

BAB III

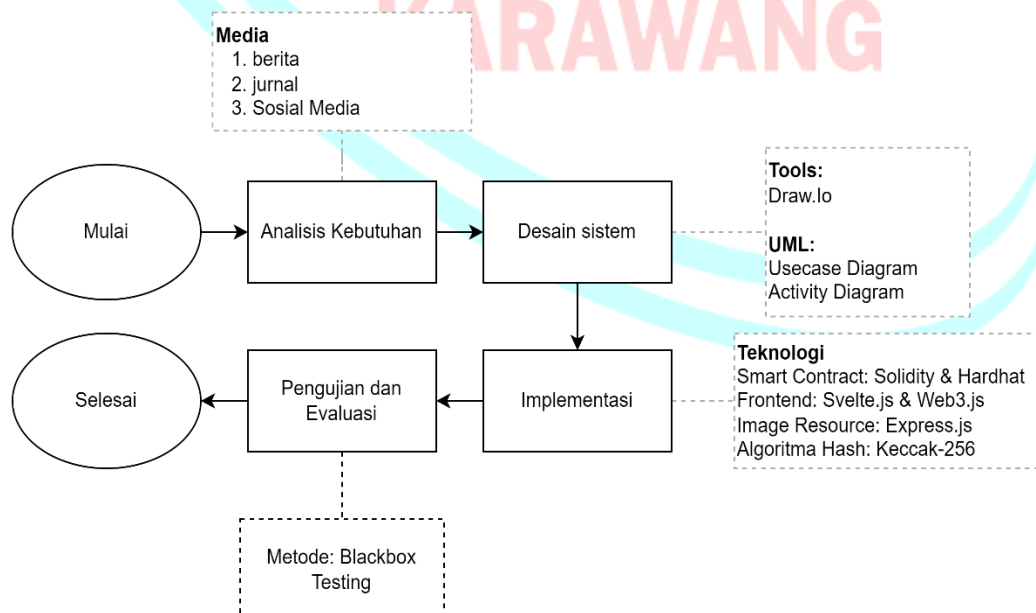
METODE PENELITIAN

3.1 Objek Penelitian

Objek Penelitian ini berfokus pada penerapan blockchain dalam platform *Crowdfunding* dengan Ethereum, yang memanfaatkan smart contract untuk penggalangan dana secara transparan. Setiap penggalangan dana diidentifikasi dengan ID unik yang dihasilkan menggunakan Keccak-256 hash dari kombinasi nama, deskripsi, gambar, target dana, dan deadline. Hash ini digunakan sebagai key dalam data index mapping untuk memastikan setiap penggalangan dana memiliki identifikasi yang unik.

3.2 Prosedur Penelitian

Prosedur penelitian ini disusun untuk merancang, mengimplementasikan, dan menguji platform *Crowdfunding* berbasis blockchain Ethereum. Tahapan penelitian meliputi analisis kebutuhan, pengembangan, implementasi dan evaluasi sistem. Diagram alur proses utama digambarkan untuk memvisualisasikan setiap langkah secara jelas.



Gambar 3. 1 Prosedur Penelitian

3.2.1 Analisis Kebutuhan

Crowdfunding telah menjadi solusi populer untuk penggalangan dana secara online (Safira Hasna & Irwansyah, 2019). Namun, berita menunjukkan bahwa masih sering terjadi penyalahgunaan dana, kurangnya transparansi dalam pengelolaan, hingga kasus penipuan pada platform *Crowdfunding* (Rohmah, 2024). Hal ini menurunkan kepercayaan masyarakat terhadap platform yang ada. Sistem ini dirancang untuk mengatasi permasalahan tersebut menggunakan teknologi blockchain. Berikut adalah kebutuhan yang diidentifikasi:

3.2.2.1 Identifikasi Masalah

Berdasarkan penelitian menurut (Asshidiq, 2023), platform *Crowdfunding* menghadapi beberapa tantangan utama yang memengaruhi kepercayaan pengguna dan efektivitas sistem. Tantangan-tantangan tersebut meliputi:

1. Kasus Penipuan dan Penyalahgunaan Dana
2. Kurangnya Transparansi Penggunaan Dana
3. Tidak Adanya Mekanisme Efektif untuk Mencegah Penyalahgunaan Dana.

3.2.2.2 Kebutuhan Fungsional

Kebutuhan fungsional merujuk pada fungsi inti yang harus dimiliki oleh sistem *Crowdfunding* berbasis blockchain. kebutuhan fungsional umum pada platform seperti ini meliputi:

1. Pembuatan Penggalangan Dana

Sistem harus memungkinkan pengguna untuk membuat kampanye penggalangan dana dengan memasukkan data seperti nama kampanye, deskripsi, target dana, dan batas waktu. Hal ini sesuai dengan penelitian oleh (Yasar, 2021), yang menekankan pentingnya fleksibilitas dalam pembuatan kampanye untuk meningkatkan partisipasi pengguna.

