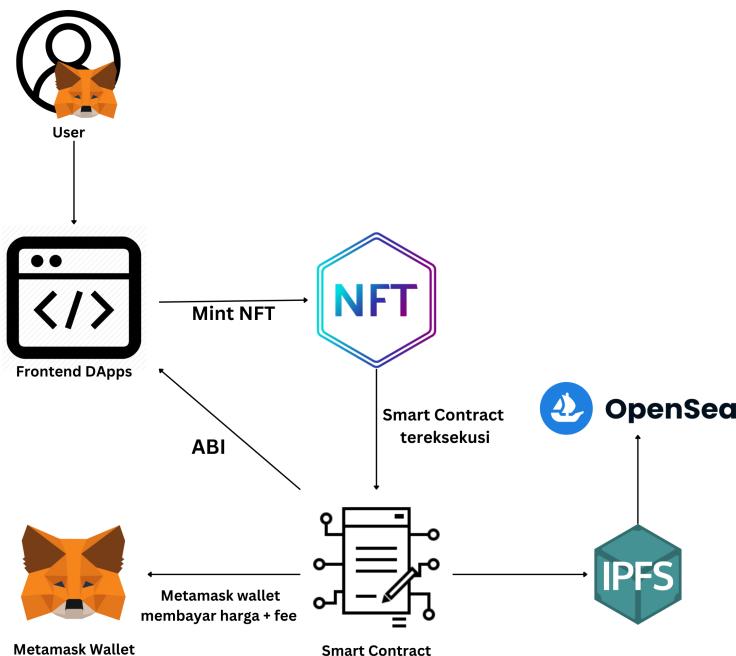
Kerangka CCIP meningkatkan sistem *oracle* tradisional dengan mengintegrasikan metode enkripsi simetris dan asimetris untuk mengamankan data di lintas rantai. Ini memastikan bahwa data sensitif yang ditransmisikan melintasi jaringan *blockchain* mempertahankan integritas dan kerahasiaan. Efektivitas 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 [15].

BAB III

METODOLOGI

3.1 Deskripsi Sistem

3.1.1 Arsitektur Sistem

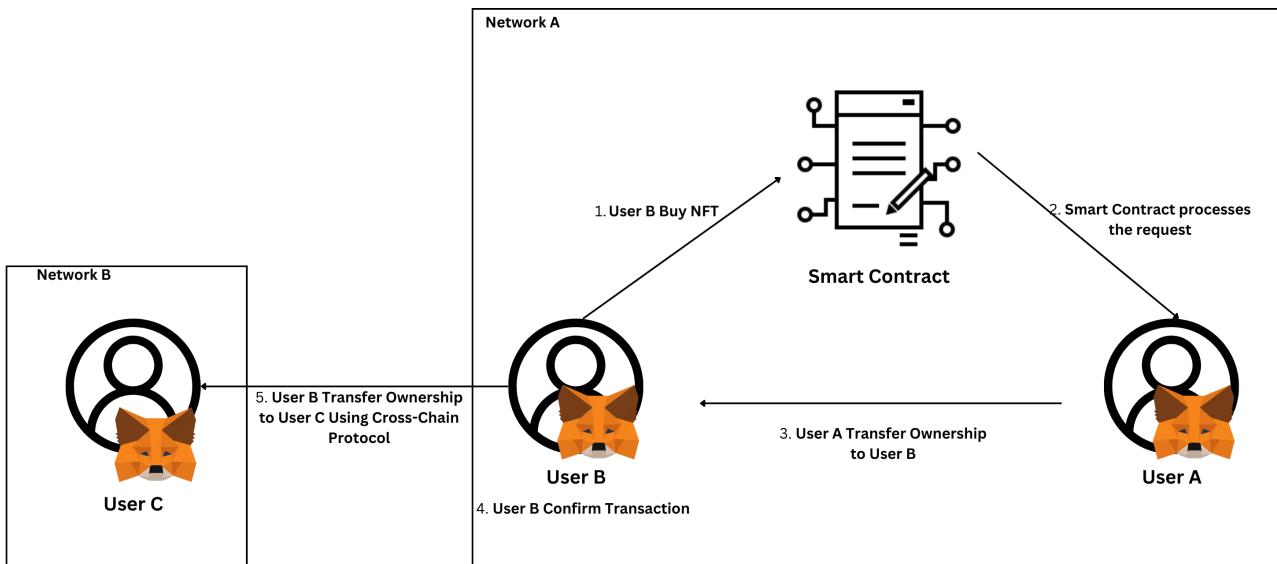


Gambar 3.1: Arsitektur Sistem

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.

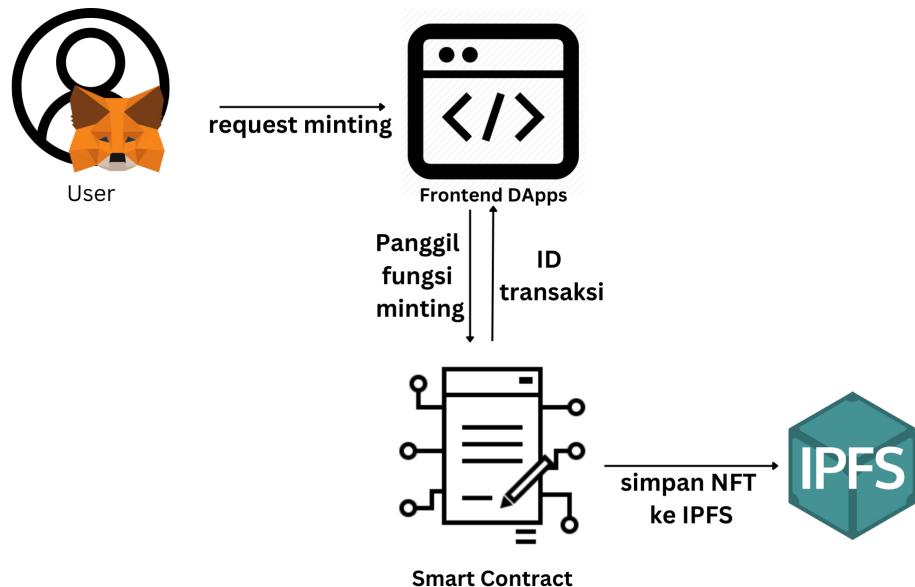
3.1.3 Metadata

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 Naturretnya 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 secara khusus dirancang untuk mengembangkan *smart contracts* pada platform *blockchain* Ethereum. Bahasa ini menawarkan sintaks yang dioptimalkan untuk menangani operasi kontrak cerdas, termasuk dukungan untuk variabel, fungsi, dan struktur kontrol yang kompleks, yang semuanya vital dalam menciptakan aplikasi terdesentralisasi yang efisien dan aman. Selain itu, Solidity dilengkapi dengan berbagai fitur keamanan yang mencegah kerentanan umum seperti serangan *reentrancy* dan *overflows*. Keunggulan utama dari Solidity adalah kemampuannya untuk dijalankan di atas Ethereum Virtual Machine (EVM), lingkungan runtime yang menjadikannya kompatibel dengan berbagai jenis *blockchain* yang mengadopsi standar EVM. Penggunaan Solidity membuka peluang bagi pengembang untuk menerapkan logika kontrak yang rumit yang bisa interaksi langsung dengan *blockchain*, memberikan tingkat transparansi dan keandalan yang tinggi dalam transaksi digital.

3.2.2 Metamask

MetaMask adalah dompet digital yang dirancang khusus untuk memudahkan pengguna dalam mengelola aset kripto seperti Ether (ETH) dan berinteraksi dengan aplikasi terdesentralisasi (dApps) pada jaringan Ethereum. Fungsinya tidak hanya terbatas pada penyimpanan dan pengelolaan kripto, tetapi juga sebagai jembatan yang menghubungkan pengguna dengan ekosistem Ethereum. Sebagai ekstensi browser, MetaMask memungkinkan pengguna untuk melakukan transaksi kripto secara langsung dari browser web mereka tanpa perlu menggunakan perangkat lunak tambahan. MetaMask juga mendukung jaringan *blockchain* lain yang kompatibel dengan Ethereum, memungkinkan pengguna untuk berpartisipasi dalam berbagai aktivitas *blockchain* seperti game berbasis *blockchain*, pasar seni digital, dan platform keuangan terdesentralisasi (DeFi). Dengan MetaMask, pengguna dapat dengan mudah beralih antara jaringan ini, menjelajahi dunia kripto dengan lebih aman dan nyaman.

3.2.3 Hardhat

Hardhat adalah kerangka kerja pengembangan *smart contract* yang dirancang khusus untuk memudahkan proses pengembangan, pengujian, dan penyebaran *smart contract*. Alat ini menyediakan serangkaian fitur yang memungkinkan pengembang *blockchain* untuk bekerja lebih efisien. Dengan menggunakan Hardhat, pengembang dapat menjalankan lingkungan Ethereum lokal yang memudahkan proses pengujian dan debugging. Hardhat mendukung tugas otomatisasi yang kompleks dan mengintegrasikan stack perangkat lunak yang luas, termasuk plugin dan skrip yang dapat disesuaikan untuk meningkatkan fungsionalitas dan efisiensi. Framework ini juga memfasilitasi interaksi yang lebih mulus dengan jaringan Ethereum publik dan testnet, memungkinkan pengembang untuk memperoleh feedback yang cepat dan efektif tentang cara kerja *smart contract* mereka di dalam kondisi jaringan yang sebenarnya.

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 pemenuaan 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*.

3.2.9 Etherscan

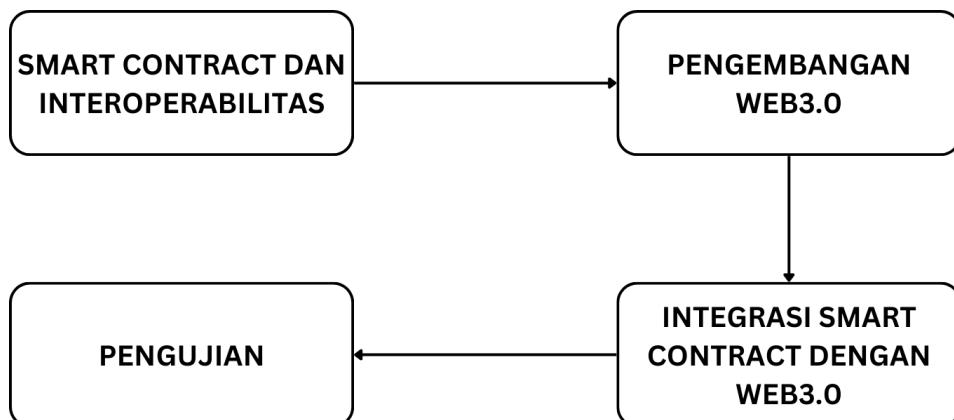
Etherscan adalah platform analitis berbasis web yang memungkinkan pengguna untuk melacak transaksi Ethereum, *smart contract*, dan status *blockchain* secara real-time. Platform

ini memberikan wawasan mendetail tentang semua aktivitas yang terjadi di jaringan Ethereum, termasuk informasi tentang blok terbaru, transaksi yang sedang berlangsung, dan token yang beredar. Etherscan juga digunakan secara luas oleh pengembang untuk memeriksa dan memverifikasi kode sumber *smart contract*, yang merupakan aspek penting dalam memastikan transparansi dan keamanan dalam ekosistem Ethereum. Dengan antarmuka yang mudah digunakan, Ether-scan telah menjadi alat esensial bagi siapa saja yang terlibat dalam industri *blockchain* Ethereum untuk memonitor, memverifikasi, dan menganalisis transaksi dan kontrak.

3.2.10 Token ERC-721

ERC-721 adalah standar untuk mewakili aset non-fungible (NFT) di *blockchain* Ethereum. Standar ini memungkinkan representasi unik dari aset digital seperti karya seni, barang koleksi, dan item dalam game dalam bentuk token digital. Setiap token ERC-721 memiliki identifikasi unik dan dapat memiliki metadata yang terkait dengannya, yang membedakannya dari token lain di *blockchain*. Fitur ini memfasilitasi pembuatan, perdagangan, dan penjualan aset digital dengan cara yang terdesentralisasi dan transparan. ERC-721 juga mendukung fungsi yang memungkinkan pemilik token untuk mentransfer, menyetujui, atau mengakses informasi tentang kepemilikan token, yang krusial untuk aplikasi di ekosistem Web3 dan ekonomi digital.

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 memperoses 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) MENGEKSPOR 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 fleksibilitas 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 koneksi. 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.

