

BAB III

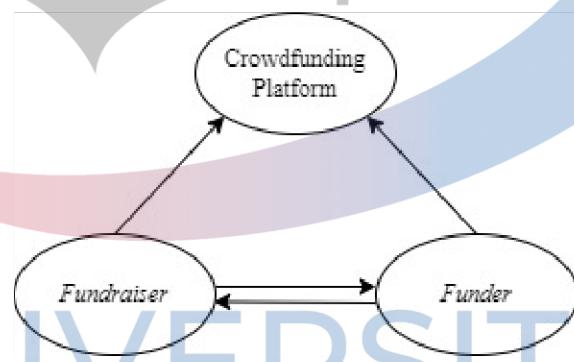
TAHAPAN PELAKSANAAN

3.1 Analisis

Tahapan analisis terdiri dari analisis masalah, analisis proses dan analisis kebutuhan. Analisis masalah akan membahas konsep kerja dan masalah *crowdfunding platform* tanpa *blockchain*. Analisis proses akan membahas cara kerja *smart contract* yang akan dirancang untuk menyelesaikan permasalahan yang ada. Analisis kebutuhan terdiri dari analisis kebutuhan fungsional dan kebutuhan non fungsional aplikasi.

3.1.1 Analisis Masalah

Analisis masalah terdiri dari analisis mengenai *crowdfunding platform* yang tidak mengimplementasikan *blockchain*. Pada umumnya, *crowdfunding platform* memiliki konsep yang diilustrasikan pada Gambar 3.1.



Gambar 3.1 Konsep *Crowdfunding Platform*

Fundraiser akan membuat *campaign* melalui *crowdfunding platform* dengan menyertakan informasi mengenai proyek yang ingin mereka buat yang kemudian diserahkan ke *platform*. Jika *platform* menyetujui pembuatan *campaign* tersebut, maka *campaign* akan muncul di laman *platform* untuk mempertemukan *fundraiser* dengan calon *funder*-nya. Setelahnya, para *funder* akan menganalisis dan mempertimbangkan peluang berkembangnya *campaign* tersebut. *Funder* yang melihat adanya kemungkinan *campaign* untuk berkembang bahkan sukses dapat memberikan kontribusi berupa dana pada *campaign* tersebut.

Interaksi yang dilakukan antara *crowdfunding platform*, *fundraiser* dan *funder* seperti demikian masih rentan masalah. Pertama, kurangnya transparansi data. Hal ini dikarenakan kebijakan setiap *crowdfunding platform* berbeda, ada yang tidak menampilkan data riwayat pencairan dana kepada *fundraiser* untuk dibaca oleh *funder*, dan ada yang tidak mewajibkan *fundraiser* untuk melampirkan penggunaan dana yang telah diterima sehingga

funder tidak tahu untuk apa saja dana itu digunakan. Kedua, karena *platform* bersifat tersentralisasi, keamanan data belum terjamin. *Platform* memiliki otoritas penuh atas segala proses yang terjadi di dalamnya, hanya pihak *platform* dan *fundraiser* yang memiliki kontrol untuk mencairkan dana sehingga ada kemungkinan terjadinya penyalahgunaan kekuasaan oleh pihak *platform* ataupun *fundraiser* itu sendiri dengan memanipulasi data *campaign* ataupun data transaksi. Data yang disimpan secara sentralisasi juga rentan hilang atau rusak jika terjadi kerusakan pada *server* karena data hanya tersimpan pada satu tempat. Masalah pertama dan kedua menyebabkan kepercayaan *funder* terhadap *platform* ataupun *fundraiser* semakin memudar.

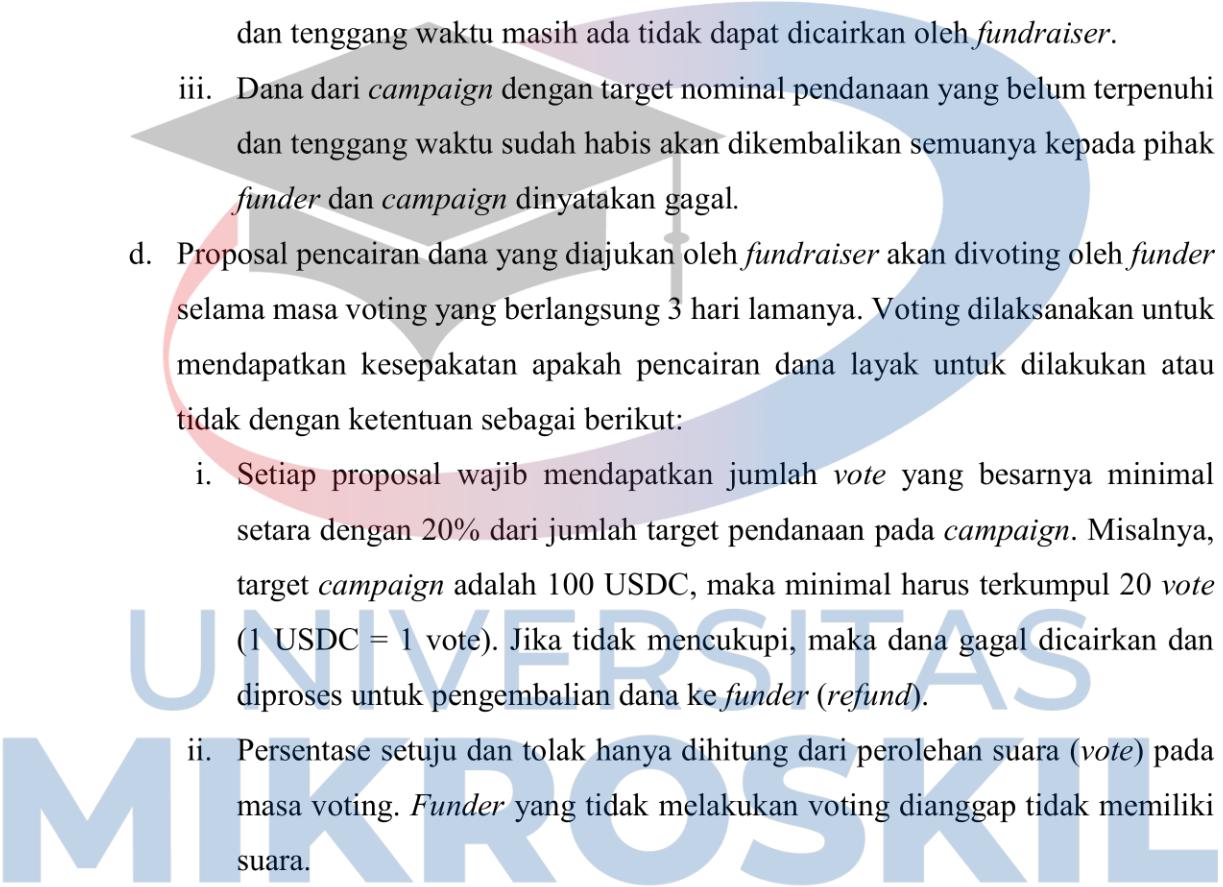
Selain kedua masalah tersebut yang berkaitan dengan data, ada masalah lain yaitu persoalan dana. Dana yang telah dikumpulkan *campaign* tidak mudah dicairkan, yang berarti distribusi dana ke *fundraiser* akan lambat. Setelah target pendanaan *campaign* tercapai, *fundraiser* tentu akan melakukan permintaan pencairan dana kepada *platform*. Namun, *platform* membutuhkan waktu untuk memverifikasi permintaan pencairan dana tersebut yang umumnya dapat mencapai 7 hari kerja lamanya. Hal ini dikarenakan proses verifikasi masih dilakukan secara manual oleh pihak *platform*. Oleh karena itu, *fundraiser* tidak dapat mengerjakan proyek *campaign*-nya dan perlu menunggu hingga dana telah dicairkan oleh *platform*.