2. Transaksi Donasi yang Aman

Setiap transaksi harus dicatat di blockchain untuk menjamin keamanan dan transparansi, seperti yang diungkapkan oleh (Khan et al., 2021). Penggunaan hashing Keccak-256 juga memberikan tingkat

keamanan tambahan pada integrasi data untuk memastikan data dapat diaudit jika terjadi manipulasi.

3. Monitoring dan Pelaporan Transparan

Pengguna harus dapat memantau status kampanye secara *real-time*, termasuk dana terkumpul dan total donatur. Berdasarkan (Asshidiq, 2023) menekankan bahwa transparansi dalam laporan meningkatkan kepercayaan pengguna terhadap platform.

3.2.2.3 Kebutuhan Non-Fungsional

Kebutuhan non-fungsional sangat penting untuk memastikan efektivitas dan keamanan sistem. Berdasarkan hal tersebut, kebutuhan non-fungsional sistem ini meliputi:

1. Keamanan

Sistem menggunakan hashing Keccak-256 untuk menjaga integritas data, sesuai rekomendasi standar keamanan data dalam pengembangan blockchain (Frikha et al., 2021; Müller, 2024).

2. Efisiensi

Sistem dirancang tanpa menggunakan database tradisional seperti NoSQL, dan data dicatat langsung di blockchain untuk mengurangi kompleksitas serta meningkatkan efisiensi pencatatan (Wang et al., 2020).

3. Antarmuka Pengguna (UI)

Antarmuka yang ramah pengguna dirancang menggunakan *framework frontend* Svelte.js, yang terkenal karena kecepatan rendering dan efisiensinya dalam aplikasi web (Scherer & Safari, 2020).

3.2.2.4 Teknologi yang digunakan

Menurut berbagai penelitian Teknologi yang cocok digunakan dalam sistem ini meliputi:

1. Solidity & Hardhat

Solidity digunakan untuk pengembangan smart contract, sementara Hardhat digunakan sebagai *framework* untuk pengujian dan debugging smart contract. Kombinasi ini sering direkomendasikan

dalam pengembangan berbasis Ethereum karena kemudahan penggunaannya (Hardhat, n.d.; Kumar & Kumar, 2022)

2. Express.js

Framework ini dipilih untuk menangani penyimpanan gambar secara tanpa memanfaatkan layanan berbasis IPFS. Pendekatan ini disarankan untuk kebutuhan penyimpanan lebih mudah karena umum dan mudah saat proses pengembangan (Nguyen, 2022).

3. Svelte.js & Web3.js

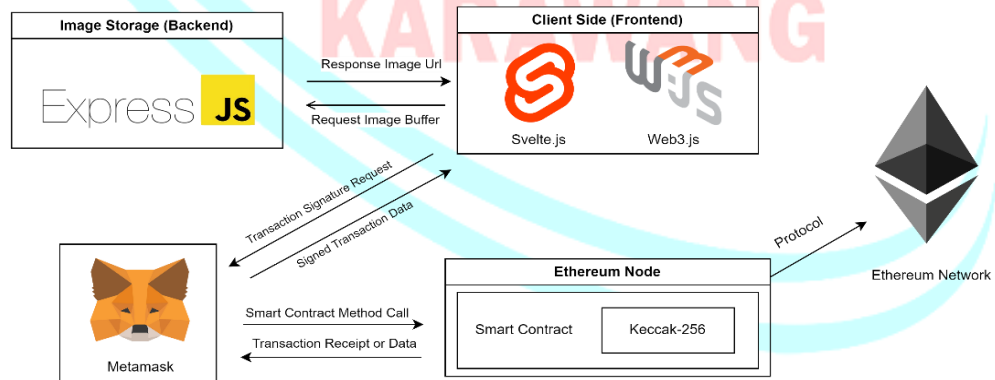
Svelte.js digunakan sebagai *framework frontend* karena kemampuannya yang efisien dalam rendering UI, sementara Web3.js memfasilitasi interaksi dengan smart contract di blockchain Ethereum (Kurada & Baruah, 2023)

4. Algoritma Keccak-256

Algoritma ini digunakan untuk hashing, yang berfungsi menjaga integritas dan keamanan data. Hal ini sejalan dengan praktik terbaik dalam pengelolaan data berbasis blockchain (Frikha et al., 2021).

3.2.2 Desain Sistem

3.2.3.1 Gambaran Umum Sistem



Gambar 3. 2 Gambaran Umum Sistem

Diagram ini menjelaskan arsitektur sistem *Crowdfunding* berbasis blockchain dengan peran terpisah untuk setiap komponen. *Backend* menggunakan Express.js untuk menyimpan gambar. *Frontend* (Svelte dan Web3.js) mengirimkan gambar ke *backend* sebagai request buffer dan menerima URL gambar sebagai respons. Interaksi dengan blockchain terjadi

melalui Metamask, yang menghubungkan *frontend* ke Ethereum Node. Di Ethereum Node, smart contract memproses transaksi, dan Keccak-256 digunakan untuk hashing ID penggalangan dana agar unik dan aman. Hasil pemrosesan dikirim kembali ke *frontend* melalui Metamask.