[Halaman ini sengaja dikosongkan]

BAB IV

HASIL DAN PEMBAHASAN

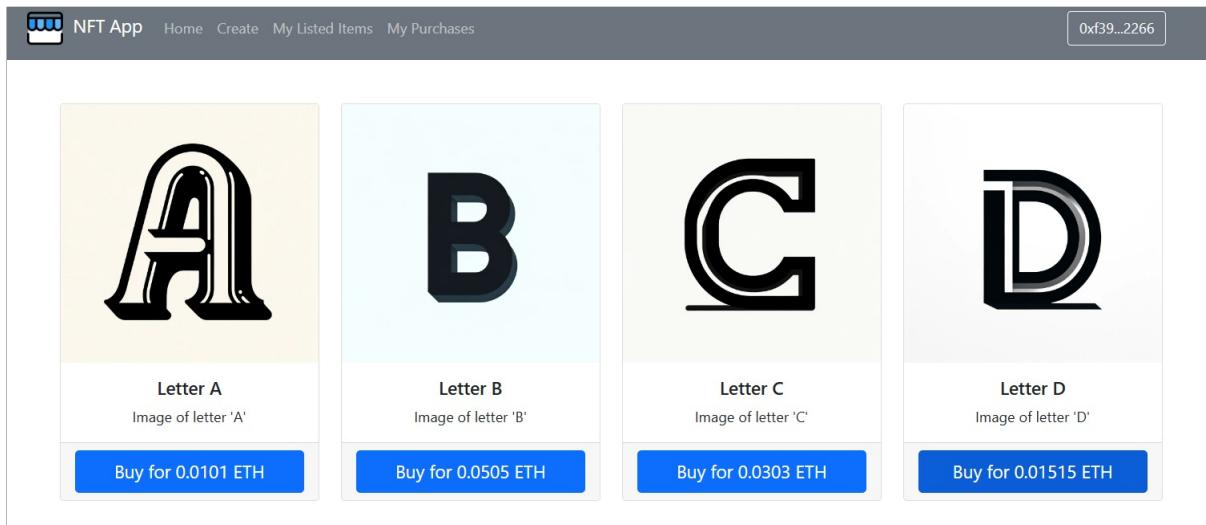
Pada subbab pertama pada bab ini akan dipaparkan hasil akhir dari penggerjaan tugas akhir Interoperabilitas NFT Berbasis *Blockchain* Menggunakan *Smart Contract* pada WEB3.0. Pada subbab selanjutnya akan dipaparkan mengenai hasil pengujian dari hasil tugas akhir untuk memastikan bahwa sistem yang dikembangkan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

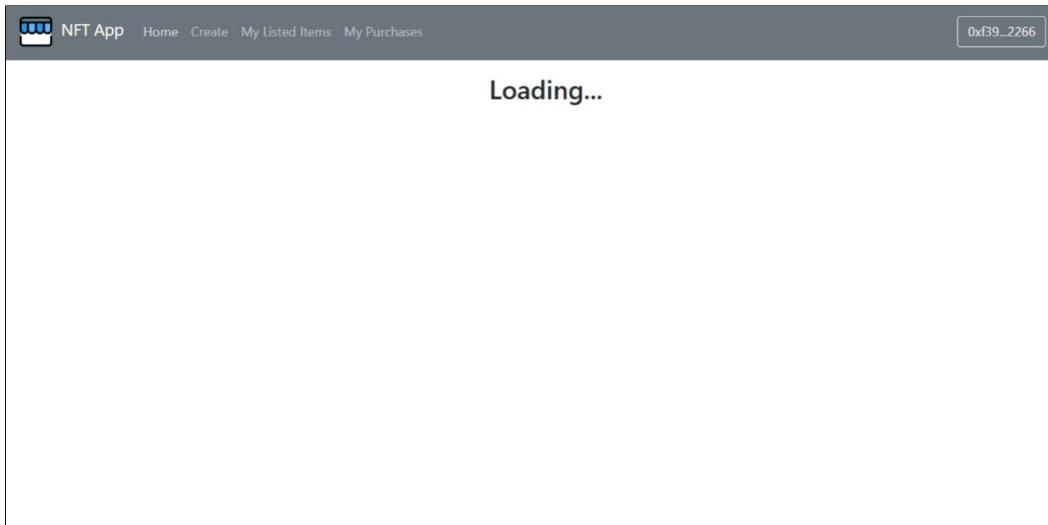
Tampilan awal web *interface*, menampilkan daftar koleksi NFT yang dapat di-*minting*. Nantinya NFT tersebut dapat dimiliki oleh *address* yang melakukan pertama kali *minting* dan juga kepemilikan dari NFT tersebut dapat diberikan ke *address* lain.



Gambar 4.1: Tampilan interface

Pada halaman dashboard awal, pengguna diharuskan untuk menekan tombol Connect Wallet, yang akan memicu proses koneksi dengan Metamask Wallet. Setelah berhasil terhubung, pengguna akan mendapatkan address yang bersifat unik. Alamat ini tidak hanya penting untuk keperluan identifikasi pengguna di dalam jaringan, tetapi juga berfungsi sebagai kunci akses untuk melakukan minting token NFT. Dengan demikian, address tersebut menjadi sangat krusial karena merupakan representasi digital dari identitas pengguna dalam ekosistem blockchain, memungkinkan mereka untuk bertransaksi, mengakses, dan mengelola aset digital mereka dengan aman. Proses ini memastikan bahwa setiap interaksi yang dilakukan melalui smart contract

untuk NFT tercatat secara transparan dan dapat diverifikasi, memberikan keamanan dan kepercayaan dalam setiap transaksi.



Gambar 4.2: Tampilan awal dari web

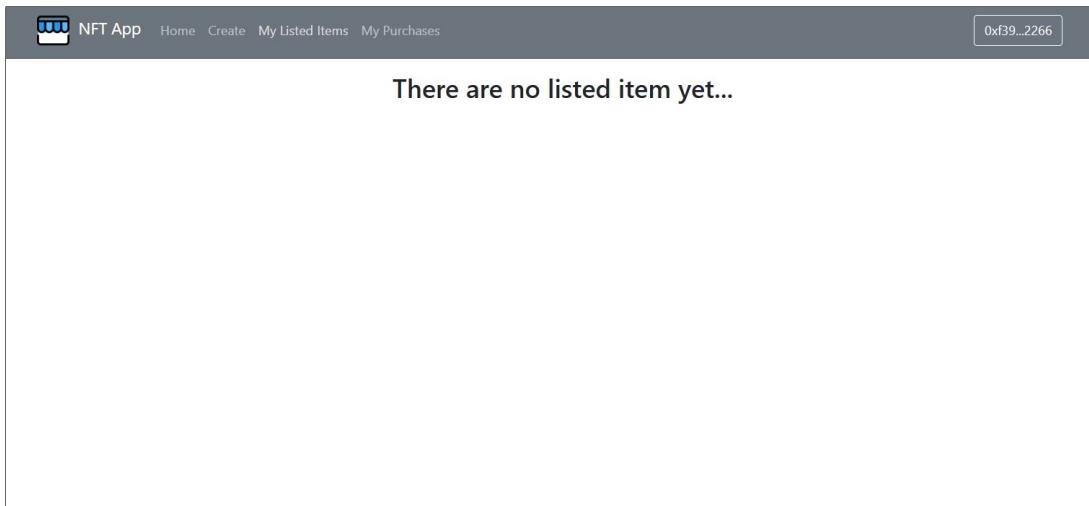
Pada gambar 4.2 adalah tampilan dari *homepage* awal ketika pengguna menekan tombol *home* pada *navigation bar* atas. Pada web ini terdapat beberapa halaman yang dapat diakses dengan menekan tombol pada *navigation bar*, seperti *Home*, *Create*, *My Listed Items*, dan *My Purchase*. Tampilan-tampilan 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...*".

A screenshot of a web browser displaying the "Create" page of the NFT application. The header bar is identical to the homepage, with "NFT App" and the address "0xf39...2266". The main content area contains four input fields: a file upload field labeled "Choose File" with "No file chosen", a text input field labeled "Name", a text input field labeled "Description", and a text input field labeled "Price in ETH". Below these fields is a large blue button with the text "Create & List NFT!". The entire form is contained within a white rectangular area.

Gambar 4.3: Tampilan halaman *create*

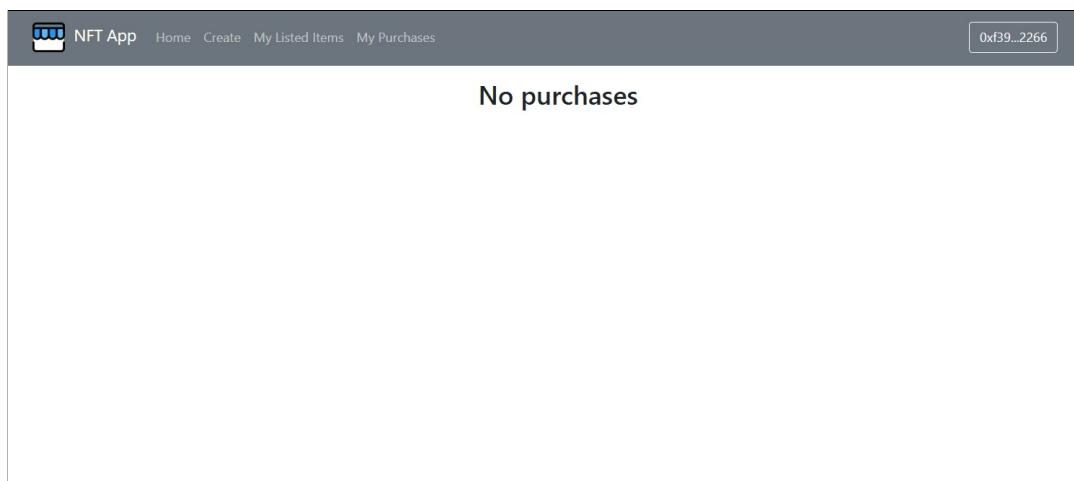
Pada gambar 4.3 adalah tampilan dari *create page*. Pada halaman ini pengguna dapat

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 4.4: Tampilan halaman *my listed items*

Pada gambar 4.4 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 4.5: Tampilan halaman *my purchases*

Pada gambar 4.5, 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.z

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 menyediakan 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.

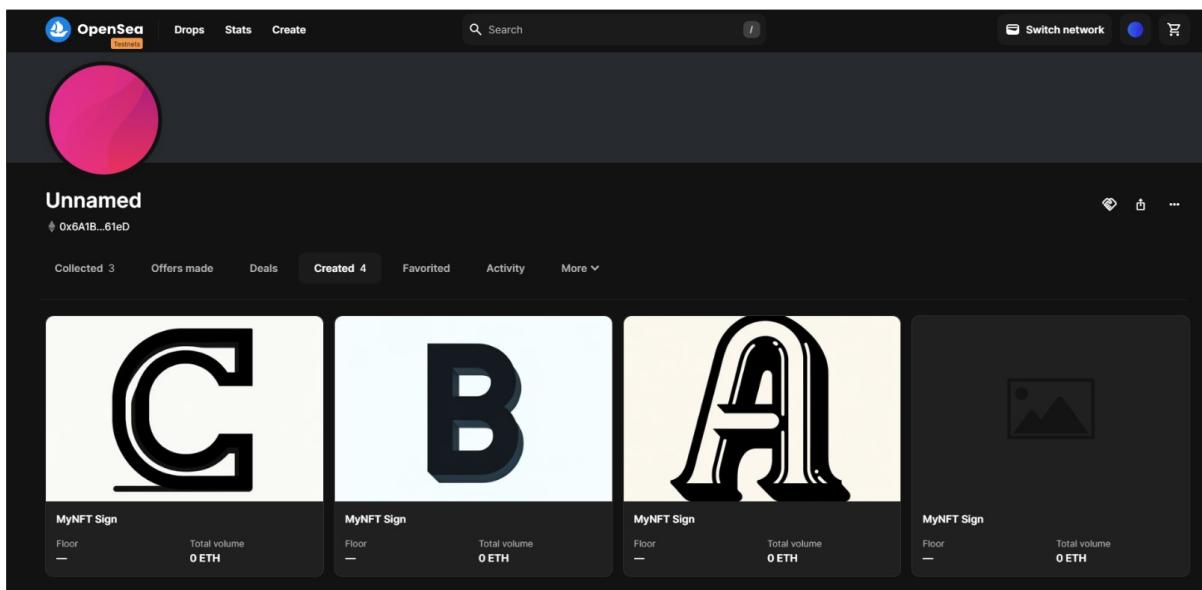
Proses *deploy* dilakukan *smart contract* akan di-deploy ke beberapa jaringan *blockchain* namun dikarenakan dalam pengujian ini yang digunakan adalah sepolia *testnet* maka yang digunakan hanyalah dari jaringan sepolia *testnet*. Ketika melakukan proses *deployment* akan dikenakan *gas* atau fee yang 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]	
② Transaction Hash:	0xf0573d90b5124a32e8947f37de9925fc6cca6fb9031132c87e3aa357a2042e8 🔗
② Status:	Success
② Block:	6062201 3560 Block Confirmations
② Timestamp:	12 hrs ago (Jun-08-2024 03:04:36 AM +UTC)
⚡ Transaction Action:	Call 0x60806040 Method by 0x6A1B77e8...5EBf361eD 🔗
② From:	0x6A1B77e82b61D54C4F1A27cd00A27325EBf361eD 🔗
② To:	[0x8d01bb70bc8bb35e9c39d7f76b370144ca235a5a Created] 🔗 ✓
② Value:	0 ETH (\$0.00)
② Transaction Fee:	0.003570899471483106 ETH (\$0.00)
② 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 penggerjaan 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 Opensea

4.2 Pengujian

Untuk memastikan bahwa sistem interoperabilitas *Smart Contract* berjalan sesuai dengan yang direncakan 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*'.

Dalam penelitian ini, dilakukan pengujian interoperabilitas dan transfer ownership NFT menggunakan dua skenario berbeda yang dicatat dalam dua tabel. Tabel pertama, yang tercantum sebagai "Informasi Akun Testing Beda Network," digunakan untuk menguji kemampuan

smart contract dalam mengirimkan NFT antar dua jaringan blockchain yang berbeda dari Sepolia Ethereum Testnet ke BNB Chain Testnet. Setiap akun diwakili oleh alamat unik yang beroperasi di network yang telah ditentukan, memungkinkan pengujian transfer aset digital lintas lingkungan blockchain yang berbeda.

Tabel 4.1: Informasi Akun Testing Beda *Network*

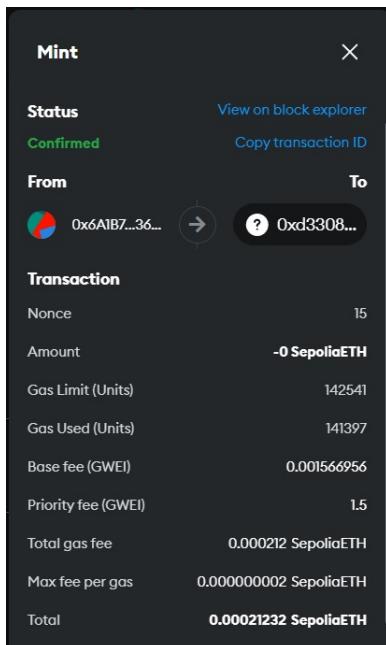
Akun	Address	Network
A	0x6A1B77e82b61D54C4F1A27cd00A27325EBf361eD	Sepolia Ethereum Testnet
B	0xD066d6576D9485Eb2c2a41BB8B52EcE17a0557d6	BNB Chain Testnet

Tabel 4.2: Informasi Akun Testing Sesama *Network*

Akun	Address	Network
Pertama	0xf39Fd6e51aad88F6F4ce6aB8827279cffFb92266	Localhost
Kedua	0x70997970C51812dc3A010C7d01b50e0d17dc79C8	Localhost
Ketiga	0x3C44CdDdB6a900fa2b585dd299e03d12FA4293BC	Localhost

Sementara itu, tabel kedua, "Informasi Akun Testing Sesama Network," berfokus pada pengujian internal dalam satu jaringan, yaitu Localhost. Pengujian ini melibatkan beberapa akun yang berinteraksi dalam satu jaringan yang sama untuk mengevaluasi proses transfer ownership NFT antar akun dalam satu ekosistem yang sama.