Dengan pemanfaatan *blockchain*, ketiga masalah tersebut dapat diatasi karena *blockchain* merupakan jaringan yang memiliki sifat terdesentralisasi dan transparan. Ini berarti segala data yang disimpan di dalam jaringan akan dapat dilihat oleh siapapun namun tidak ada yang dapat memodifikasinya jika tidak diberikan akses. Serta data disimpan tidak hanya pada satu tempat sehingga jika ada kerusakan pada *server* utama maka akan ada *server* cadangan lain yang memiliki data tersebut. Hal ini menjamin transparansi dan keamanan data dari transaksi yang terjadi pada *platform* yang merupakan penyelesaian untuk masalah pertama dan kedua. Kemudian, masalah ketiga dapat diatasi oleh *smart contract* milik *blockchain*. *Smart contract* merupakan program yang dieksekusi secara otomatis sehingga jika ada pengajuan pencairan dana oleh *fundraiser* dan syarat pencairan dana telah terpenuhi, maka *smart contract* akan langsung dieksekusi dan dana akan langsung diterima oleh *fundraiser*.

Pada solusi aplikasi yang ditawarkan akan memiliki fitur voting untuk *funder*. Hal ini dilakukan agar para *funder* tidak hanya berpartisipasi dalam memberi atau donasi saja, namun juga dapat berpartisipasi dalam kontrol pencairan dana. Dengan menerapkan *blockchain*, fitur ini akan berjalan berdasarkan *smart contract* sehingga dapat dipercaya.

Pada aplikasi ini, ada batasan yang ditetapkan untuk menunjang keamanan dan kenyamanan *user* saat menggunakan aplikasi yaitu:

1. Model *crowdfunding* yang digunakan adalah *donation-based* dengan tipe *all or nothing* (AON) sehingga jika target dana yang dikumpulkan *campaign* tidak terpenuhi, maka dana tidak dapat dicairkan dan akan dikembalikan ke *funder*.
2. Jumlah aktor ada 3 (tiga) yaitu admin, *fundraiser* dan *funder*. Seorang *fundraiser* dapat menjadi *funder* dan begitu juga sebaliknya. Aksi-aksi yang dapat dilakukan oleh setiap aktor berupa:
 - a. Admin
 - i. Menguji aplikasi
 - ii. Mengawasi aktivitas pada aplikasi
 - iii. Memvalidasi kesuksesan *campaign*
 - iv. Memverifikasi data *user*
 - v. Mengelola *user*
 - vi. Mengelola pengaduan *user*
 - b. *Fundraiser*
 - i. Membuat *campaign*
 - ii. Mengajukan pencairan dana
 - iii. Mengajukan verifikasi hasil penggerjaan *campaign*
 - iv. Melihat riwayat pembuatan *campaign*
 - v. Mengedit data diri
 - vi. Mengajukan verifikasi data diri
 - c. *Funder*
 - i. Melakukan donasi
 - ii. Voting proposal pencairan dana
 - iii. Mengklaim dana *refund*
 - iv. Melihat riwayat transaksi
 - v. Mengedit data diri
 - vi. Mengajukan verifikasi data diri
 - vii. Mengirim pengaduan
3. Aktor admin tidak dapat ditambahkan atau dimodifikasi, hanya berdasarkan *database*
4. Alur transaksi antara *fundraiser* dan *funder* berupa:
 - a. Transaksi dimulai dari pembuatan *campaign* oleh *fundraiser*.