3.2.3.2 Desain Smart Contract

Smart Contract merupakan komponen inti dalam sistem *Crowdfunding* berbasis blockchain. Fungsi-fungsi dalam smart contract mencakup pembuatan kampanye, penyimpanan donasi, pencairan dana, dan mitigasi risiko penipuan. Tabel berikut menjelaskan rincian fungsi-fungsi utama dalam smart contract yang diimplementasikan.

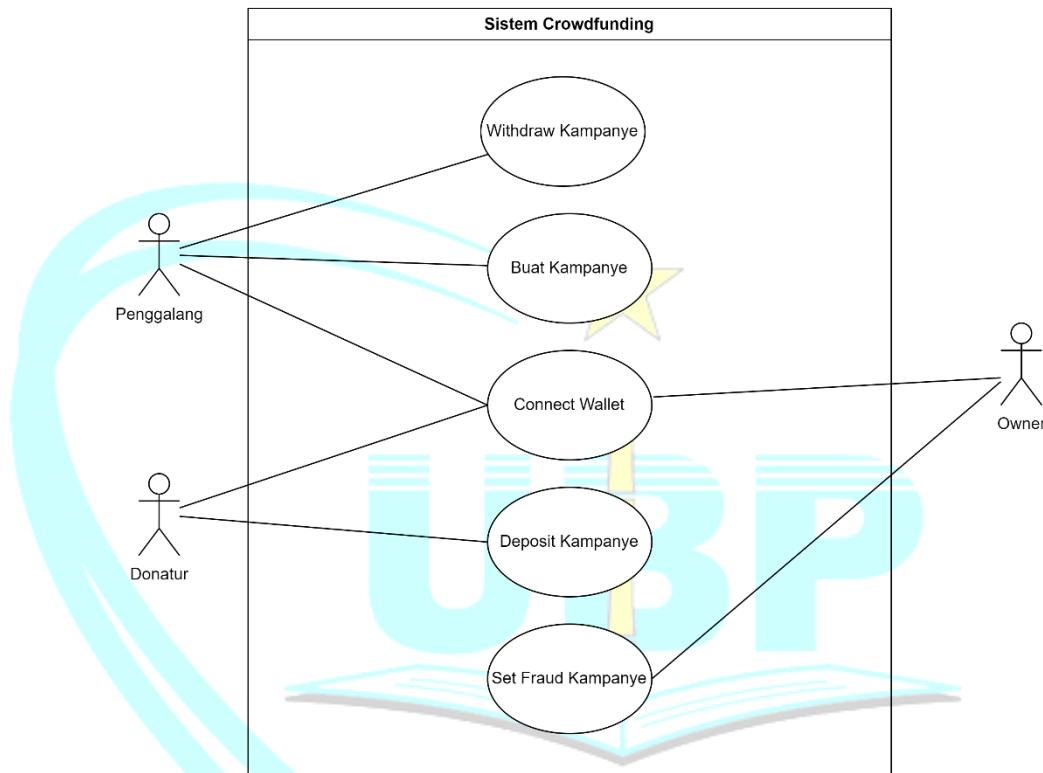
Tabel 3. 1 Desain Smart Contract

Komponen	Tipe Data	Deskripsi
Penggalang	Struct	menyimpan informasi penggalangan dana, seperti nama, deskripsi, target, dan status.
IDonatur	Struct	menyimpan informasi donatur, seperti alamat donatur dan jumlah donasi.
owner	address	Alamat pemilik kontrak
GalangData	mapping(bytes32 => Penggalang)	Menyimpan data penggalangan dana berdasarkan ID unik (hash Keccak-256)
galangIds	bytes32[]	Array untuk menyimpan ID penggalangan dana yang telah dibuat.
GalangDataLen gth	uint256	Menghitung jumlah total penggalangan dana yang telah dibuat.
donatur	mapping(bytes32 => IDonatur[])	Menyimpan daftar donatur untuk setiap penggalangan dana berdasarkan ID unik.

Komponen	Tipe Data	Deskripsi
createGalang	public function, payable	Membuat penggalangan dana baru dengan parameter seperti nama, deskripsi, target, dan deadline.
deposit	public function, payable	Menyumbang dana ke penggalangan yang ditentukan berdasarkan ID, dengan validasi target dan pengembalian jika melampaui target
withdraw	public function	Penarikan dana oleh penggalang setelah tenggat waktu berakhir.
FraudDonation	public function, onlyOwner	Menandai penggalangan dana sebagai penipuan dan mengembalikan semua dana ke donatur (hanya dapat dilakukan oleh pemilik kontrak).
getGalangData	public function, view	Mengembalikan semua data penggalangan dana yang telah dibuat.
getDonatur	public function, view	Mengembalikan daftar donatur untuk penggalangan dana tertentu berdasarkan ID.
GalangCreated	event	log emit saat penggalangan baru dibuat
Deposited	event	log emit saat ada donasi masuk
Withdrawn	event	log emit saat dana berhasil ditarik

3.2.3.3 Usecase Diagram

Use case diagram ini memvisualisasikan interaksi antara aktor dan sistem pada platform *Crowdfunding* berbasis blockchain, mencakup fungsi utama dan alur kerja yang mendukung keamanan serta transparansi.