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



Gambar 4.8: Riwayat transaksi minting pada Metamask Wallet

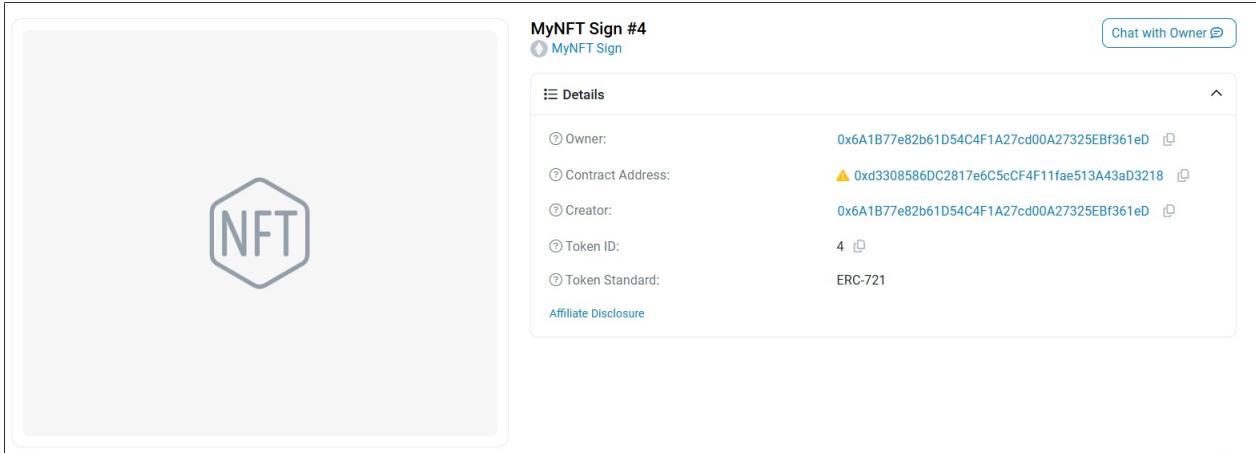
- User A akan melakukan *minting* pada NFT dengan menggunakan MetaMask wallet, yang terhubung langsung ke Ethereum *blockchain* melalui jaringan *Sepolia Testnet*. Transaksi *minting* ini mengkonfirmasi kreasi token baru di *blockchain*, yang dicatat dengan detail transaksi seperti nonce, gas limit, dan biaya transaksi yang terdiri dari base fee dan priority fee. Proses ini mencatat NFT sebagai aset digital yang unik di *blockchain*, memberikan hak kepemilikan digital kepada User A. Transaksi semacam ini memastikan keamanan dan transparansi dalam pencatatan kepemilikan aset digital, sekaligus mengintegrasikan NFT ke dalam ekosistem yang lebih luas di mana aset ini dapat diperdagangkan atau digunakan dalam aplikasi lain di masa depan.
- Tambahan dari gambar 4.9, detail pada Etherscan memperlihatkan informasi kunci dari transaksi NFT yang dilakukan. *Transaction hash* yang ditampilkan berfungsi sebagai pengenal unik transaksi, menunjukkan keberhasilannya pada *blockchain*. Tanggal dan waktu transaksi juga tercatat, memberikan data penting mengenai kapan transaksi tersebut terjadi. Token yang terlibat adalah tipe ERC-721, standar untuk NFT yang memungkinkan representasi unik aset digital di Ethereum. Transaksi ini mencerminkan proses *minting*, di mana NFT baru dibuat dan terdaftar di *blockchain*, dan kemudian dialihkan dari alamat nol (menunjukkan penciptaan) ke alamat pemilik baru, yang dalam kasus ini adalah akun pertama yang melakukan *minting*. Selain itu, biaya transaksi atau *gas fee* yang relatif rendah menandakan efisiensi dalam pemrosesan transaksi tersebut di jaringan.

[This is a Sepolia Testnet transaction only]	
② Transaction Hash:	0x2d6a8d1647bbc5a014b36b9e00b76699647471019f206f97a4f56e673c2d9c2e
② Status:	Success
② Block:	6070424 2 Block Confirmations
② Timestamp:	26 secs ago (Jun-09-2024 08:14:12 AM +UTC)
⚡ Transaction Action:	Call Mint Function by 0x6A1B77e8...5EBf361eD on 0xd3308586...A43aD3218
② From:	0x6A1B77e82b61D54C4F1A27cd00A27325EBf361eD
② Interacted With (To):	0xd3308586DC2817e6C5cCF4F11fae513A43aD3218 ✓
② ERC-721 Tokens Transferred:	 ERC-721 Token ID [4] MyNFT Sign(MNS) From 0x00000000...00000000 To 0x6A1B77e8...5EBf361eD
② Value:	0 ETH (\$0.00)
② Transaction Fee:	0.000212317062877532 ETH (\$0.00)
② Gas Price:	1.501566956 Gwei (0.00000001501566956 ETH)

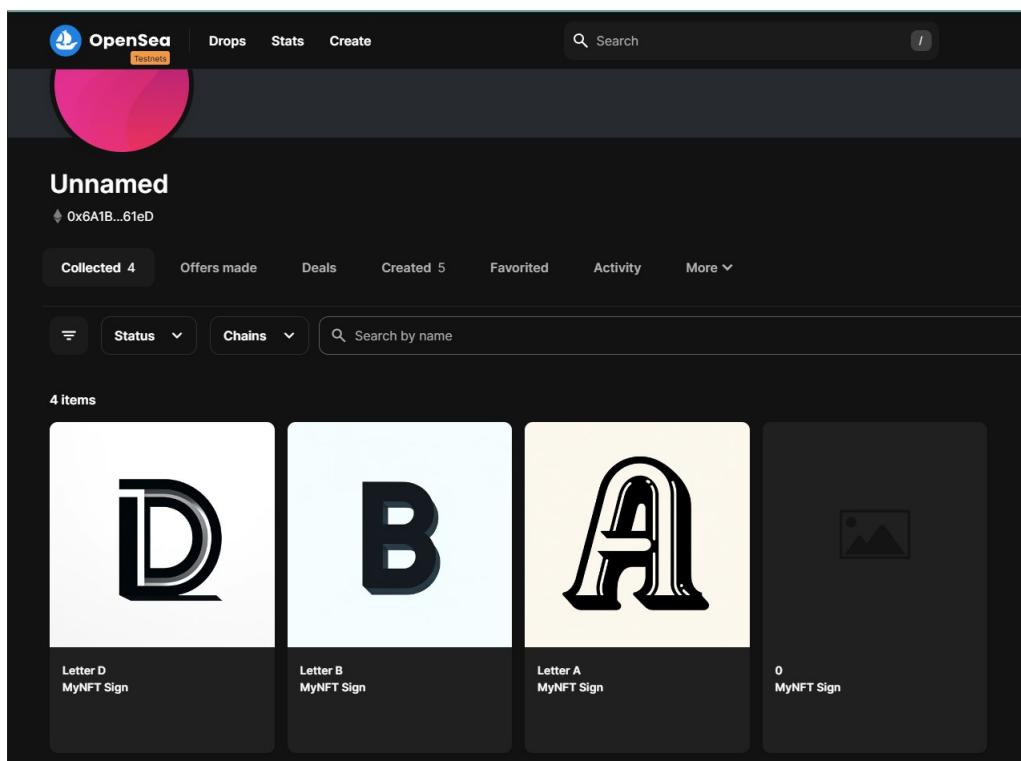
Gambar 4.9: Detail transaksi pada Etherscan

- Pada gambar 4.10, detail lebih lanjut mengenai NFT yang telah di-*mint* juga dapat dilihat di Etherscan, seperti yang ditampilkan dalam gambar terlampir. Halaman Etherescan ini menyediakan data menyeluruh tentang NFT tersebut, termasuk alamat pemilik, alamat kontrak, dan pembuatnya. Setiap aspek dicatat dengan presisi untuk memastikan transparansi dan kemampuan dilacak dalam jaringan blockchain. Secara khusus, NFT ini, yang diidentifikasi sebagai "MyNFT Sign #4", ditampilkan dengan ID token uniknya dan memastikan kepatuhan pada standar token ERC-721, yang menegaskan sifat non-fungible-nya. Tingkat detail ini penting untuk memverifikasi keaslian dan riwayat

kepemilikan aset digital, membuat platform seperti Etherscan menjadi alat yang tak tergantikan dalam pengelolaan dan pertukaran NFT. Alat-alat ini juga meningkatkan keamanan transaksi digital dengan menyediakan jejak audit yang dapat diandalkan untuk setiap aset.



Gambar 4.10: Detail NFT pada Etherscan



Gambar 4.11: NFT pada OpenSea

- Setelah proses *minting* berhasil, seperti pada gambar 4.11 NFT yang telah di-*minting* dapat dilihat pada platform OpenSea, yang merupakan pasar digital untuk aset kripto dan NFT. Dalam kasus pengujian ini, NFT yang telah di-minting adalah "Letter D", ditampilkan bersama dengan NFT lainnya yang telah dibuat oleh pengguna yang sama. Halaman OpenSea menampilkan berbagai detail seperti nama NFT, gambar, dan informasi

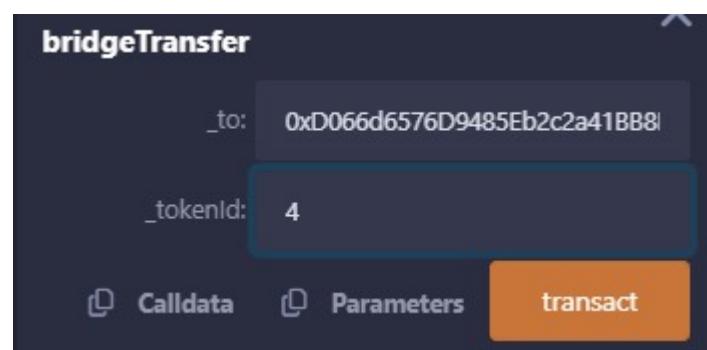
lain yang relevan. Pengguna dapat melihat koleksi lengkap yang telah mereka mint atau beli. Setiap NFT ditampilkan dengan *thumbnail* yang jelas, dan ketika di-klik, pengguna dapat melihat detail lebih lanjut seperti riwayat transaksi dan keaslian NFT tersebut. Ini memungkinkan pengguna untuk tidak hanya melihat koleksi mereka tetapi juga untuk berinteraksi dengan pasar, membuat penawaran, atau menjual NFT mereka.

- 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.

[This is a Sepolia Testnet transaction only]	
② Transaction Hash:	0xa162b2a1099373c83ad3d8302c114ef8edb651e3fc7641a5586f527eb3925f61 ⓘ
② Status:	Success
② Block:	6070693 2 Block Confirmations
② Timestamp:	24 secs ago (Jun-09-2024 09:10:36 AM +UTC)
⚡ Transaction Action:	Call Lock Token Function by 0x6A1B77e8...5EBf361eD on 0xd3308586...A43aD3218 ⓘ
② From:	0x6A1B77e82b61D54C4F1A27cd00A27325EBf361eD ⓘ
② To:	0xd3308586DC2817e6C5cCF4F11fae513A43aD3218 ⓘ ⓘ
② Value:	0 ETH (\$0.00)
② Transaction Fee:	0.000075337467521175 ETH (\$0.00)
② Gas Price:	1.501494121 Gwei (0.000000001501494121 ETH)

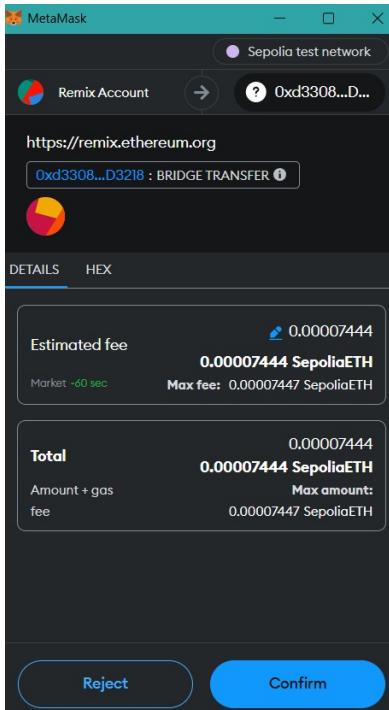
Gambar 4.12: Detail fungsi *Lock Token* pada Etherscan

- 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.



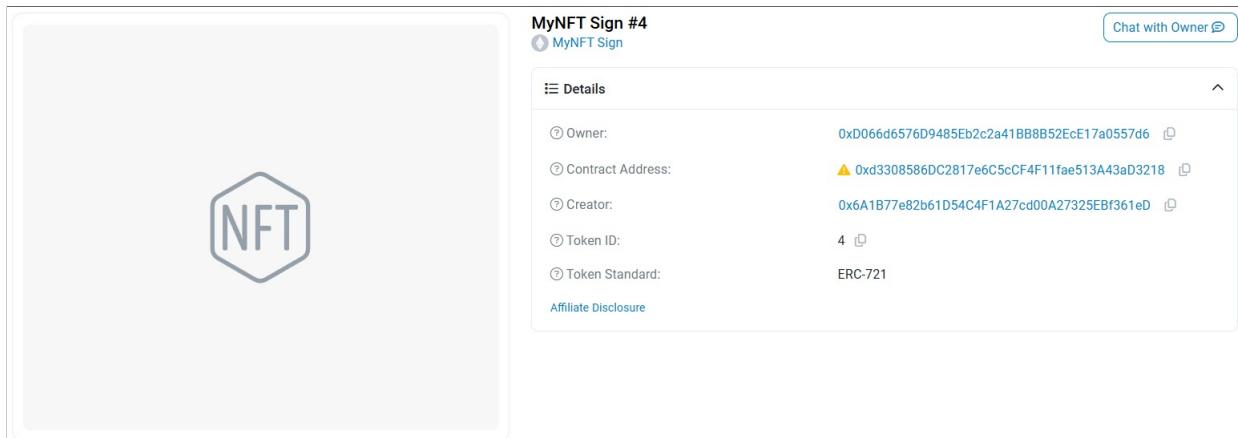
Gambar 4.14: Pembayaran pada Metamask Wallet

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

[This is a Sepolia Testnet transaction only]	
② Transaction Hash:	0x60395c10795a2f249e89b3e2ba68d6f1aae7f6fab96b2bd97db5680f316b9995 ⓘ
② Status:	Success
② Block:	6070823 1 Block Confirmation
② Timestamp:	7 secs ago (Jun-09-2024 09:37:48 AM +UTC)
⚡ Transaction Action:	Call <code>0xb35dce0</code> Method by <code>0x6A1B77e8...5EBf361eD</code> on <code>0xd3308586...A43aD3218</code> ⚡
② From:	0x6A1B77e82b61D54C4F1A27cd00A27325EBf361eD ⓘ
② Interacted With (To):	0xd3308586DC2817e6C5cCF4F11fae513A43aD3218 ⓘ ✓
② ERC-721 Tokens Transferred:	 ERC-721 Token ID [4] ⓘ MyNFT Sign(MNS) From <code>0x6A1B77e8...5EBf361eD</code> To <code>0xD066d657...17a0557d6</code>
② Value:	0 ETH (\$0.00)
② Transaction Fee:	0.00006700975638943 ETH (\$0.00)
② Gas Price:	1.501450961 Gwei (0.000000001501450961 ETH)