- 
- b. *Funder* yang tertarik dengan *campaign* tersebut dapat memberikan donasi setelah menghubungkan *wallet*-nya.
 - c. Jika *campaign* menyentuh target pendanaan sebelum maupun pada saat tenggang waktu, maka pihak *fundraiser* wajib melakukan pengajuan pencairan dana. Adapun aturan pencairan pendanaan terdiri dari:
 - i. Dana dari *campaign* dengan target nominal pendanaan yang telah terpenuhi dan tenggang waktu masih ada atau sudah habis dapat dicairkan oleh *fundraiser* dengan mengajukan proposal permintaan pencairan dana.
 - ii. Dana dari *campaign* dengan target nominal pendanaan yang belum terpenuhi dan tenggang waktu masih ada tidak dapat dicairkan oleh *fundraiser*.
 - iii. Dana dari *campaign* dengan target nominal pendanaan yang belum terpenuhi dan tenggang waktu sudah habis akan dikembalikan semuanya kepada pihak *funder* dan *campaign* dinyatakan gagal.
 - d. Proposal pencairan dana yang diajukan oleh *fundraiser* akan divoting oleh *funder* selama masa voting yang berlangsung 3 hari lamanya. Voting dilaksanakan untuk mendapatkan kesepakatan apakah pencairan dana layak untuk dilakukan atau tidak dengan ketentuan sebagai berikut:
 - i. Setiap proposal wajib mendapatkan jumlah *vote* yang besarnya minimal setara dengan 20% dari jumlah target pendanaan pada *campaign*. Misalnya, target *campaign* adalah 100 USDC, maka minimal harus terkumpul 20 *vote* (1 USDC = 1 *vote*). Jika tidak mencukupi, maka dana gagal dicairkan dan diproses untuk pengembalian dana ke *funder* (*refund*).
 - ii. Persentase setuju dan tolak hanya dihitung dari perolehan suara (*vote*) pada masa voting. *Funder* yang tidak melakukan voting dianggap tidak memiliki suara.
 - iii. Proposal dengan persentase setuju diatas 50% (persen) akan diproses untuk pencairan dana.
 - iv. Proposal dengan persentase setuju dibawah atau setara dengan 50% (persen) akan diproses untuk *refund*.
 - 5. Menggunakan *extension wallet* Phantom.
 - 6. Batas waktu masa *campaign* adalah 90 hari.
 - 7. Nominal donasi harus bulat dan batas minimal donasi adalah 1 USDC.
 - 8. Dana yang dikumpulkan oleh *campaign* tidak dapat melebihi target pendanaan yang telah ditetapkan saat pembuatan *campaign*.

9. Adapun kriteria aksi admin dalam mengelola *user* dan *campaign* yaitu:

a. Untuk *user*:

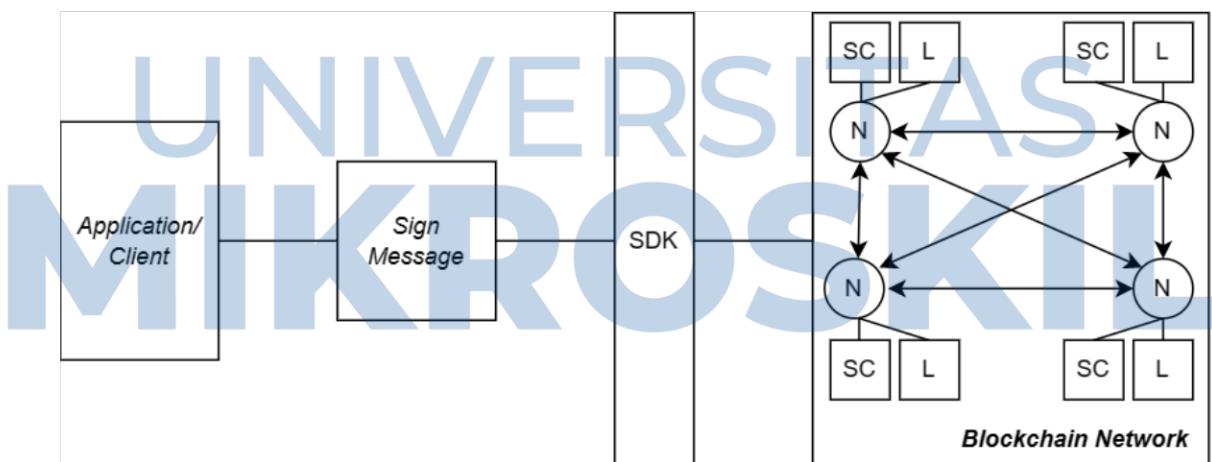
- i. Menyetujui permintaan verifikasi data KYC jika data telah dipastikan benar dan lengkap.
- ii. Menolak permintaan verifikasi data KYC jika data telah dipastikan salah atau lengkap.
- iii. Mencabut kembali status verifikasi data *user* jika admin salah melakukan aksi yang telah disebutkan sebelumnya.

b. Untuk *campaign*:

- i. Menyetujui permintaan verifikasi keberhasilan *campaign* jika data administrasi telah dipastikan benar dan lengkap.
- ii. Menolak permintaan verifikasi keberhasilan *campaign* jika data administrasi telah dipastikan salah atau tidak lengkap.

3.1.2 Analisis Proses

Analisis proses digunakan untuk menjelaskan arsitektur sistem, proses terbentuknya sebuah transaksi dan proses kerja tahapan transaksi yang memanfaatkan teknologi dari *smart contract* yang akan dirancang pada *blockchain platform* untuk menyelesaikan permasalahan yang ada.



Gambar 3.2 Arsitektur Sistem

Arsitektur sistem dari aplikasi yang akan dirancang dapat dilihat pada Gambar 3.2 dengan komponennya yaitu:

- *Application/Client*, merupakan antarmuka yang digunakan *user* untuk terhubung dengan sistem aplikasi.

- *Sign Message*, merupakan awal proses interaksi ke jaringan *blockchain* dimana sistem melakukan *hash* menggunakan *key* yang dimiliki *user* yang melakukan transaksi dengan penerapan enkripsi SHA-256.
- *Software Development Kit* (SDK), merupakan *library* yang digunakan untuk berkomunikasi dengan jaringan *blockchain*.
- *Blockchain Network* yang digunakan adalah jaringan Solana. Solana bekerja menggunakan algoritma *Proof-of-History*.
- *Node* (N), merupakan komputer yang terhubung ke jaringan *blockchain* dan menjalankan protokol *blockchain* sehingga memiliki kemampuan untuk membantu validasi transaksi yang masuk ke jaringan.
- *Smart Contract* (SC), merupakan komponen yang ada di dalam jaringan *blockchain* yang dipanggil setiap ada transaksi yang memanggil.
- *Ledger* (L), merupakan komponen yang ada di dalam jaringan *blockchain* sebagai tempat penyimpanan.