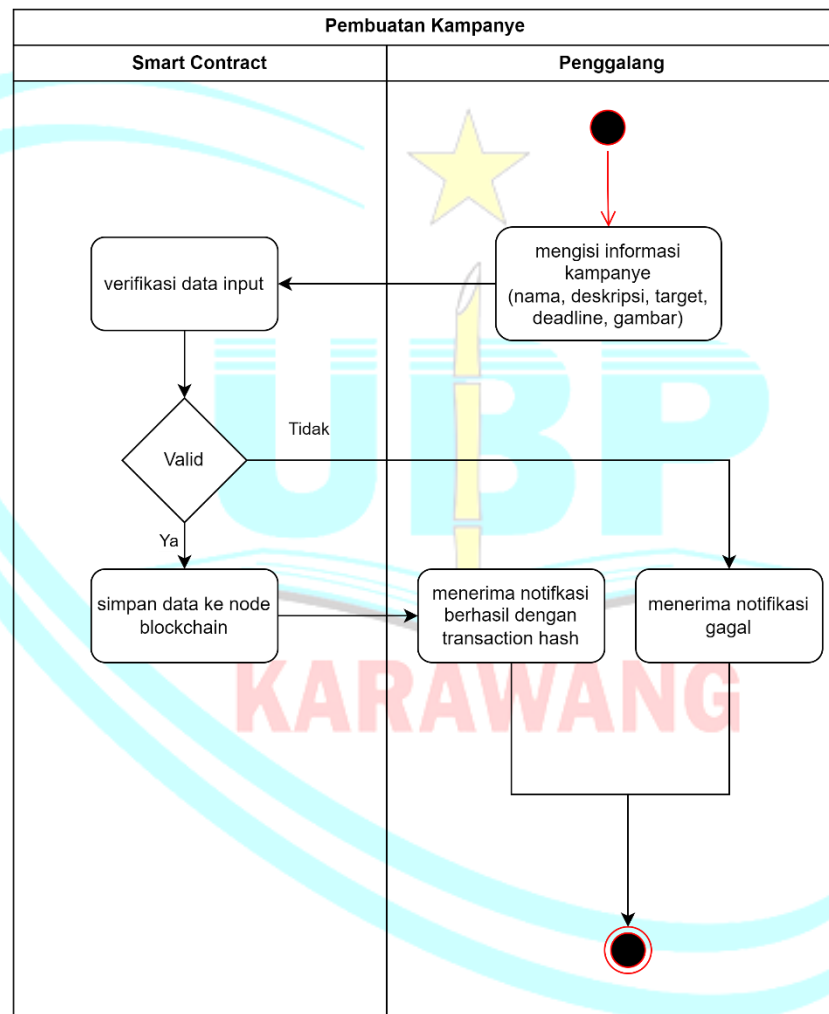


Gambar 3. 3 Usecase Diagram Sistem *Crowdfunding*

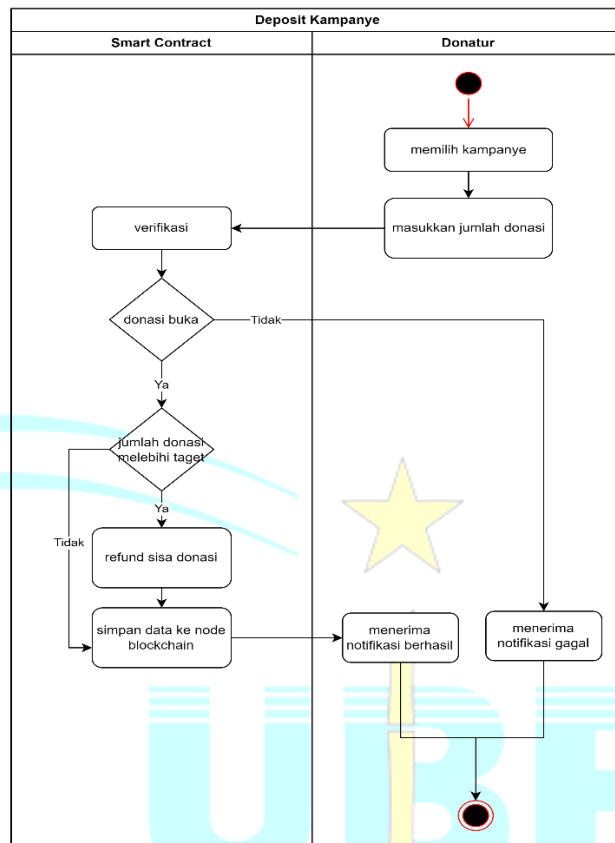
Usecase Diagram ini menjelaskan sistem *Crowdfunding* berbasis blockchain Ethereum dengan tiga aktor utama: Donatur, Penggalang, dan Owner. Proses dimulai ketika Donatur atau Penggalang menghubungkan dompet Ethereum melalui fitur Connect Wallet. Setelah itu, Penggalang dapat membuat kampanye baru menggunakan fitur Buat Kampanye. Donatur mendukung kampanye dengan menyumbangkan dana melalui Deposit Kampanye. Untuk menjaga transparansi, Owner atau pihak berwenang dapat menandai kampanye yang dicurigai sebagai penipuan melalui fitur Set Fraud Kampanye. Jika dana terkumpul sesuai target, Penggalang dapat menarik dana menggunakan fitur Withdraw Kampanye. Semua transaksi tercatat di blockchain untuk memastikan keamanan dan transparansi sistem.