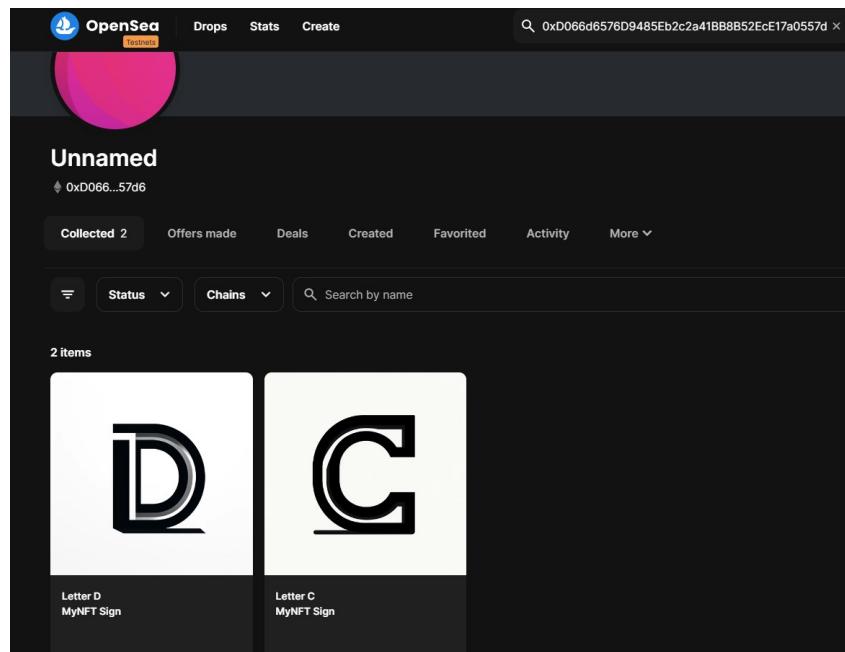
Gambar 4.15: Detail pada Etherscan

- 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 *addrress user B* maka juga terdapat NFT yang telah terkirim ke alamatnya.



Gambar 4.17: NFT pada OpenSea *address B*

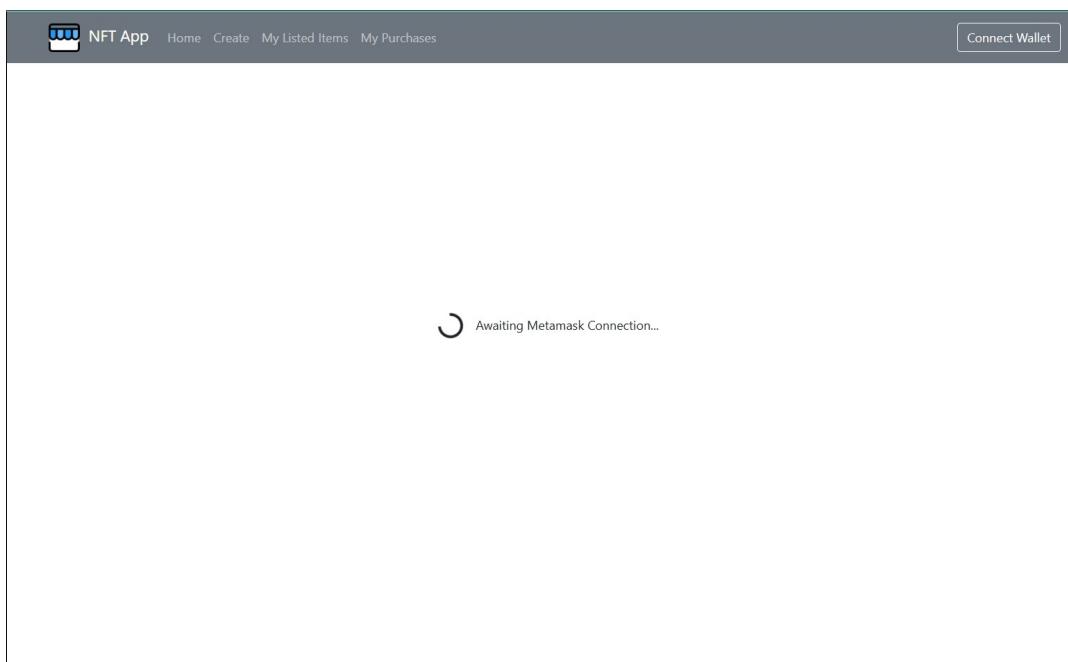
4.2.2 Pengujian Integrasi *Smart Contract* Dengan Web3.0

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*.

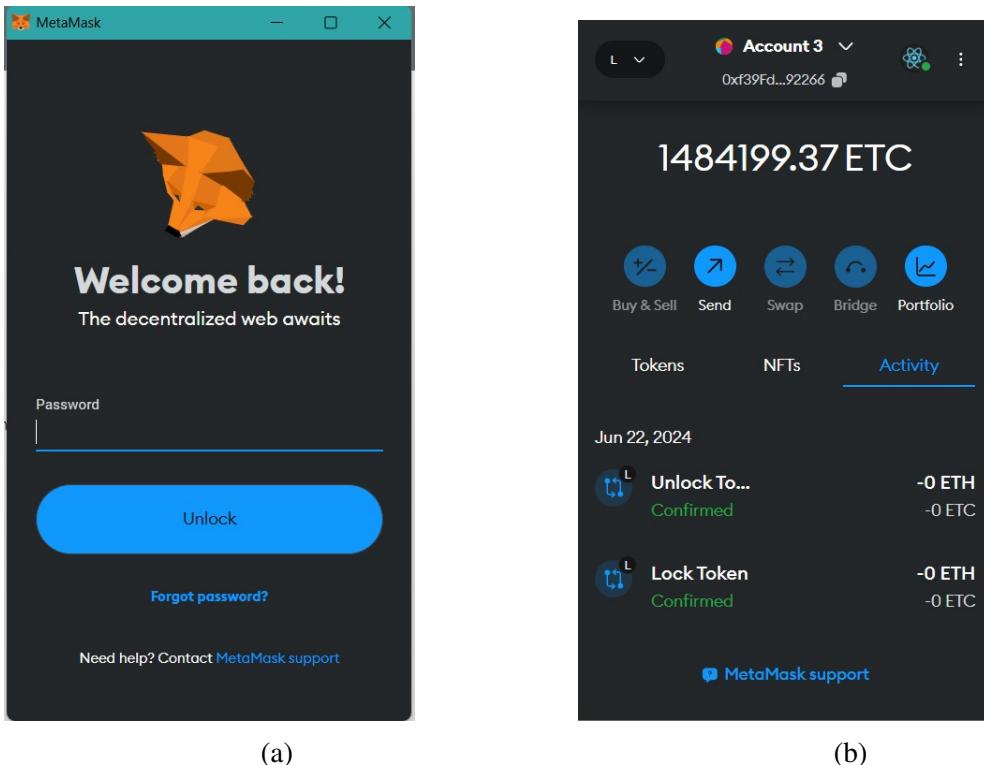
Berikut merupakan langkah-langkah dari pengujian integrasi *smart contract* dengan Web3.0 yang akan dilakukan:

- *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.

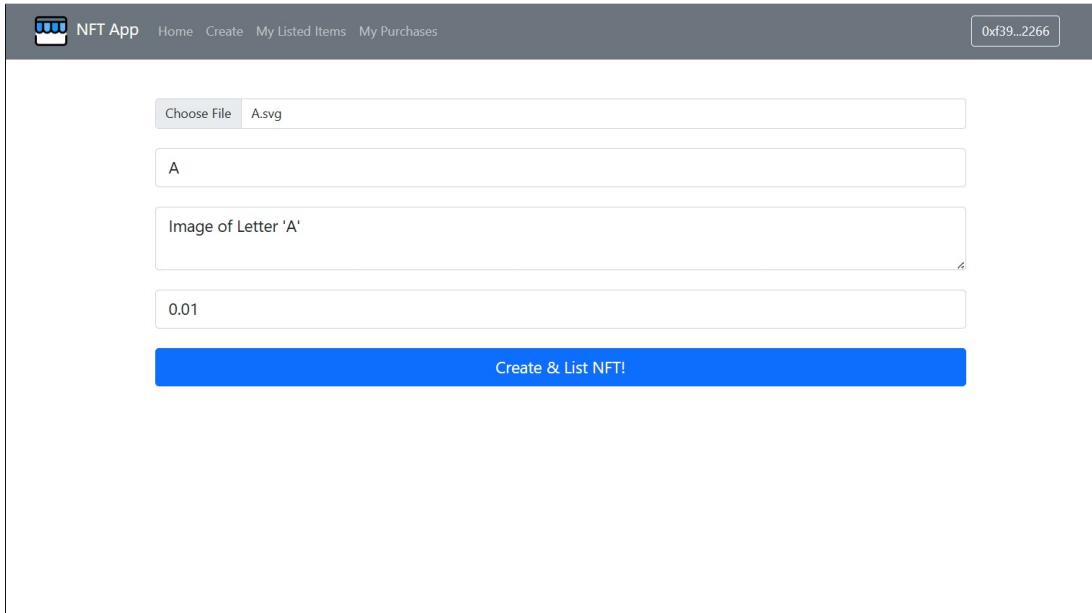


(a)

(b)

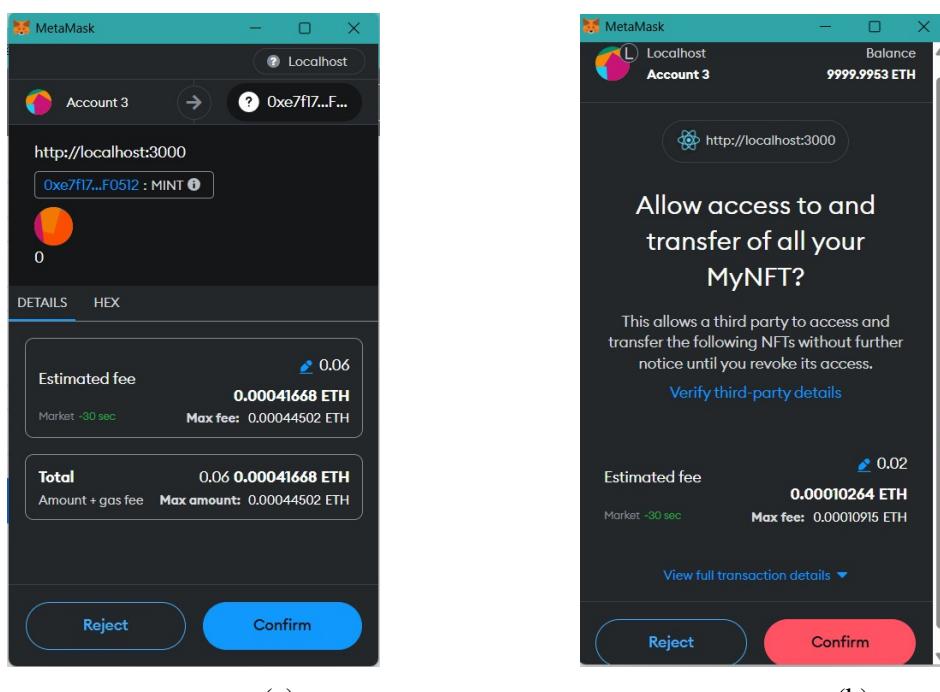
Gambar 4.19: Koneksi Metamask dengan Web

- Pada gambar 4.21 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.
- 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.



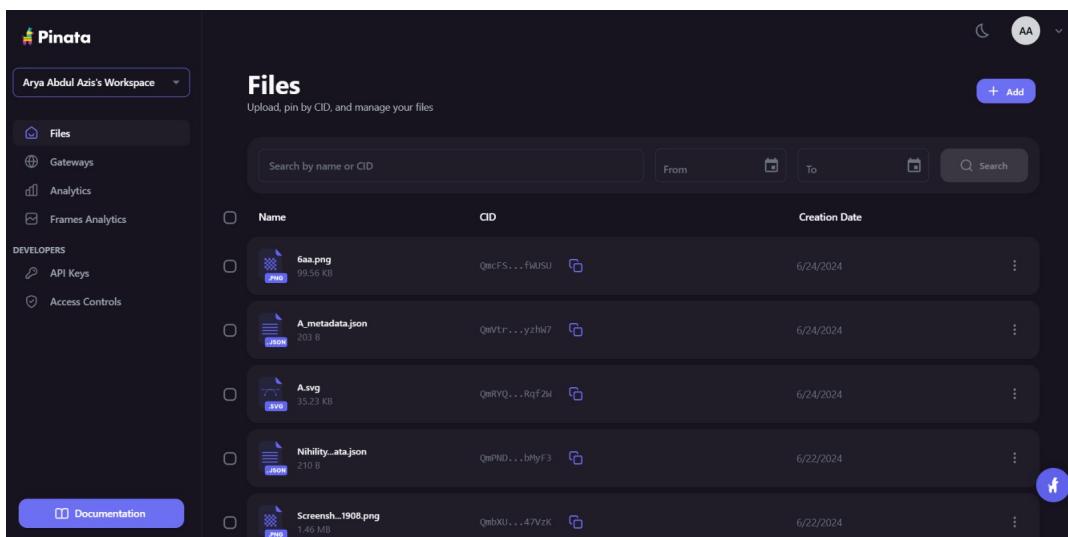
Gambar 4.20: Melakukan pengunggahan NFT pada halaman *create*

- 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*.