Dalam *blockchain* Solana yang digunakan, sebuah transaksi perlu dikirimkan ke jaringan *blockchain*-nya dalam bentuk pesan (*message*) agar dapat diproses. Masing-masing pesan memiliki atribut sebagai berikut:

- Instruksi
Atribut ini menentukan jenis aksi yang akan dijalankan pada pesan, seperti transfer koin/*token*, dsb.
- *Account Addresses*
Atribut ini berisi *account address* dari pihak yang terlibat dalam transaksi, seperti *address* milik pengirim, penerima dan juga *address* dari sistem program *smart contract* itu sendiri.
- Data
Atribut ini berisi data tambahan yang dibutuhkan oleh instruksi, seperti jumlah koin/*token* yang akan ditransfer, argumen fungsi *smart contract*, dsb.

Selain itu, transaksi pada Solana juga memiliki atribut lain seperti biaya transaksi (*gas fee*), tanda tangan digital, dan juga *nonce*. Semua atribut ini akan digabungkan untuk membentuk sebuah transaksi Solana yang kemudian dikirim ke jaringan *blockchain*-nya untuk diproses oleh *validator* Solana. Setelah transaksi diproses dan diverifikasi, hasilnya akan ditampilkan pada daftar histori transaksi yang dapat diakses secara publik.

Sebagai contoh, untuk membuat sebuah transaksi transfer koin Solana (SOL) dari satu *address* ke *address* yang lain maka hal pertama yang perlu dilakukan adalah

menetapkan instruksi. Instruksi terdiri dari data yang berperan sebagai argumen, *id* program dan *account addresses*. Diumpamakan dalam transaksi ini *address* pengirim adalah A5ipty7wL4SKwrviwDBTdgwdTwBLCroxi3duGLA7NhRZ dan *address* penerima adalah p5UAW1ps29KFd9Utj2cySbZ4fcsoMk7xZRy6eVi9vys.

Instruksi transfer koin pada Solana membutuhkan nilai *index* dan jumlah koin yang akan ditransfer. Nilai *index* untuk instruksi transfer pada *library* solana adalah 2. Kemudian umpamakan jumlah koin yang akan ditransfer adalah 0,1 SOL yang jika dikonversikan ke satuan terkecil dari koin Solana (*lamports*) bernilai 100.000.000 *lamports*. Selanjutnya, kedua data tersebut perlu dikonversikan ke dalam bentuk *bytes*. Solana sendiri menggunakan format *little-endian* dalam penyusunan yaitu *byte* terkecil atau paling signifikan akan disimpan pada alamat memori yang lebih tinggi (berurut dari kanan ke kiri) dan *byte* terbesar/tidak signifikan akan disimpan pada alamat memori yang lebih rendah (berurut dari kiri ke kanan). Berikut tahapan untuk melakukan konversi terhadap kedua data tersebut menjadi *bytes*:

- Untuk data pertama yaitu data *index*, nilai data *index* yang masih dalam bentuk *decimal* 2 akan dikonversikan ke bentuk *hex* menjadi 0x2. Karena data *index* harus bertipe *unsigned-integer* 32 bit yang memiliki 4 *byte*, maka perlu ditambahkan tujuh digit nol menjadi 0x00000002 yang jika dituliskan ke dalam bentuk *bytes* menjadi [0,0,0,2]. Kemudian terapkan format *little-endian* dengan mengubah urutan *bytes* tersebut dari kanan ke kiri sehingga nilai *bytes* data *index* sekarang adalah [2,0,0,0].
- Untuk data kedua yaitu jumlah koin yang ditransfer, nilai yang digunakan adalah jumlah koin dalam bentuk *lamports* yaitu 100.000.000. Nilai ini masih berbentuk *decimal* sehingga perlu dikonversikan ke *hex* terlebih dahulu menjadi 0x5f5e100. Karena data jumlah koin harus bertipe *unsigned-integer* 64 bit yang memiliki 8 *byte*, maka perlu ditambahkan sembilan digit nol menjadi 0x0000000005f5e100 yang jika dituliskan dalam bentuk *bytes* menjadi [0, 0, 0, 0, 5, 245, 225, 0]. Kemudian terapkan format *little-endian* dengan mengubah urutan *bytes* tersebut dari kanan ke kiri sehingga nilai *bytes* jumlah koin yang ditransfer sekarang adalah [0, 225, 245, 5, 0, 0, 0, 0].
- Setelah itu, *bytes* kedua data ini digabungkan menjadi satu dengan urutan data *index* di depan yang kemudian diikuti oleh data jumlah koin sehingga nilai *bytes* menjadi [2, 0, 0, 0, 225, 245, 5, 0, 0, 0, 0]

Selanjutnya, *addresses* yang ada pada instruksi yaitu *address* pengirim dan penerima juga perlu dikonversi. Berikut tahapan konversi untuk kedua *address* tersebut:

- Untuk *address* pengirim, nilai *address* yang masih bertipe *base58* akan dikonversikan menjadi *hex* sehingga nilai A5ipty7wL4SKwrviwDBTdgwdTwBLCr oxi3duGLA7NhRZ menjadi 0x86eecc53737e2a0ba2beff4f355e07c0c744fcc63f471 1b0ab42c96f08dad9b8. Kemudian nilai tersebut dikonversi kembali menjadi nilai *bytes* yang berjumlah 32 menjadi [134, 238, 204, 83, 115, 126, 42, 11, 162, 190, 255, 79, 53, 94, 7, 192, 199, 68, 252, 198, 63, 71, 17, 176, 171, 66, 201, 111, 8, 218, 217, 184].
 - Untuk *address* penerima, nilai *address* yang masih bertipe *base58* juga akan dikonversikan menjadi *hex* sehingga nilai p5UAW1ps29KFd9Utj2cySbZ4fcso Mk7XzRy6eVi9vys menjadi 0x0c0f59b390649a8a5effb45214145287131b0806b4 88f19c bd1c8c5c90840766. Kemudian nilai tersebut dikonversi kembali ke nilai *bytes* menjadi [12, 15, 89, 179, 144, 100, 154, 138, 94, 255, 180, 82, 20, 20, 82, 135, 19, 27, 8, 6, 180, 136, 241, 156, 189, 28, 140, 92, 144, 132, 7, 102].