3.2.4.1 Activity Diagram

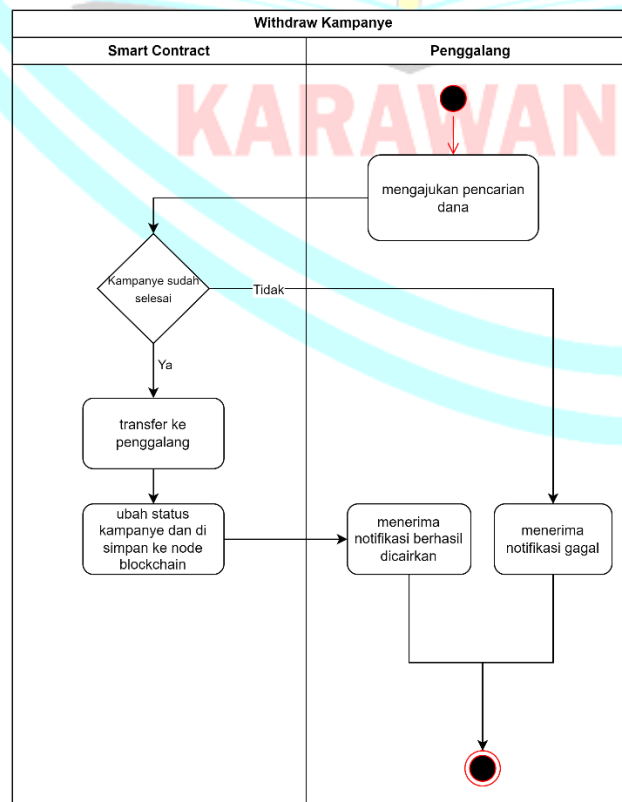
Sistem *Crowdfunding* mencakup beberapa aktivitas utama yang dirancang untuk mendukung transparansi, keamanan, dan efisiensi pengelolaan penggalangan dana. Aktivitas-aktivitas ini melibatkan interaksi antara pengguna (penggalang dana, donatur), sistem *backend*, dan blockchain sebagai media pencatatan data.



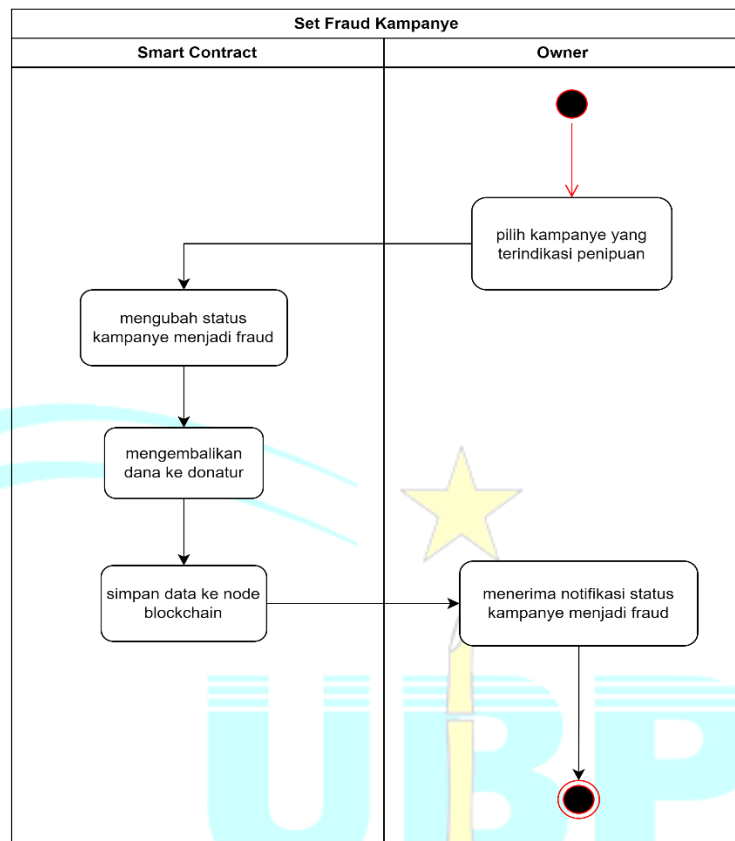
Gambar 3. 4 Activity Diagram Buat Kampanye