Gambar 4.21: Pembayaran gas IPFS menggunakan Metamask Wallet

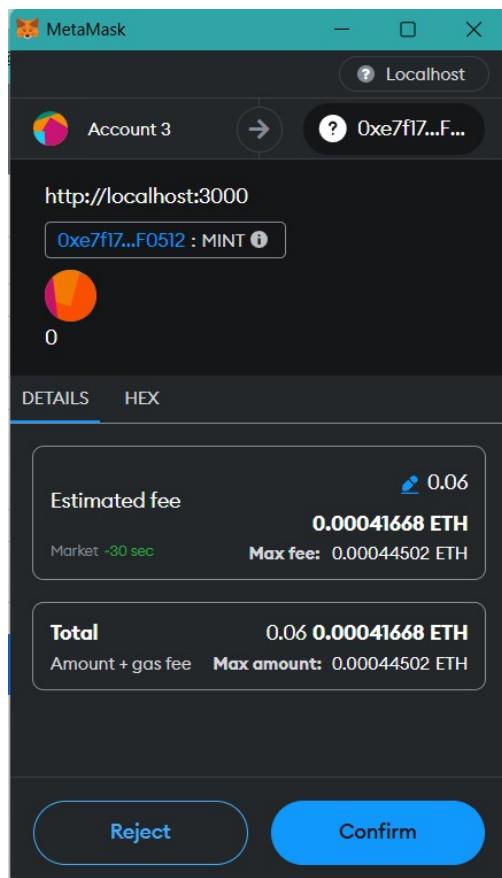
- Gambar 4.21a adalah pembayaran *gas* dari metamask wallet. Pembayaran *gas* tersebut terjadi karena pada *smart contract* terjadi proses pengunggahan gambar NFT ke platform penyedia IPFS. Pada web ini kita mengintegrasikan dengan platform bernama Pinata. Pinata sendiri merupakan platform penyedia servis IPFS. Kemudian juga terdapat konfirmasi pada gambar 4.21b, 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 4.22: Data dari NFT ter-*upload* pada platform Pinata

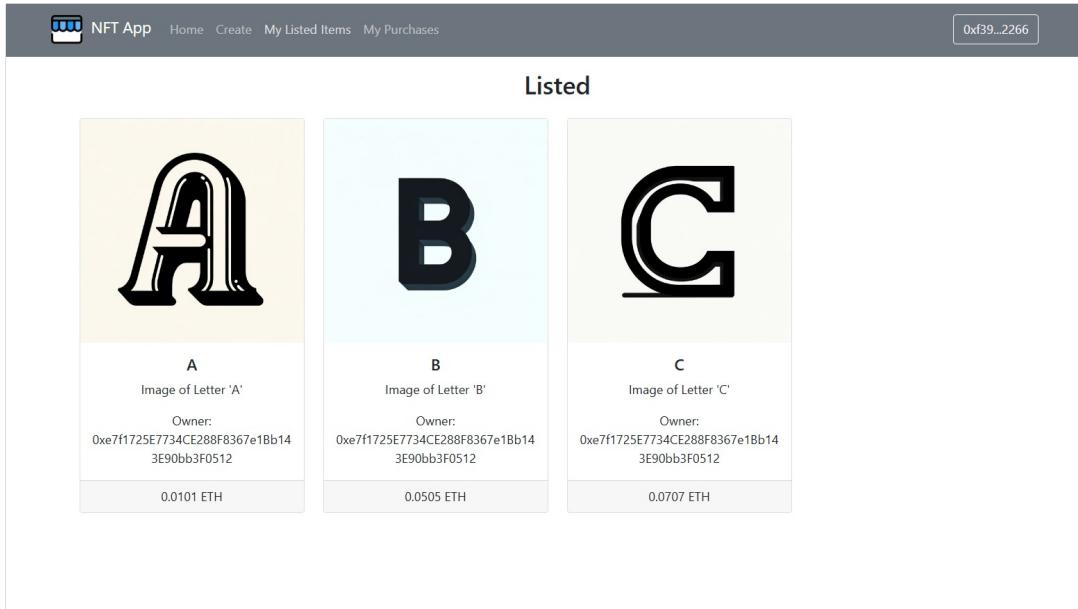
- Dalam platform Pinata seperti pada gambar 4.22, setiap file yang diunggah, termasuk gambar untuk NFT, akan diberikan Content Identifier (CID) unik yang memudahkan pelacakan dan akses tanpa perlu mengkhawatirkan perubahan isi file, karena CID ini akan berubah jika konten file berubah. Hal ini sangat 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.



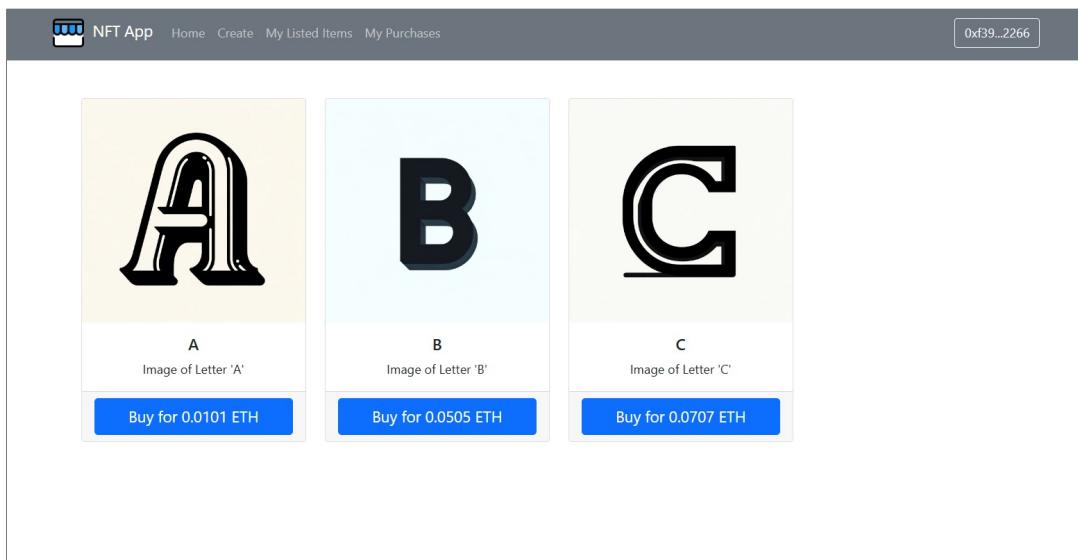
Gambar 4.23: Melakukan konfirmasi *minting* NFT pada Metamask Wallet

- Pada gambar 4.23, 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.



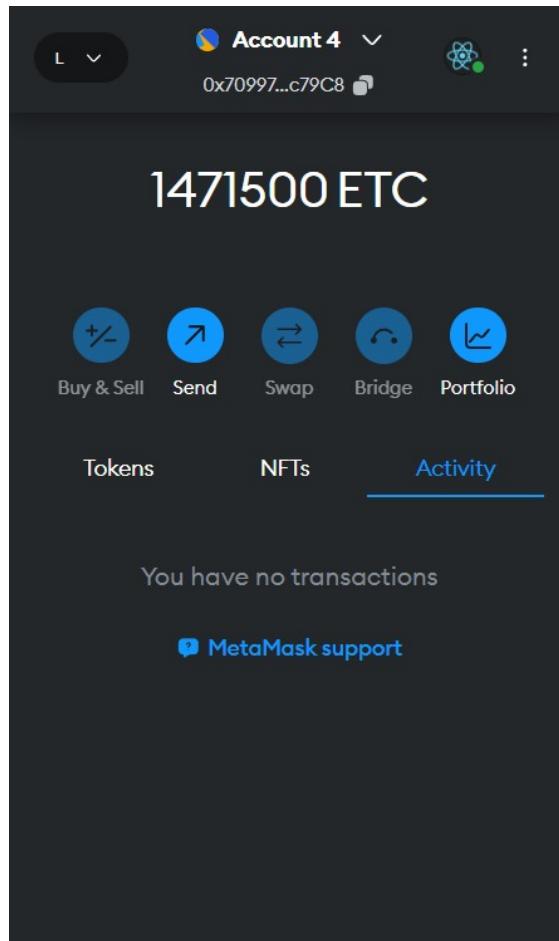
Gambar 4.24: NFT yang telah di-*minting* terlihat pada halaman *My listed items*

- 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. Kemudian juga terdapat NFT dengan desain huruf "B" yang berhasil di-list dengan harga yang tertera di gambar yaitu 0.0505 ETH, dan juga NFT dengan desain huruf "C" yang berhasil di-list dengan harga yang tertera di gambar yaitu 0.0707 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.
- 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.



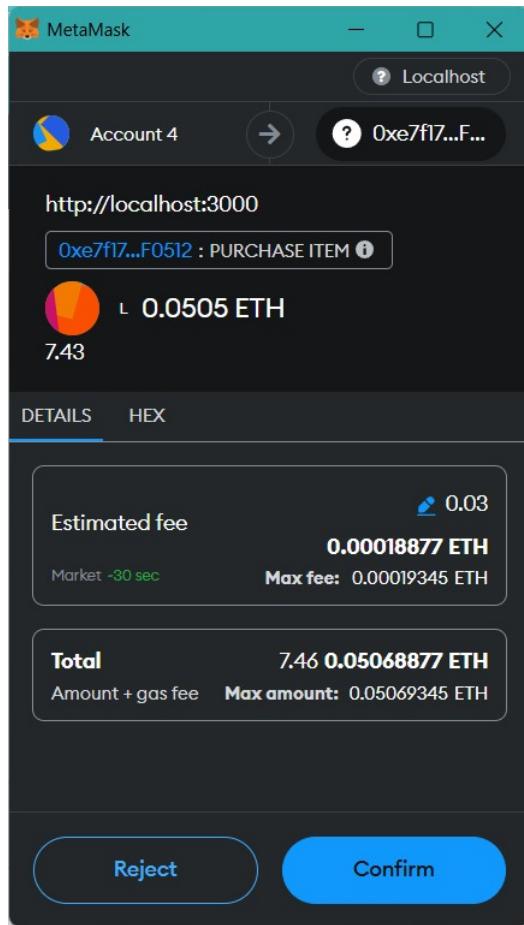
Gambar 4.25: NFT yang telah di-*minting* terlihat pada halaman *home*

- Pada gambar 4.25 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.
- Sebelum melakukan pembelian NFT, pengguna harus mengganti akun yang aktif pada Metamask Wallet ke "Account 4", yang telah disiapkan khusus sebagai akun kedua. Akun ini berfungsi sebagai platform untuk melakukan transaksi pembelian, memungkinkan pengguna untuk menguji proses pembelian NFT yang telah diunggah oleh akun pertama. Penggunaan akun kedua ini sangat penting dalam pengujian fungsionalitas transfer kepemilikan, memastikan bahwa seluruh proses berjalan lancar dan tanpa hambatan.

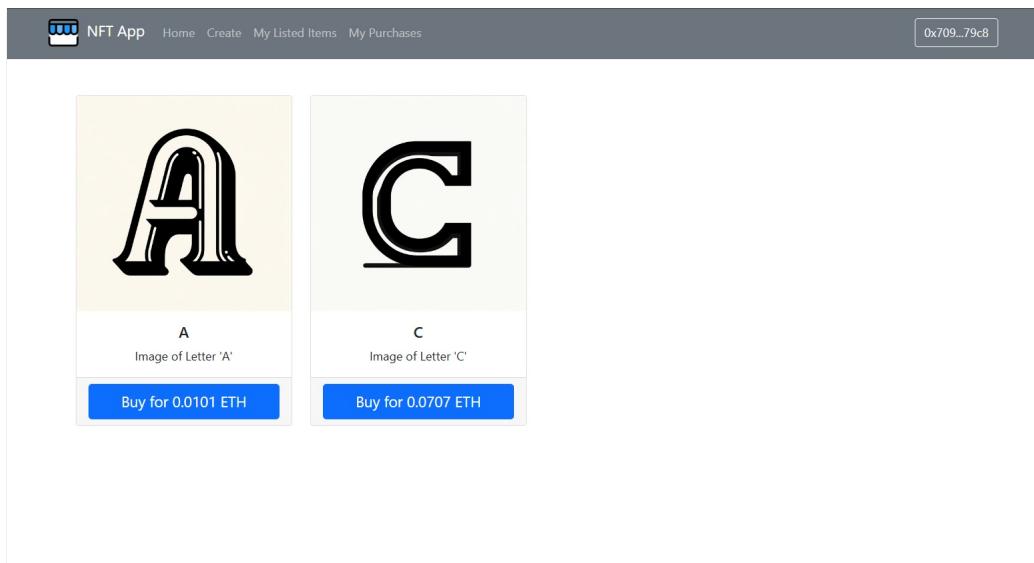


Gambar 4.26: Akun Metamask kedua

- Pada gambar 4.27 ketika pengguna memilih untuk membeli sebuah NFT yang ditampilkan pada halaman utama aplikasi, mereka akan mengklik tombol "Buy" yang berada di bawah item yang diinginkan. Langkah ini memicu proses transaksi, di mana Metamask Wallet secara otomatis terbuka, meminta pengguna untuk mengonfirmasi pembelian. Layar Metamask yang muncul merupakan jendela penting yang menampilkan detail transaksi yang harus diperiksa pengguna sebelum mereka melanjutkan. Dalam jendela konfirmasi ini, pengguna akan melihat jumlah total Ethereum (ETH) yang harus dibayarkan, yang sudah termasuk estimasi biaya gas. Biaya gas ini adalah biaya yang dibutuhkan untuk memproses transaksi di jaringan Ethereum, dan nilainya dapat berfluktuasi berdasarkan kepadatan trafik jaringan saat itu. Metamask memberikan dua estimasi: "Estimated fee" yang merupakan perkiraan biaya gas yang akan dikenakan, dan "Max fee" yang menunjukkan batas maksimal biaya yang mungkin ditarik jika kondisi jaringan berubah secara signifikan selama transaksi diproses. Proses ini tidak hanya memindahkan kepemilikan NFT dari penjual ke pembeli, tetapi juga memperbarui semua record yang berkaitan di *blockchain*. Setelah konfirmasi berhasil, NFT akan muncul dalam koleksi pengguna di aplikasi, dan item tersebut akan dihapus dari daftar yang tersedia untuk dijual, menandakan bahwa transaksi telah selesai secara resmi dan NFT kini memiliki pemilik baru. Proses ini menunjukkan integrasi yang mulus antara platform NFT berbasis web dengan teknologi wallet *blockchain* seperti Metamask, menawarkan pengalaman pengguna yang aman dan efisien.