Kemudian nilai-nilai *bytes* ini digabungkan dengan urutannya sebagai berikut:

Jumlah Signature + Jumlah Signed Account (Readonly) + Jumlah Unsigned Account (Readonly) + Jumlah Address + [Address] + Blockhash Terbaru + Jumlah Instruksi + [index dari program id + jumlah index address + [index address] + jumlah byte data + data]

Dengan penjelasannya sebagai berikut:

- Jumlah *signed account* yang *read-only* Karena pada transaksi ini tidak memiliki *signer read-only* maka nilainya adalah 0 dan dalam representasi *bytes* adalah [0].

- Jumlah *byte* data, merupakan jumlah *byte* pada data yang sebelumnya telah dijelaskan tahapannya dan berjumlah 12 sehingga representasi *bytes*-nya adalah [12]
 - Data, merupakan representasi *bytes* dari data untuk instruksi ini yang telah dijelaskan sebelumnya pada tahapan di atas dengan hasilnya berupa [2, 0, 0, 0, 0, 225, 245, 5, 0, 0, 0, 0]

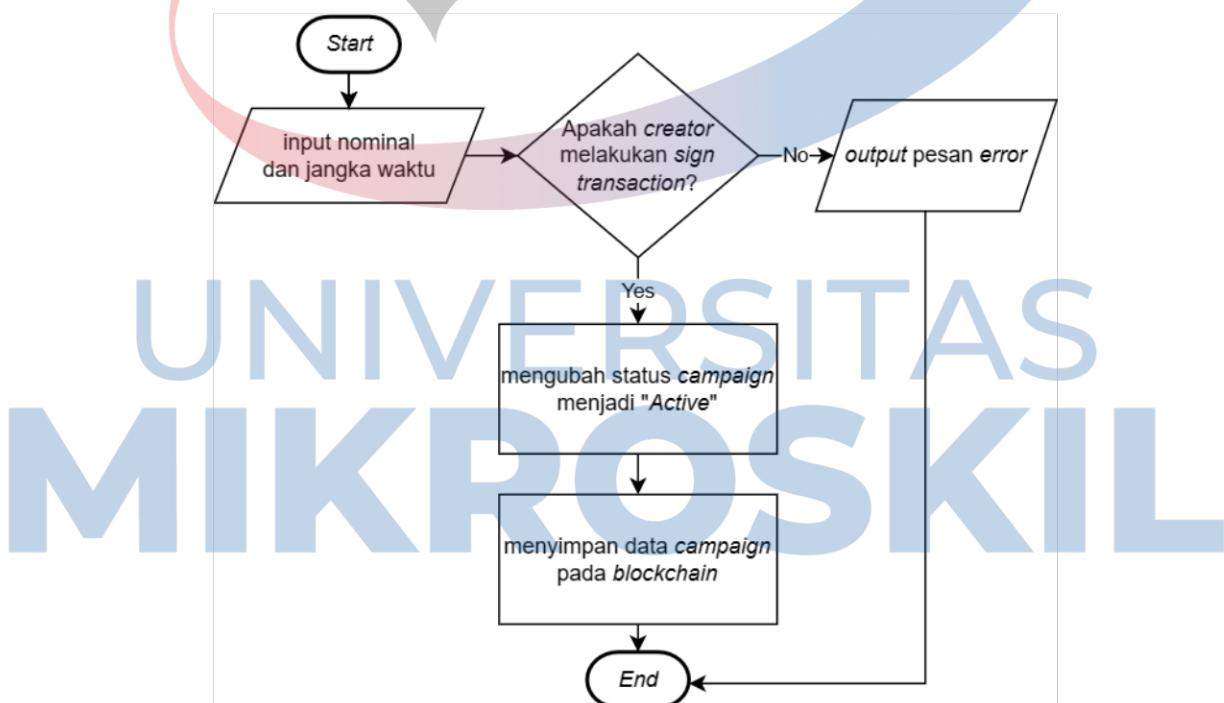
37, 210, 193, 173, 117, 194, 153, 204, 200, 32, 244, 135, 25, 243, 39, 122, 240, 125, 120, 247, 208, 13, 98, 1, 2, 2, 0, 1, 12, 2, 0, 0, 0, 0, 225, 245, 5, 0, 0, 0, 0].

Proses kerja transaksi yang dimaksud meliputi proses pembuatan *campaign*, proses donasi ke *campaign*, proses pengajuan pencairan dana, proses voting proposal, proses penutupan voting proposal, dan proses *refund* dana *campaign* yang gagal. Berikut penjelasan dari tiap proses:

1. Proses pembuatan *campaign*:

- a. Sistem menerima input berupa target nominal pendanaan dan jangka waktu dari *campaign* yang akan dibuat.
- b. Sistem akan melakukan pengecekan apakah *fundraiser* melakukan *sign transaction*. Jika iya, maka lanjut ke tahapan berikutnya. Jika tidak, maka *output* pesan *error* dan proses selesai.
- c. Sistem akan membuat *campaign* dengan status “*Active*” serta menyimpannya pada *blockchain* dan proses selesai.

Ilustrasi *flowchart* proses pembuatan *campaign* dapat dilihat pada Gambar 3.3.