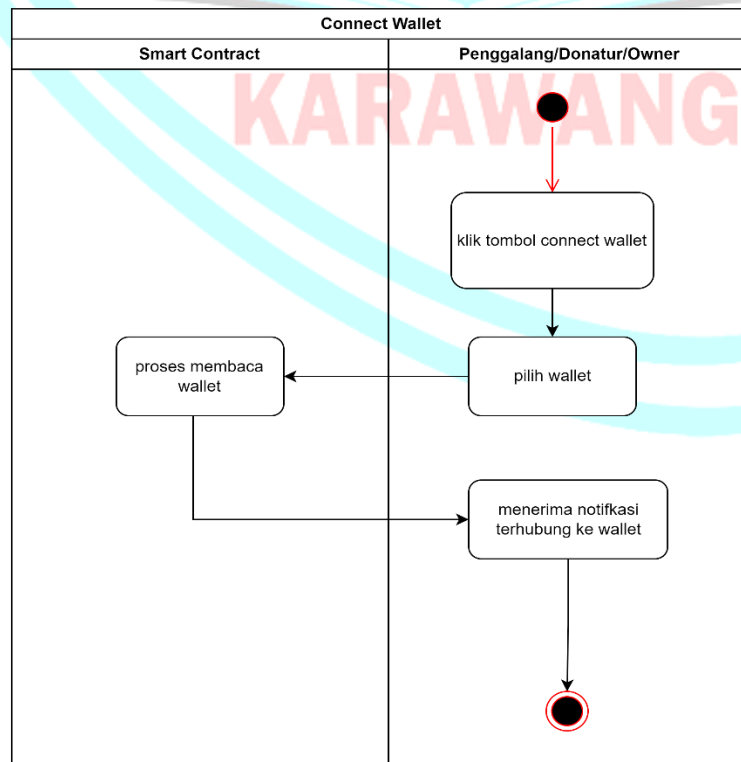
Gambar 3. 5 Activity Diagram Deposit Kampanye



Gambar 3. 6 Activity Diagram Withdraw Kampanye



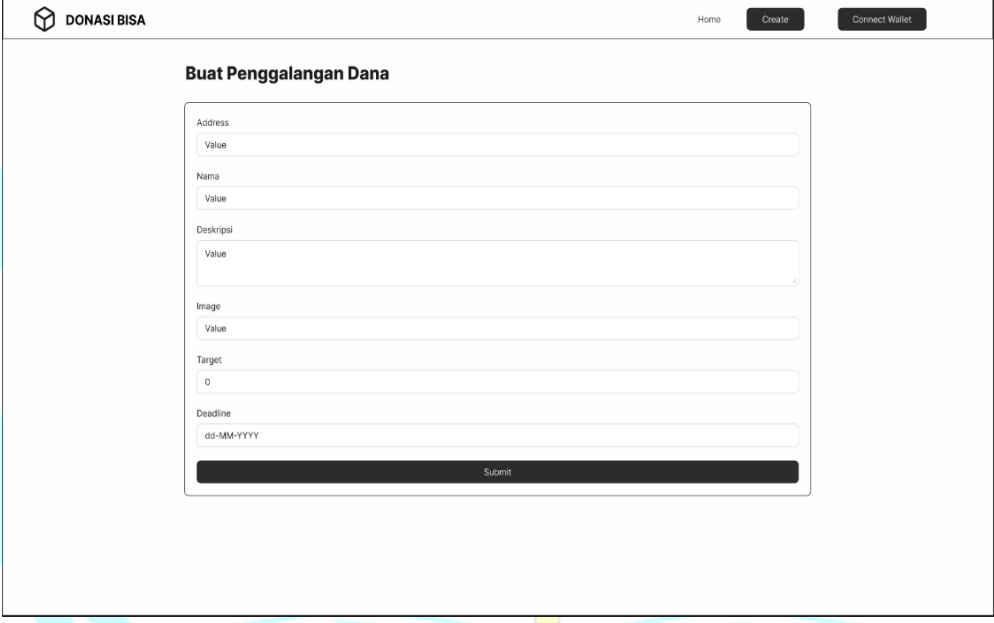
Gambar 3. 7 Activity Diagram Set Fraud Kampanye



Gambar 3. 8 Activity Diagram Connect Wallet

3.2.3.4 User Interface

User Interface (UI) dirancang untuk memudahkan pengguna dalam membuat kampanye, melihat daftar kampanye, dan berdonasi. Prototipe UI ini fokus pada kemudahan navigasi, penyajian informasi kampanye secara jelas, serta mendukung transparansi dalam penggalangan dana.