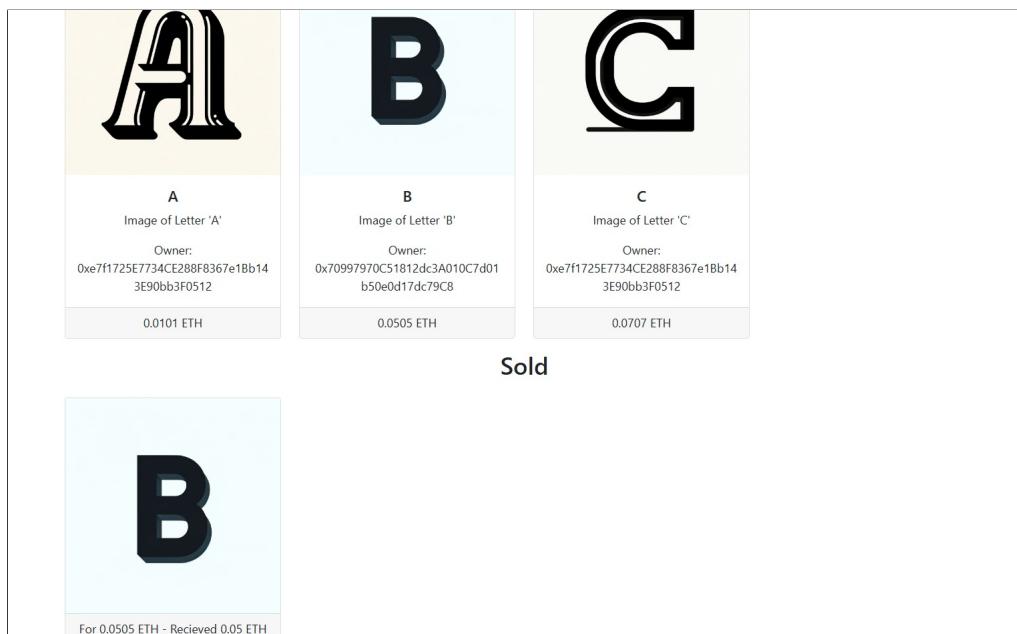
Gambar 4.27: Proses pembelian NFT



Gambar 4.28: Tampilan *home* setelah NFT dibeli

- Pada gambar 4.28, NFT B telah berhasil dibeli oleh *user* B. Proses pembelian ini menyebabkan perubahan langsung pada tampilan *home*, di mana sistem secara otomatis

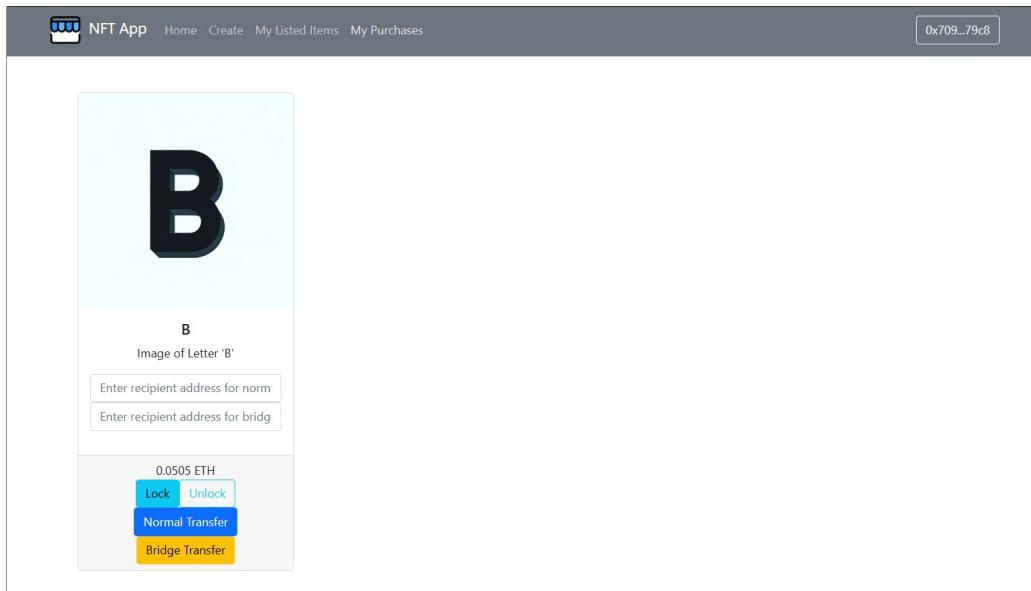
melakukan pembaruan atau *refresh*. Sebelum transaksi, seperti yang terlihat pada gambar 4.25, ada tiga NFT yang tersedia untuk dibeli. Namun, setelah NFT B dibeli, jumlah NFT yang ditampilkan berkurang menjadi dua. Hal ini menunjukkan bahwa sistem berhasil memperbarui daftar NFT yang tersedia sesuai dengan status penjualan terkini. Proses otomatis ini memastikan bahwa informasi yang disajikan kepada pengguna selalu akurat dan terkini, mencerminkan perubahan real-time pada inventori NFT yang ada dalam platform. Perubahan ini tidak hanya memudahkan pengguna dalam melihat NFT yang masih tersedia tetapi juga meningkatkan pengalaman pengguna dengan menampilkan data yang akurat dan responsif terhadap interaksi pengguna di platform.



Gambar 4.29: Tampilan *Listed items* ketika melakukan pembelian

- Pada gambar 4.29 di atas, tampilan menunjukkan NFT 'B' yang telah berhasil dibeli oleh pengguna dari akun kedua, sebagaimana terlihat dari pergantian address pemilik ('Owner') pada NFT tersebut dari *address* akun 1 ke alamat *address* dari akun 2. Proses pembelian NFT ini diawali ketika pengguna mengklik tombol pembelian pada salah satu item NFT yang terdaftar di halaman utama, yang kemudian memicu interaksi dengan MetaMask wallet. Pengguna kemudian diminta untuk mengonfirmasi dan menyelesaikan pembayaran melalui wallet tersebut. Setelah transaksi berhasil diverifikasi dan konfirmasi pembayaran diterima, sistem otomatis mengubah status item di halaman "My Listed Items" menjadi "Sold", yang menandakan bahwa item tersebut telah sukses terjual dan tidak lagi tersedia untuk pembelian oleh pengguna lain. Perubahan status ini tidak hanya memastikan transparansi dalam status ketersediaan NFT tetapi juga menunjukkan pembaruan real-time dari kepemilikan aset di platform.
- Pada gambar 4.29 juga setelah pengguna berhasil menyelesaikan pembelian NFT dari halaman utama, sistem akan memperbarui status NFT tersebut di halaman "My Listed Items" untuk menunjukkan bahwa item telah terjual. Bagian "Sold" di halaman ini akan menampilkan NFT yang telah dibeli, memberikan informasi bahwa NFT tersebut tidak lagi tersedia untuk dibeli oleh pengguna lain. Tampilan ini mencakup gambar NFT, harga

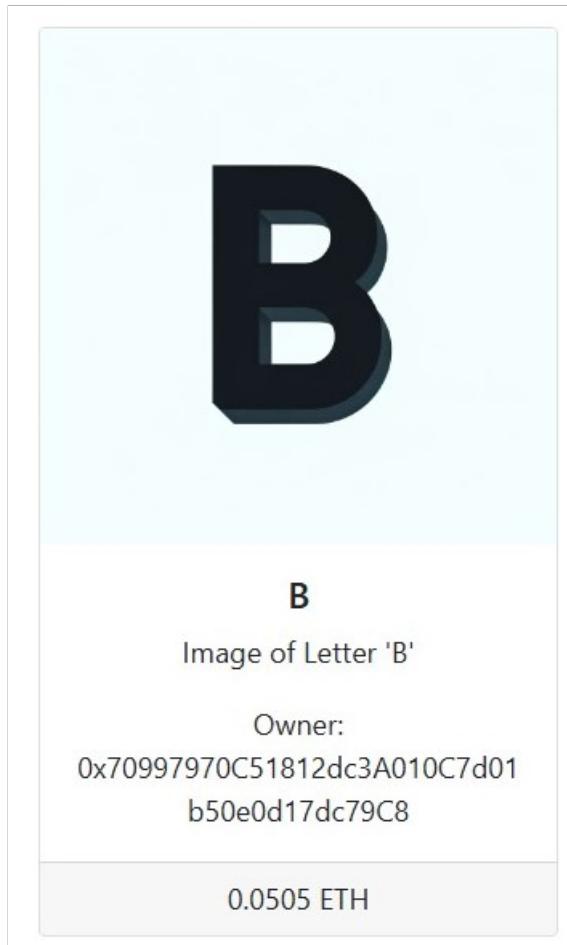
jual, serta indikasi bahwa transaksi telah berhasil dengan menunjukkan jumlah ETH yang diterima. Proses ini memastikan bahwa semua pengguna aplikasi dapat melihat secara jelas item yang telah terjual, meningkatkan transparansi dan memudahkan pelacakan kepemilikan NFT. Fungsi ini sangat penting dalam ekosistem NFT, di mana verifikasi kepemilikan dan status penjualan harus diperbarui secara real-time untuk mencegah penjualan ganda dan memastikan integritas platform.



Gambar 4.30: Tampilan halaman *my purchases* ketika sudah melakukan pembelian

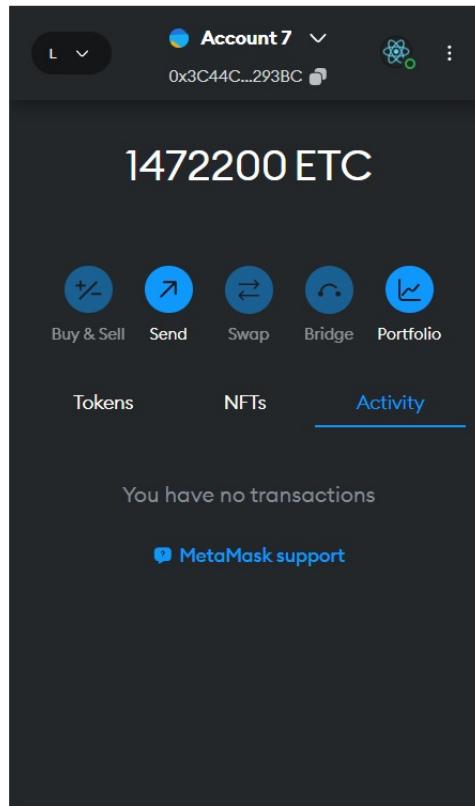
- Lalu pada gambar 4.30 adalah tampilan dari halaman *my purchases* di address akun ke dua. Pada tampilan ini pemilik dari NFT dapat melihat dari koleksi NFT yang telah dibeli dan juga NFT yang telah dibeli dapat dilakukan hal yaitu *transfer ownership*. *Transfer ownership* yang dapat dilakukan ada dua hal, yaitu *transfer* pada *network* atau *blockchain* yang sama dan juga ada *transfer* pada *network* atau *blockchain* yang berbeda. Kedua fungsi tersebut berbeda karena untuk lintas *blockchain* diperlukan beberapa tahapan atau protokol yaitu terdapat protokol *lock* dan *unlock*. Pada tahapan selanjutnya yang akan dilakukan adalah bagaimana cara melakukan *transfer* dalam satu *network* yang sama.

4.2.3 Pengujian Fitur Transfer *Ownership* Secara *Localhost* Dan Secara Lintas *Blockchain*



Gambar 4.31: Tampilan halaman *my purchases* ketika sudah melakukan pembelian

Sebelum melakukan pengujian pemindahan kepemilikan NFT, penggunaan akun ketiga dengan *address* "0x3C44CdDdB6a900fa2b585dd299e03d12FA4293BC" sangat krusial. Akun ini secara khusus disiapkan untuk menangani proses transfer kepemilikan NFT dari akun kedua. Seperti yang terlihat pada gambar 4.24, awalnya NFT diunggah melalui halaman *create* dan secara otomatis, *owner* pertama NFT adalah *address* dari *smart contract* yang melakukan *minting*. Hal ini berubah setelah NFT dibeli, sebagaimana ditunjukkan dalam gambar 4.31 di mana *address* akun kedua menjadi pemilik baru. Perpindahan kepemilikan ini terjadi melalui transaksi yang tercatat dan diverifikasi dalam blockchain, memastikan bahwa NFT berpindah tangan dari akun pencipta ke pembeli dengan aman dan terdokumentasi dengan jelas, menunjukkan kekuatan dan fleksibilitas teknologi *smart contract* dalam mengelola aset digital dalam ekosistem blockchain.



Gambar 4.32: Detail akun ketiga pada Metamask Wallet

Pada gambar 4.32, terlihat tampilan akun MetaMask yang disebut "Account 7" dengan saldo yang menonjol sebanyak 1472200 ETC, yang menunjukkan jumlah Ethereum Classic di dalam wallet. Akun ini, yang beralamat di 0x3C44CdDdB6a900fa2b585dd299e03d12FA4293 BC, belum memiliki catatan transaksi apapun, menandakan bahwa belum ada aktivitas jual beli, pengiriman, penukaran, atau transaksi lainnya yang dilakukan. Hal ini juga memperlihatkan bahwa akun ini adalah akun baru atau belum digunakan untuk transaksi. Akun ini didapatkan dari kunci privat yang berasal dari node Hardhat, sering digunakan dalam pengembangan dan pengujian smart contract di lingkungan lokal. Akun ini direncanakan akan digunakan sebagai penerima dalam proses transfer kepemilikan NFT, memanfaatkan fasilitas yang disediakan MetaMask untuk mengelola aset kripto dan interaksi dengan *blockchain*.

Ekspektasi dari pengujian fitur *transfer ownership* pada web3.0 yang telah dibuat lain adalah sebagai berikut:

- Pengguna dari pemilik NFT yang sekarang (pengguna *address* kedua) dapat melakukan pemindahan kepemilikan NFT yang telah dimiliki menjadi kepemilikan pengguna *address* ketiga yang berada dalam *network* yang sama yaitu *localhost*.
- Pengguna dari pemilik NFT yang sekarang (pengguna pertama) dapat melakukan pemindahan kepemilikan NFT yang telah dimiliki menjadi kepemilikan pengguna pengguna kedua berada dalam *network* yang berbeda yaitu *Sepolia Testnet* pada pengguna pertama dan *BNB Chain Testnet* pada pengguna kedua.
- Pada halaman *my listed item* pembuat dari NFT awal (pengguna *address* pertama) *owner* dari NFT yang telah dilakukan *transfer ownership* secara *localhost* maupun lintas *blockchain*

itu alamatnya berganti dari *address* pemilik lama menjadi *address* pemilik baru.