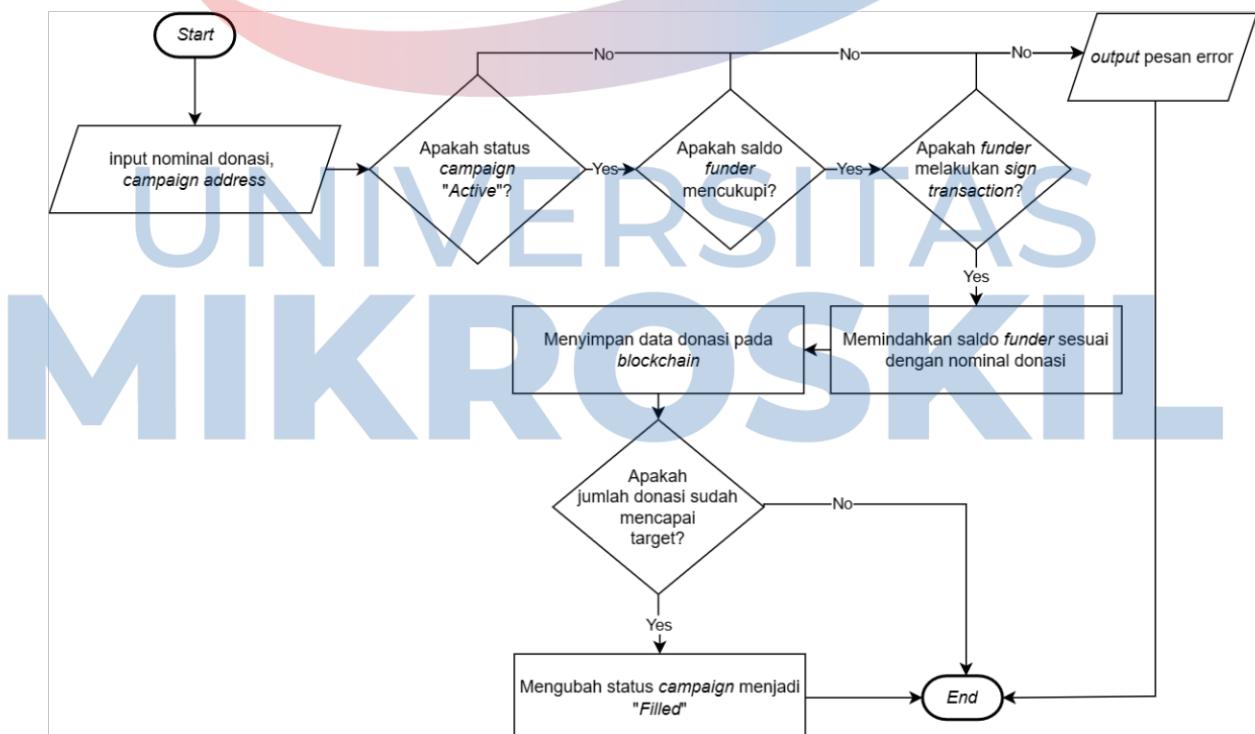
Gambar 3.3 Flowchart Pembuatan Campaign

2. Proses donasi ke *campaign*:

- a. Sistem menerima input berupa nominal dana yang akan didonasikan, dan *campaign address*.

- b. Sistem akan melakukan pengecekan apakah saldo *funder* cukup untuk melakukan donasi dengan nominal yang telah diinput. Jika iya, maka lanjut ke tahapan berikutnya. Jika tidak, maka *output* pesan *error* dan proses selesai.
- c. Sistem akan melakukan pengecekan apakah *funder* melakukan *sign transaction*. Jika iya, maka lanjut ke tahapan berikutnya. Jika tidak, maka *output* pesan *error* dan proses selesai.
- d. Sistem akan melakukan pengecekan apakah status *campaign* masih “*Active*”. Jika iya, maka lanjut ke tahapan berikutnya. Jika tidak, maka *output* pesan *error* dan proses selesai.
- e. Sistem akan memindahkan saldo pada *wallet* milik *funder* ke *campaign* sesuai dengan nominal donasi yang diinput.
- f. Sistem akan menyimpan data donasi pada *blockchain*.
- g. Sistem akan melakukan pengecekan apakah jumlah donasi sudah mencapai target pendanaan. Jika iya, maka lanjut ke tahapan berikutnya. Jika tidak, maka proses selesai.
- h. Sistem akan mengubah status *campaign* menjadi “*Filled*” dan proses selesai.

Ilustrasi *flowchart* proses donasi ke *campaign* dapat dilihat pada Gambar 3.4.