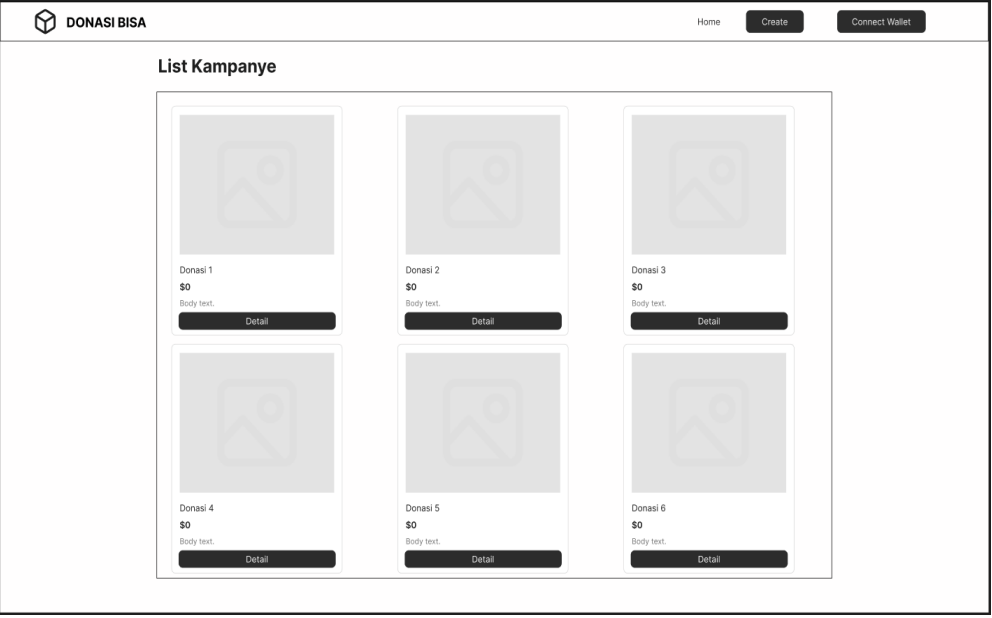


The screenshot shows the 'Buat Penggalangan Dana' (Create Fundraising) form in the DONASI BISA application. The form is titled 'Buat Penggalangan Dana' and includes the following fields:

- Address:** A text input field with a placeholder 'Value'.
- Nama:** A text input field with a placeholder 'Value'.
- Deskripsi:** A text input field with a placeholder 'Value'.
- Image:** A text input field with a placeholder 'Value'.
- Target:** A text input field with a placeholder '0'.
- Deadline:** A text input field with a placeholder 'dd-MM-YYYY'.

At the bottom of the form is a 'Submit' button. The top navigation bar includes the 'DONASI BISA' logo, a 'Home' link, a 'Create' button, and a 'Connect Wallet' button.

Gambar 3. 9 UI Buat Penggalangan Dana



The screenshot shows the 'List Kampanye' (Campaign List) in the DONASI BISA application. The list displays six campaigns, each with a placeholder image, a title, a status, and a 'Detail' button.

Donasi 1	Donasi 2	Donasi 3	Donasi 4	Donasi 5	Donasi 6
\$0	\$0	\$0	\$0	\$0	\$0
Body text.	Body text.	Body text.	Body text.	Body text.	Body text.
Detail	Detail	Detail	Detail	Detail	Detail

The top navigation bar includes the 'DONASI BISA' logo, a 'Home' link, a 'Create' button, and a 'Connect Wallet' button.

Gambar 3. 10 UI List Kampanye

DONASI BISA Home Create Connect Wallet

Detail Kampanye



Address Penggalang : 0x70997970C3B12dc3A010C7d01b50e0d79c79C8

Nama : Text

Target : 20 ETH

Terkumpul : 3 ETH

Deskripsi : Lorem ipsum, or ipsum as it is sometimes known, is dummy text used in laying out print, graphic or web designs.

Deadline : 31/10/2024

Status : Belum Tercapai

0 Donasi Sekarang

Donatur

No	Address Donatur	Jumlah Donasi
#20462	0x70997970C3B12dc3A010C7d01b50e0d79c79C8	0.15 ETH
#34304	0x70997970C3B12dc3A010C7d01b50e0d79c79C8	0.15 ETH
#20462	0x70997970C3B12dc3A010C7d01b50e0d79c79C8	0.15 ETH
#34304	0x70997970C3B12dc3A010C7d01b50e0d79c79C8	0.15 ETH
#20462	0x70997970C3B12dc3A010C7d01b50e0d79c79C8	0.15 ETH
#34304	0x70997970C3B12dc3A010C7d01b50e0d79c79C8	0.15 ETH
#20462	0x70997970C3B12dc3A010C7d01b50e0d79c79C8	0.15 ETH

Gambar 3. 11 UI Detail Kampanye

3.2.3 Implementasi

Pada tahap implementasi, sistem *Crowdfunding* berbasis blockchain ini dikembangkan menggunakan teknologi dan metode yang telah direncanakan sebelumnya. Proses implementasi melibatkan pengembangan *frontend*, *backend*, dan smart contract, yang saling berintegrasi untuk menyediakan layanan penggalangan dana.

3.2.2.1 Implementasi Frontend

Frontend dikembangkan menggunakan Svelte.js, sebuah *framework* JavaScript yang ditulis dengan Svelte Compiler. Beberapa fitur yang diimplementasikan meliputi:

1. Buat Penggalangan Dana, Form untuk memasukkan nama, deskripsi, target dana, gambar, dan batas waktu.
2. List Kampanye, Halaman yang menampilkan kampanye penggalangan dana secara dinamis dengan data yang diambil langsung dari blockchain.
3. Detail Kampanye, Informasi terperinci mengenai kampanye tertentu, termasuk jumlah dana terkumpul, jumlah donatur, dan status penggalangan.