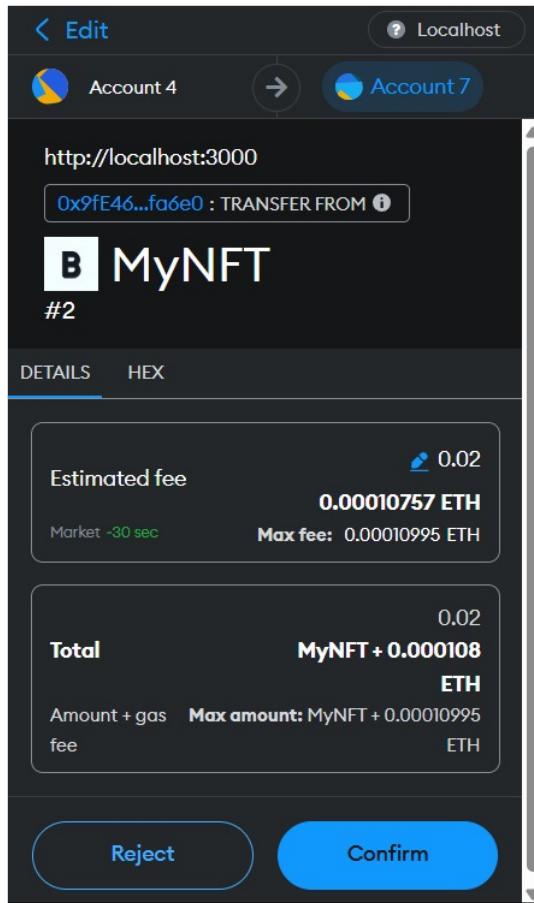
Berikut ini merupakan langkah pengujian beserta dengan pembahasan dari pengujian fitur transfer *ownership* Secara *Localhost* Dan Secara Lintas *Blockchain*:

- Pada gambar 4.30, dapat dilihat pada halaman *my purchases* milik *address* akun kedua ter-dapat NFT "B". Setelah transaksi pembelian NFT ini, platform menyediakan opsi untuk mengelola NFT tersebut, termasuk kemampuan untuk melakukan transfer kepemilikan atau memblokirnya untuk transaksi lebih lanjut. Pengguna dapat memasukkan alamat penerima untuk transfer normal atau transfer lintas rantai, memanfaatkan fungsionalitas yang memudahkan transfer NFT antara berbagai blockchain atau dalam jaringan yang sama.



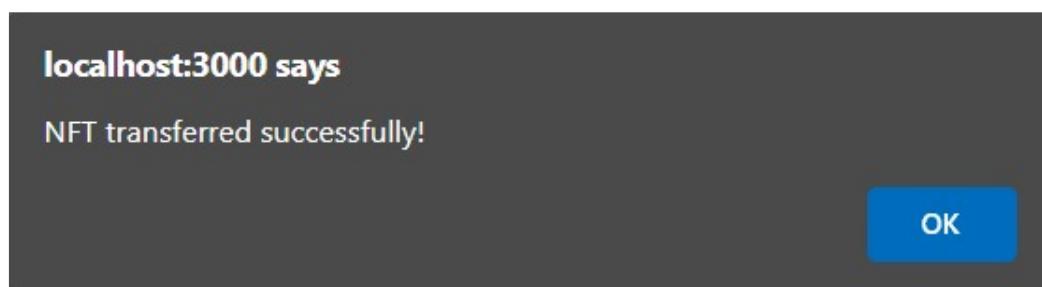
Gambar 4.33: Tampilan NFT yang siap dilakukan pindah kepemilikan

- Pada pengujian ini, digunakan NFT "B" yang akan dilakukan transfer *ownership* dalam satu *network*, pada kolom pertama yang bertuliskan "Enter recipient address for normal transfer *ownership*" akan dimasukkan *address* dari akun ketiga yang berupa "*0x3C44CdD dB6a900fa2b585dd299e03d12FA4293BC*". Setelah dimasukkan *address* tersebut maka akan langsung saja ditekannya tombol "Normal Transfer". Tombol "Normal Transfer" tersebut lalu akan membuat pengguna langsung membuka window dari Metamask Wallet untuk melakukan transaksi pembayaran dari "gas" atau fee yang harus dibayarkan setiap kali kode dari *smart contract* tereksekusi. Setiap aksi yang dilakukan oleh *smart contract* seperti transaksi, perubahan data, atau eksekusi fungsi—memerlukan gas tertentu.



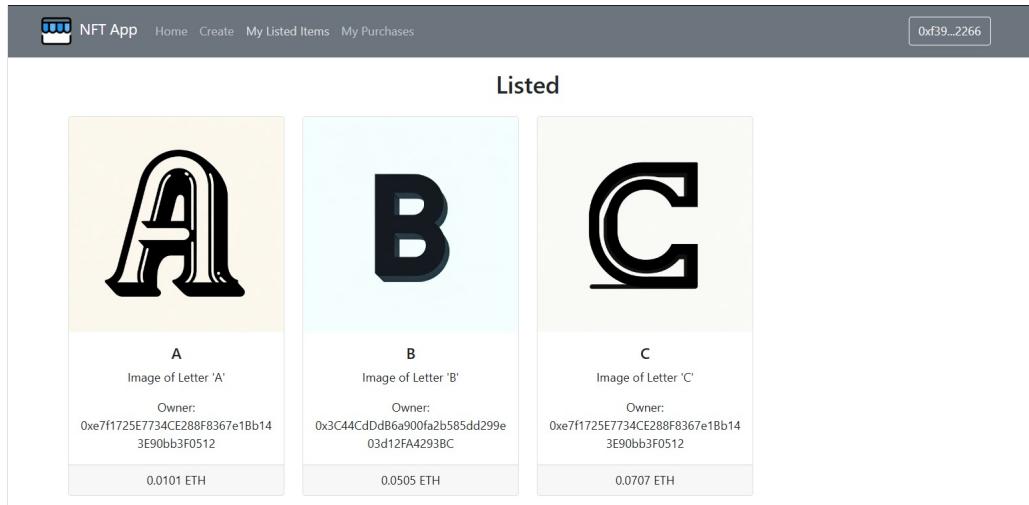
Gambar 4.34: Tampilan NFT yang siap dilakukan pindah kepemilikan

- Gambar 4.34 ini menunjukkan tampilan antarmuka Metamask saat pengguna kedua siap melakukan transaksi transfer dari NFT bernama "MyNFT" dengan nomor #2. Nomor #2 ditunjukkan bahwa NFT itu memiliki indeks urutan kedua. Pada tampilan ini, pengguna diberi informasi mengenai jumlah ETH yang akan ditransfer sebesar 0.02 ETH dan biaya transaksi yang diperkirakan. Biaya ini terbagi menjadi dua, yaitu biaya perkiraan (estimated fee) sebesar 0.00010757 ETH dan biaya maksimal (max fee) sebesar 0.00010995 ETH, yang memberikan kejelasan tentang berapa banyak gas yang mungkin diperlukan untuk menyelesaikan transaksi tersebut dalam waktu kurang lebih 30 detik. Fitur ini memfasilitasi pengguna dalam memastikan bahwa mereka memiliki cukup saldo untuk menutupi biaya transaksi saat melakukan transfer NFT. Tampilan juga menawarkan pilihan untuk menyetujui atau menolak transaksi yang akan dilakukan.



Gambar 4.35: Alert "NFT transferred successfully!" ketika NFT berhasil dipindahkan

- Setelah pengguna melakukan konfirmasi transaksi pada Metamask Wallet seperti pada gambar 4.33 dan transaksi berhasil, maka akan muncul *alert* berupa "NFT transferred successfully!".



Gambar 4.36: Alert "NFT transferred successfully!" ketika NFT berhasil dipindahkan

- Dapat dilihat pada gambar 4.36 pada halaman *my listed items* milik pengguna *address* pertama yang me-*minting* NFT B, jika dibandingkan dengan gambar 4.31, *owner* dari NFT B telah berganti menjadi *address* milik akun ketiga. Ini membuktikan bahwa NFT B kepemilikannya berhasil dipindahkan dari *address* pengguna kedua menjadi *address* pengguna ketiga.

```

From: 0x3c44cdDd6a900fa2b585dd299e03d12fa4293bc
To: 0xe7f1725e7734ce288f8367e1bb143e90bb3f0512

eth_call
Contract call: Marketplace#items
From: 0x3c44cdDd6a900fa2b585dd299e03d12fa4293bc
To: 0xe7f1725e7734ce288f8367e1bb143e90bb3f0512

eth_call
Contract call: NFT#tokenURI
From: 0x3c44cdDd6a900fa2b585dd299e03d12fa4293bc
To: 0x9fe46736679d2d9a65f0992f2272de9f3c7fa6e0

eth_call
Contract call: Marketplace#getTotalPrice
From: 0x3c44cdDd6a900fa2b585dd299e03d12fa4293bc
To: 0xe7f1725e7734ce288f8367e1bb143e90bb3f0512

eth_call
Contract call: Marketplace#items
From: 0x3c44cdDd6a900fa2b585dd299e03d12fa4293bc
To: 0xe7f1725e7734ce288f8367e1bb143e90bb3f0512

eth_getLogs
eth_call
Contract call: NFT#tokenURI
From: 0x3c44cdDd6a900fa2b585dd299e03d12fa4293bc
To: 0x9fe46736679d2d9a65f0992f2272de9f3c7fa6e0

eth_call
Contract call: Marketplace#getTotalPrice
From: 0x3c44cdDd6a900fa2b585dd299e03d12fa4293bc
To: 0xe7f1725e7734ce288f8367e1bb143e90bb3f0512

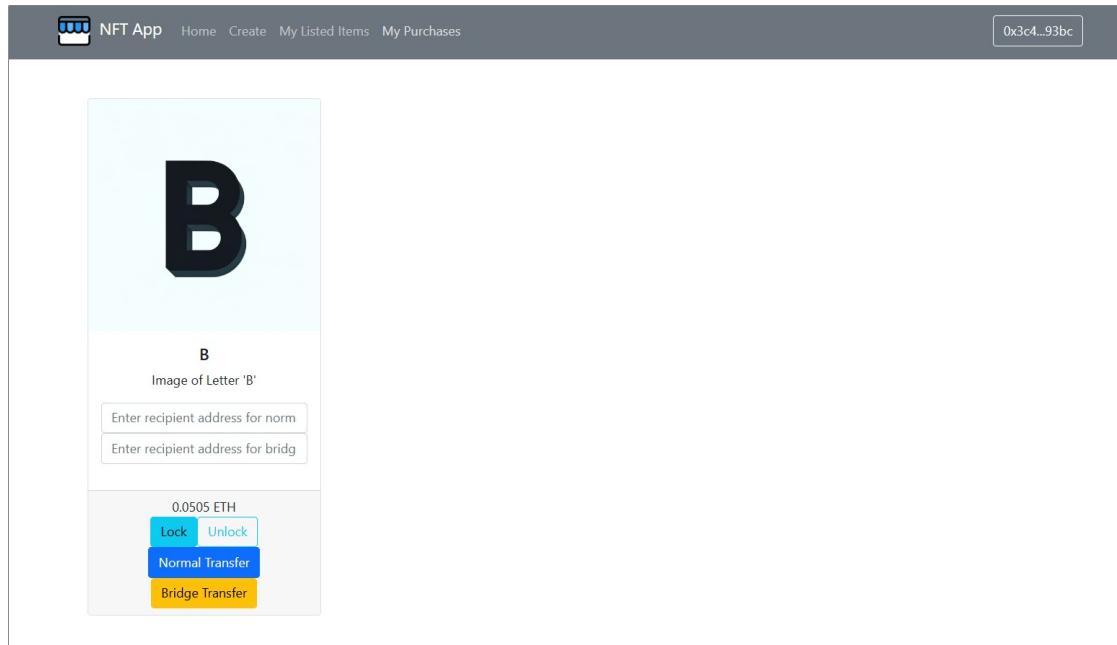
eth_blockNumber
eth_getBalance (7)
eth_feeHistory
* History restored

```

Gambar 4.37: Console Transaksi Pada Hardhat Node

- Pada gambar 4.37 adalah detail transaksi yang terjadi pada *smart contract* ketika ada fungsi yang terseleksi. Dikarenakan pengembangan web3.0 dan juga sistem *smart con-*

tract ini dilakukan secara lokal, maka *contract call* dan juga detail dari *hash block* hanya dapat dilihat pada *node* yang berada pada EVM milik Hardhat.



Gambar 4.38: Tampilan halaman *my purchases* pada akun ketiga

- Lalu selain dari 4.36 juga terdapat hasil bahwa NFT telah berpindah pada akun ketiga. Hal ini dapat dilihat pada *box* kanan atas menunjukkan *address* dari akun ketiga yang berarti ini adalah halaman *my purchases* milik akun ketiga. Lalu akun ketiga ini juga yang memiliki wewenang penuh dalam kepemilikan NFT yang sebelumnya dimiliki oleh akun kedua.

(a)
(b)

Gambar 4.39: Detail Akun Untuk Pengujian Lintas Rantai

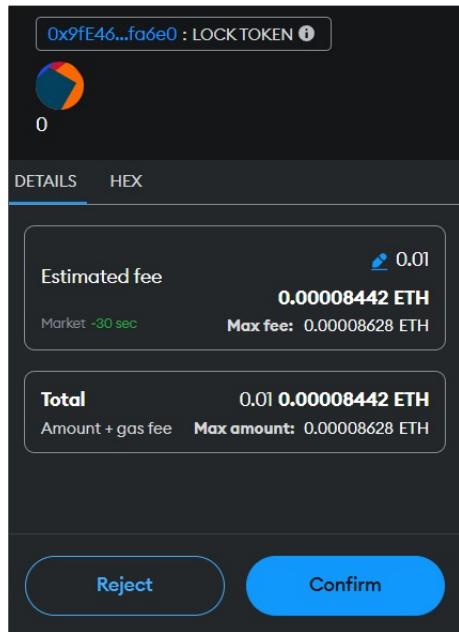
- Gambar pertama menggambarkan akun di Sepolia Ethereum Testnet dengan saldo SepoliaETH yang ditampilkan secara jelas. Ini termasuk aktivitas terkini yang mencakup penerimaan SepoliaETH, memperlihatkan aktivitas yang dikonfirmasi dalam jaringan. Akun pada gambar pertama ini akan digunakan sebagai *address* penerima dari NFT yang dikirimkan dari *network* yang berbeda. Akun ini memiliki *address* "0x6A1B77e82b61D54C4F1A27cd00A27325EBf361eD"
- Gambar kedua menampilkan BinanceTestNet Wallet dalam mode testnet, yang memungkinkan pengguna untuk mengirim, menerima, bertukar, dan membeli aset kripto seperti AVAX, BNB, DEV, dan ETH dalam lingkungan tes. Akun pada gambar kedua ini akan digunakan sebagai *address* penerima dari NFT yang dikirimkan dari *network* yang berbeda. Akun ini memiliki *address* "0xD066d6576D9485Eb2c2a41BB8B52EcE17a0557d6"



Gambar 4.40: Memasukkan *address* tujuan untuk *bridge transfer*

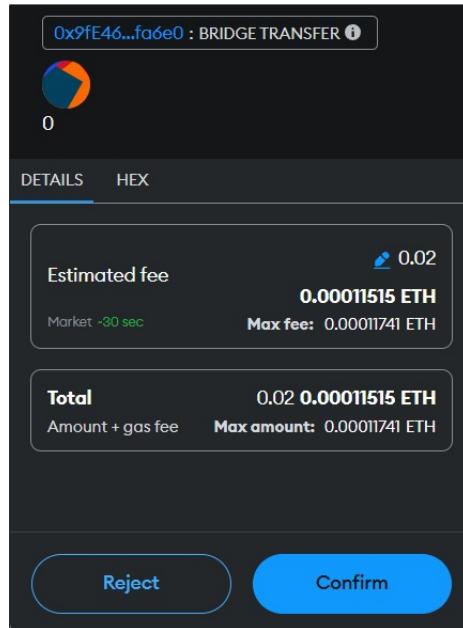
- Pengujian untuk melakukan tes interoperabilitas atau transfer *ownership* lintas *blockchain* dilakukan dengan prosedur yang terstruktur. Pertama, akun pengirim yang beroperasi di jaringan *localhost* akan memilih NFT yang ingin ditransfer. Kemudian, alamat tujuan dari akun yang berjalan di jaringan Sepolia diisi ke dalam sistem. Proses transfer diawali dengan penguncian (*lock*) NFT untuk memastikan keamanan selama transisi. Fungsi *lock* ini penting untuk mencegah transaksi atau interaksi lain yang mungkin mempengaruhi NFT

selama proses transfer belum selesai. Setelah NFT terkunci, barulah transfer ownership akan dijalankan melalui fungsi *bridge transfer*.