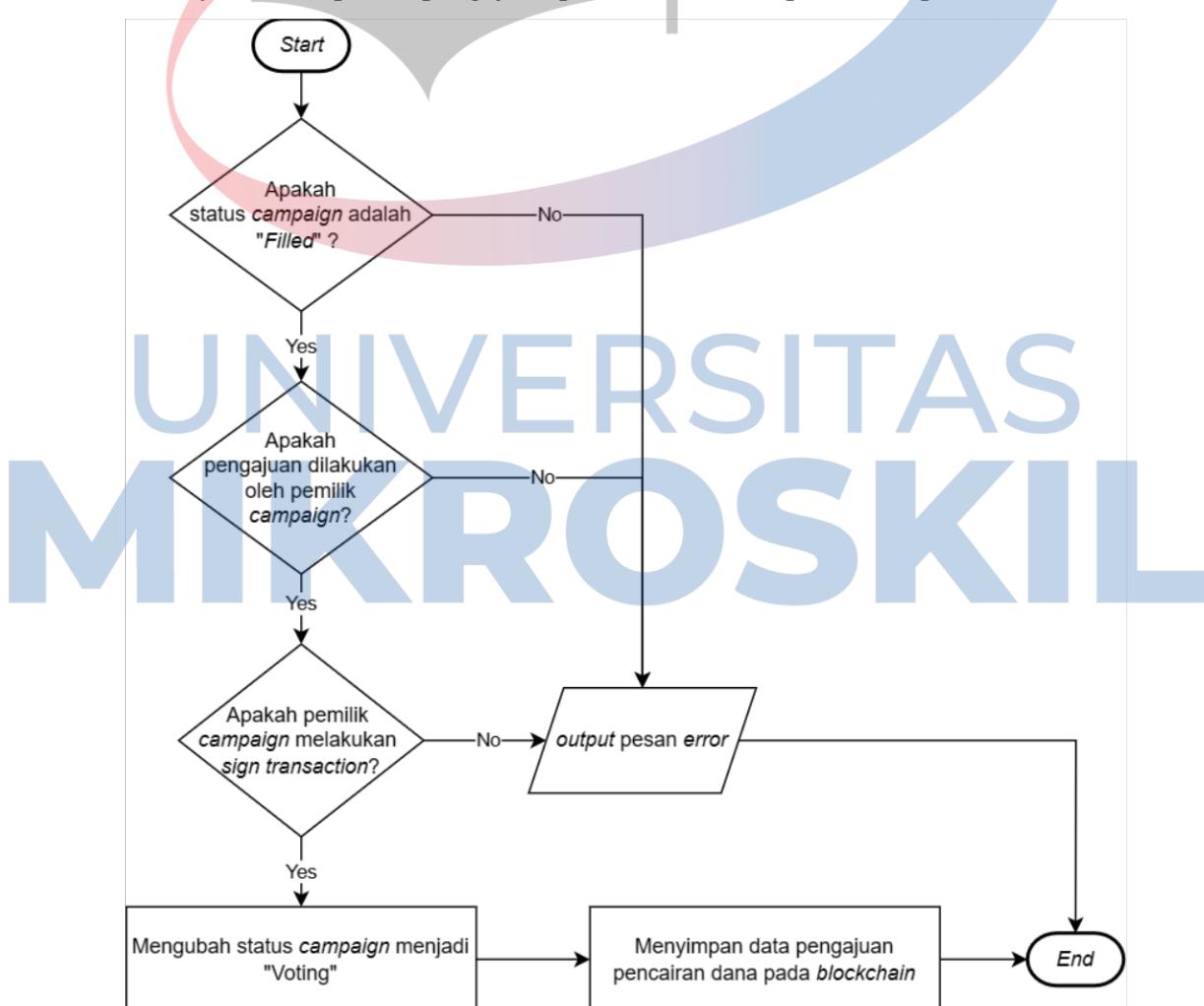


Gambar 3.4 *Flowchart* Donasi Pada *Campaign*

3. Proses pengajuan pencairan dana:

- a. Sistem akan melakukan pengecekan apakah pengajuan dilakukan oleh pemilik *campaign*. Jika iya, maka lanjut ke tahapan berikutnya. Jika tidak, maka *output error* dan proses selesai.
- b. Sistem akan melakukan pengecekan apakah pemilik *campaign* melakukan *sign transaction*. Jika iya, maka lanjut ke tahapan berikutnya. Jika tidak, maka *output error* dan proses selesai.
- c. Sistem akan melakukan pengecekan apakah target pendanaan sudah tercapai dan *campaign* belum pernah mengajukan proposal. Jika iya, maka lanjut ke tahapan berikutnya. Jika tidak, maka *output error* dan proses selesai.
- d. Sistem akan mengubah status *campaign* menjadi “Voting”.
- e. Sistem akan menyimpan data pengajuan pencairan dana pada *blockchain* dan proses selesai.

Ilustrasi *flowchart* proses pengajuan pencairan dana dapat dilihat pada Gambar 3.5.