Frontend berkomunikasi dengan smart contract melalui *library* Web3.js untuk membaca dan mengirimkan data ke blockchain.

3.2.2.2 Implementasi Backend

Backend dikembangkan menggunakan Express.js dengan bahasa pemrograman Typescript untuk menangani pengelolaan file gambar yang diunggah pengguna.

Gambar kampanye disimpan dalam server lokal menggunakan middleware seperti Multer, memastikan integritas file yang diunggah. *Backend* tidak menggunakan database seperti NoSQL, data kampanye sepenuhnya dikelola oleh smart contract dan blockchain.

3.2.2.3 Implementasi Smart Contract

Smart Contract dikembangkan dengan bahasa Solidity menggunakan Hardhat sebagai lingkungan pengembangan. Beberapa fitur penting meliputi:

1. Pembuatan Kampanye, Data kampanye di-hash menggunakan algoritma Keccak-256 untuk menghasilkan ID unik.
2. Pencatatan Donasi, Fungsi untuk mencatat setiap donasi, menyimpan data donatur dan jumlah yang disumbangkan.
3. Pencairan Dana, Penggalang dana dapat mencairkan dana setelah kampanye selesai sesuai aturan yang telah ditetapkan.

Smart Contract diuji menggunakan Hardhat *Testing Framework* untuk memastikan fungsionalitas berjalan sesuai dengan spesifikasi.

3.2.2.4 Integrasi Sistem

Bagian ini menjelaskan bagaimana komponen sistem terhubung untuk mendukung fungsionalitas utama:

1. Otentikasi dan Transaksi

Pengguna dapat menghubungkan akun Ethereum mereka menggunakan MetaMask, yang berfungsi sebagai perantara antara aplikasi *frontend* dan blockchain. MetaMask digunakan untuk mengotorisasi transaksi seperti donasi atau pembuatan kampanye.

2. Komunikasi dengan Blockchain

Aplikasi frontend yang dibuat dengan Svelte.js menggunakan library Web3.js untuk mengirim dan menerima data dari blockchain. Proses ini mencakup transaksi dan pengambilan informasi kampanye.

3. Keamanan Data

Sistem memanfaatkan hashing Keccak-256 untuk memastikan integritas data. Hash ini digunakan sebagai indeks unik untuk setiap penggalangan.

Bagian ini memastikan semua komponen bekerja bersama untuk menjalankan sistem secara terintegrasi.

3.2.4 Pengujian Sistem

Pengujian sistem bertujuan untuk memastikan bahwa platform *Crowdfunding* berbasis blockchain berfungsi dengan baik sesuai dengan tujuan yang telah ditentukan. Pengujian dilakukan menggunakan metode *blackbox testing*, yang berfokus pada pengujian fungsionalitas sistem tanpa mempertimbangkan struktur internal atau kode sumbernya. Berdasarkan penelitian (Santi et al., 2022), pengujian *blackbox* sangat efektif untuk mengevaluasi apakah sistem memenuhi persyaratan yang telah ditetapkan, terutama dalam pengujian fungsionalitas antarmuka pengguna.

Dalam penelitian ini, pengujian dilakukan untuk mengevaluasi beberapa aspek utama sebagai berikut:

1. Verifikasi Smart Contract, menguji apakah smart contract yang diterapkan dapat berfungsi sesuai ketentuan, seperti pembuatan penggalangan dana, pengumpulan dana, hingga penarikan dana oleh penggalang dana.
2. Keamanan dan Keaslian Data, memastikan bahwa data transaksi yang disimpan di blockchain tidak dapat dimanipulasi serta menguji penerapan algoritma Keccak-256 untuk memvalidasi integritas data.
3. Interaksi Pengguna, menguji bagaimana pengguna berinteraksi dengan platform melalui tampilan *frontend*, termasuk pembuatan penggalangan dana, menyumbang, dan menarik dana; Hal ini memastikan sistem memberikan *user experience* yang mudah dipahami dan digunakan.

4. Transparansi, menguji kemampuan sistem untuk memberikan informasi yang jelas terkait status penggalangan dana, jumlah donasi yang diterima, dan proses penarikan dana. Transparansi ini bertujuan untuk meningkatkan kepercayaan pengguna terhadap platform.

Setiap pengujian dilakukan dengan skenario berbeda untuk mengidentifikasi potensi masalah dan memastikan semua fitur berfungsi dengan baik. Hasil pengujian ini digunakan sebagai dasar untuk evaluasi sistem sebelum deployment.

Selain itu, dilakukan juga pengamatan terhadap gas fee pada setiap jenis transaksi sebagai bagian dari analisis efisiensi sistem. Estimasi biaya transaksi ini diperoleh menggunakan Hardhat Gas Reporter pada jaringan lokal, dan tetap diuji melalui blackbox untuk mengetahui seberapa besar beban biaya yang ditanggung pengguna dalam menjalankan fitur-fitur utama sistem.