Gambar 4.41: Konfirmasi pembayaran gas dari fungsi *lock token* di Metamask Wallet

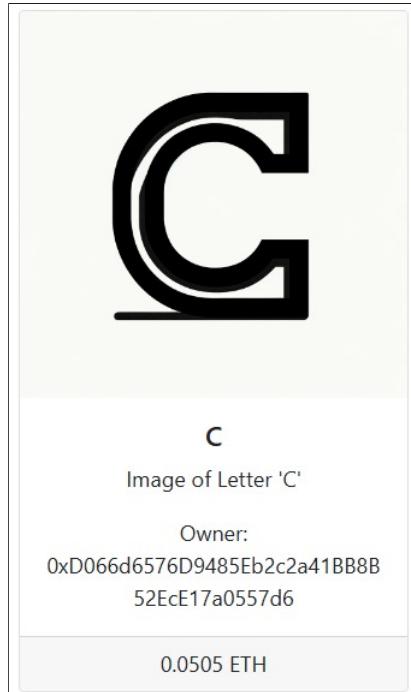
- Setelah melakukan konfirmasi pembayaran gas dari fungsi *lock token*, kondisi NFT berubah menjadi *locked*. Setelah NFT menjadi *locked* maka NFT sudah aman untuk dilakukan transaksi lintas *blockchain*.



Gambar 4.42: Konfirmasi pembayaran gas dari fungsi *bridge transfer* di Metamask Wallet

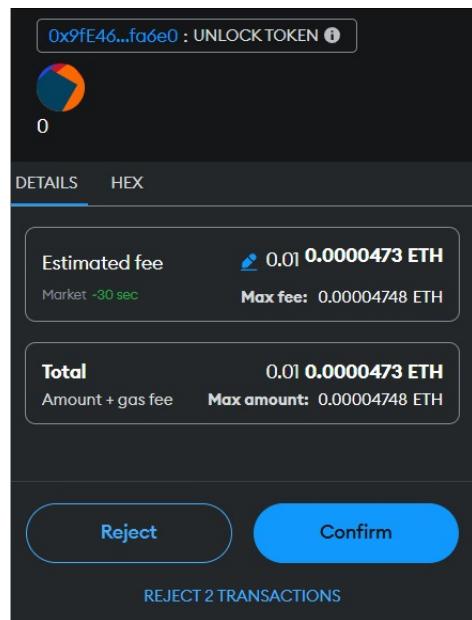
- Setelah melakukan konfirmasi pembayaran gas dari fungsi *bridge transfer*, kepemilikan dari NFT "C" berpindah dari akun pertama yang memiliki *address* "0xf39Fd6e51aad88F6

F4ce6aB8827279cffFb92266” yang berjalan pada *network localhost* menjadi kepemilikan akun yang berjalan pada *network BNB Smart Chain* dengan alamat ”0xD066d6576D9485Eb2c2a41BB8B52EcE17a0557d6”. Hal ini dapat dibuktikan pada gambar di bawah ini.



Gambar 4.43: Kepemilikan NFT ”C” Berganti Menjadi Akun *Network BNB Smart Chain*

- Kemudian hal yang perlu dilakukan yang terakhir adalah mengubah *state* dari NFT tersebut yang awalnya ”*locked*” menjadi ”*unlocked*”. Hal ini harus dilakukan agar NFT tersebut dapat digunakan untuk transaksi lagi pada *network* tersebut.



Gambar 4.44: Konfirmasi Pembayaran Gas *Unlock Token*

- Setelah melakukan penguncian NFT, langkah berikutnya adalah membuka kunci atau “unlock” NFT tersebut agar dapat kembali digunakan dalam transaksi. Proses ini melibatkan interaksi dengan sistem *blockchain* yang tercatat sebagai transaksi dengan estimasi biaya yang diperlihatkan dalam gambar. Di sini, biaya transaksi atau ‘*gas fee*’ yang diperkirakan untuk membuka kunci token ditampilkan, menekankan pentingnya memperhatikan biaya transaksi yang harus dibayar untuk memproses kegiatan ini di *blockchain*. Proses membuka kunci ini tidak hanya mengizinkan pemilik untuk memindahkan atau menjual NFT tersebut, tetapi juga penting untuk memastikan fleksibilitas dan likuiditas NFT di pasar.

		
A Image of Letter 'A'	B Image of Letter 'B'	C Image of Letter 'C'
Owner: 0xe7f1725E7734CE288F8367e1Bb14 3E90bb3F0512	Owner: 0x6A1B77e82b61D54C4F1A27cd00 A27325EBf361eD	Owner: 0xD066d6576D9485Eb2c2a41BB8B 52EcE17a0557d6
0.0505 ETH	0.0505 ETH	0.0505 ETH

Gambar 4.45: Hasil Akhir Dari Transfer Lintas Rantai

- Setelah dilakukan pengujian yang sama tetapi memasukkan *address* dari akun yang berada pada *network sepolia ethereum*, untuk transfer *ownership* NFT B dapat dilihat di gambar 4.45 kepemilikan atau *owner* dari NFT B juga berubah menjadi *address* dari akun *sepolia*.

```
Recent Block Hashes:
0xe4f2f8dc65242ddbb9f280f20133054c808dbbf7316c85cc5f149b4e2ec63168
0xc9b6169b53f42419271469a21bfced7249745b8cb56ad5714faf631de97b2837
0x02b9df7282d47b5c8db095bcc106cd94415baff4ba204d7ede98ab2de765e1df
0x5ae4d5c933905d3ecce88b837d02c82e7f3238420fcba0e5aeeff1177fbe867b2
0x41d0e5194a4a2090ba5eb2cdc5bbf1a2034bbac57739f587e47cc1e6a137a546
```

Gambar 4.46: *Block Hash* dari setiap transaksi

- Pada gambar 4.46, merupakan *block hash* dari transaksi yang telah terjadi. *Hash Blocks* ini didapatkan dari Hardhat. *Block hash* adalah nilai unik yang dihasilkan dari *blok* data

dalam *blockchain*, yang bertindak sebagai identitas digital *blok* tersebut. Hash ini dihasilkan menggunakan algoritma hash kriptografi, yang mengubah informasi *blok* menjadi string karakter tetap panjang yang kompleks dan unik. Proses ini memastikan keamanan dan integritas data dalam *blockchain* karena setiap perubahan kecil pada *blok* akan menghasilkan hash yang sangat berbeda, sehingga mudah untuk mendeteksi perubahan atau manipulasi data. Hash *blok* juga digunakan untuk menghubungkan *blok-blok* dalam *blockchain* *hash* dari *blok* sebelumnya dimasukkan ke dalam *blok* berikutnya, menciptakan rantai data yang terkait dan aman.

[Halaman ini sengaja dikosongkan]

BAB V

PENUTUP

5.1 Kesimpulan

Dalam penelitian ini, penulis telah mengeksplorasi dan mengimplementasikan konsep interoperabilitas NFT berbasis blockchain menggunakan smart contract dalam konteks Web3.0. Studi ini berhasil menunjukkan potensi dan efektivitas teknologi blockchain, khususnya Ethereum, dalam mendukung penciptaan dan pengelolaan NFT yang dapat beroperasi lintas berbagai token. Lalu Kesimpulan dari penelitian ini adalah

- Implementasi Smart contract yang dirancang telah berhasil memfasilitasi penciptaan, transaksi, dan verifikasi NFT secara aman dan efisien. Dengan menggunakan standar ERC-721, penulis telah mengimplementasikan NFT yang unik dan dapat dilacak asal-usulnya, yang sangat penting dalam ekosistem Web3.0.
- NFT yang dikembangkan menunjukkan interoperabilitas lintas berbagai platform *blockchain*. Hal ini dicapai melalui penggunaan protokol *cross-chain interoperability protocol* yang konsisten.

5.2 Saran

Untuk meningkatkan interoperabilitas dan efisiensi penggunaan NFT d disarankan agar penelitian lebih lanjut difokuskan pada pengembangan protokol yang lebih universal untuk integrasi lintas rantai. Selain itu, penelitian lebih lanjut diperlukan untuk menyempurnakan mekanisme keamanan yang dapat melindungi privasi pengguna sambil mempertahankan transparansi dan keandalan sistem blockchain.

[Halaman ini sengaja dikosongkan]

DAFTAR PUSTAKA

- [1] Q. Wang, R. Li, Q. Wang, and S. Chen, “Non-fungible token (nft): Overview, evaluation, opportunities and challenges.,” *NFT's*, 2021.
- [2] S. N. Khan, F. Loukil, C. Ghedira-Guegan, E. Benkhelifa, and A. Bani-Hani, “Blockchain smart contracts: Applications, challenges, and future trends.,” *Blockchain smart contracts: Applications, challenges, and future trends.*, 2023.
- [3] N. Malik, Y. ”. Wei, G. Appel, and L. Luo, “Blockchain technology for creative industries: Current state and research opportunities,” *International Journal of Research in Marketing*, 2023.
- [4] T. R. Gadekallu, T. Huynh-The, W. Wang, *et al.*, “Blockchain for the metaverse: A review,” *Blockchain*, 2022.
- [5] D. Tjokrosetio, “The complete guide to blockchain.,” *Tempo Publishing*, 2022.
- [6] A. Antonopoulos, G. Wood, and G. Wood, “Mastering ethereum: Building smart contracts and dapps.,” *O'Reilly Media, Incorporated.*, 2018.
- [7] C. Dannen, “Introducing ethereum and solidity,” *Foundations of Cryptocurrency and Blockchain Programming for Beginners*, 2017.
- [8] P. Freni, E. Ferro, and R. Moncada, “Tokenomics and blockchain tokens: A design-oriented morphological framework.,” *Blockchain: Research and Applications*, 2022.
- [9] Z. Zheng, S. Xie, H.-N. Dai, *et al.*, “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*, 2020.
- [10] M. Steichen, B. Fiz, R. Norvill, W. Shbair, and R. State, “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)*, 2018.
- [11] P. P. Ray, “Web3: A comprehensive review on background, technologies, applications, zero-trust architectures, challenges and future directions,” *Internet of Things and Cyber-Physical Systems*, vol. 3, pp. 213–248, 2023, ISSN: 2667-3452. doi: <https://doi.org/10.1016/j.iotcps.2023.05.003>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2667345223000305>.
- [12] U. W. Chohan, “Web 3.0: The future architecture of the internet?..,” *Web3.0*, 2022.
- [13] J. Bambacht and J. Pouwelse, *Web3: A decentralized societal infrastructure for identity, trust, money, and data*, 2022. arXiv: 2203.00398.
- [14] S. Lu, J. Pei, R. Zhao, *et al.*, “Ccio: A cross-chain interoperability approach for consortium blockchains based on oracle,” *Sensors*, vol. 23, no. 4, 2023, ISSN: 1424-8220. doi: 10.3390/s23041864. [Online]. Available: <https://www.mdpi.com/1424-8220/23/4/1864>.

- [15] B. Pillai, K. Biswas, and V. Muthukkumarasamy, “Cross-chain interoperability among blockchain-based systems using transactions,” *The Knowledge Engineering Review*, vol. 35, e23, 2020. doi: [10.1017/S026988920000314](https://doi.org/10.1017/S026988920000314).

BIOGRAFI PENULIS



Arya Abdul Azis, lahir di kota Tulungagung pada 24 Maret 2002, adalah seorang mahasiswa yang memiliki kecintaan mendalam terhadap teknologi dan inovasi. Saat ini, Arya menempuh pendidikan di Institut Teknologi Sepuluh Nopember, di jurusan Teknik Komputer, yang dikenal dengan keunggulan akademis dan kontribusinya dalam teknologi informasi.

Di luar kelas, Arya mengejar berbagai hobi yang memperkaya wawasan dan keterampilannya, seperti mendengarkan musik, membaca buku, dan yang terutama adalah mengeksplorasi kemajuan terbaru dalam teknologi. Kecintaannya terhadap musik dan buku membuka perspektif baru dan menambah kedalaman pemahamannya tentang dunia.

Arya telah terlibat dalam sejumlah proyek yang menunjukkan keahliannya dalam pengembangan backend menggunakan Node.js, serta pengalaman dalam DevOps dan cloud computing dengan menggunakan platform Google Cloud. Keahliannya tidak terhenti di situ; ia juga mengembangkan kemampuan dalam bidang machine learning dan robotic operating system, yang menunjukkan keberanian dan kecakapan teknisnya dalam menghadapi tantangan baru.

Lebih lanjut, Arya juga mendalami pengembangan frontend menggunakan React JS, mengintegrasikan desain yang responsif dengan pengalaman pengguna yang intuitif. Terbaru, ia mulai mengeksplorasi ranah blockchain dan smart contract dengan menggunakan Solidity, menandai langkah selanjutnya dalam perjalanan teknologinya.

Dengan basis pengetahuan yang luas dan terus berkembang, Arya berambisi untuk menjadi inovator dan pemimpin di dunia teknologi, menggunakan setiap proyek dan kesempatan pembelajaran sebagai batu loncatan menuju keberhasilan yang lebih besar.

[Halaman ini sengaja dikosongkan]