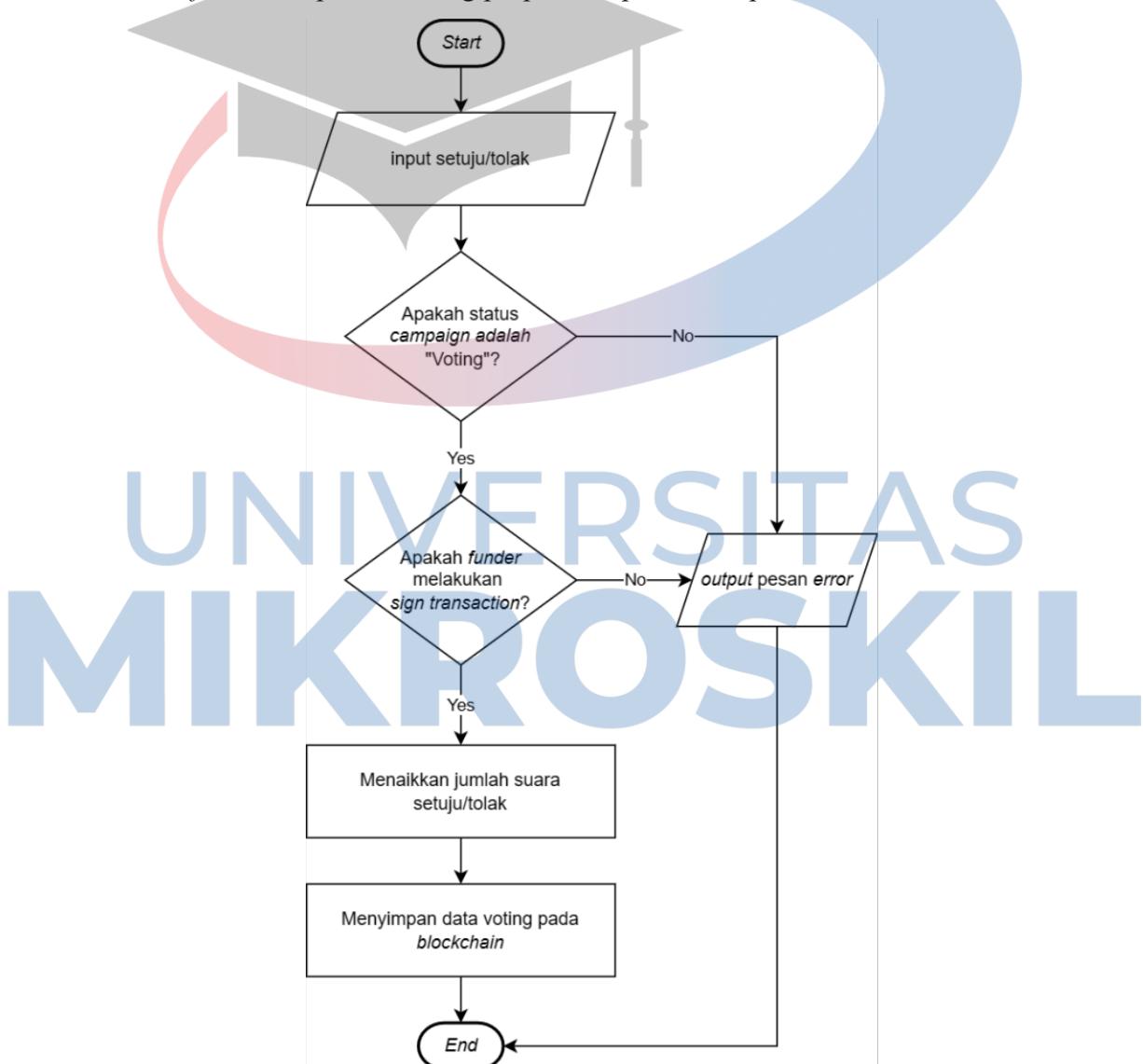


Gambar 3.5 Flowchart Pengajuan Pencairan Dana Campaign

4. Proses voting proposal:

- a. Sistem menerima inputan berupa setuju atau tolak.
- b. Sistem akan melakukan pengecekan apakah *funder* melakukan *sign transaction*. Jika iya, maka lanjut ke tahapan berikutnya. Jika tidak, maka *output* pesan *error* dan proses selesai.
- c. Sistem akan melakukan pengecekan apakah status *campaign* adalah “Voting”. Jika iya, maka lanjut ke tahapan berikutnya. Jika tidak, maka *output* pesan *error* dan proses selesai.
- d. Sistem akan menaikkan jumlah suara setuju atau tolak dalam pengumpulan suara.
- e. Sistem akan menyimpan data voting pada *blockchain* dan proses selesai.

Ilustrasi *flowchart* proses voting proposal dapat dilihat pada Gambar 3.6.

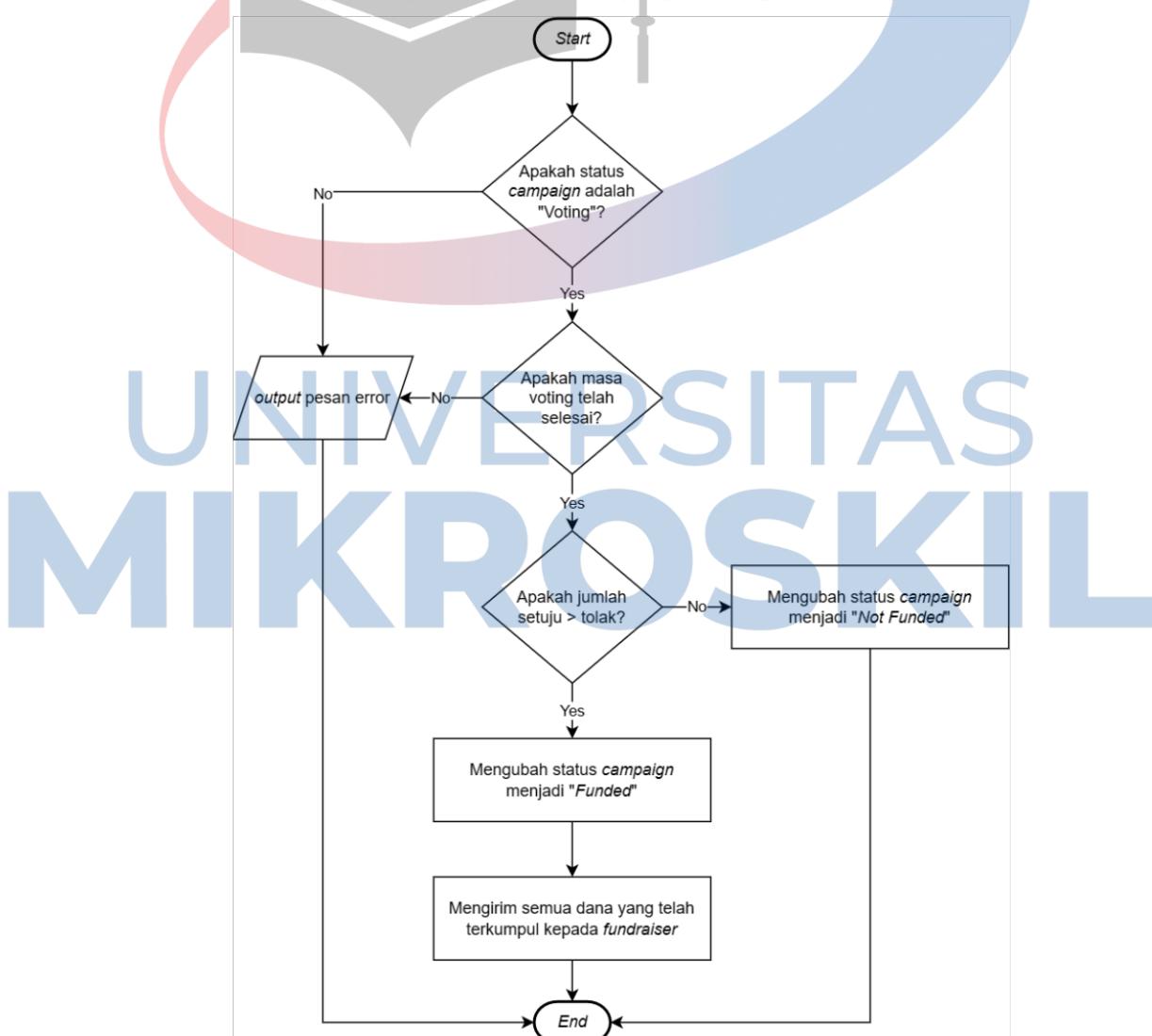


Gambar 3.6 *Flowchart* Voting Proposal

5. Proses penutupan voting proposal:

- a. Sistem akan melakukan pengecekan apakah status campaign adalah “Voting”. Jika iya, maka lanjut ke tahapan berikutnya. Jika tidak, maka *output* pesan *error* dan proses selesai.
- b. Sistem akan melakukan pengecekan apakah masa voting telah selesai. Jika iya, maka lanjut ke tahapan berikutnya. Jika tidak, maka *output* pesan *error* dan proses selesai.
- c. Sistem akan melakukan pengecekan apakah jumlah suara setuju lebih besar dari suara tolak. Jika iya, maka lanjut ke tahapan berikutnya. Jika tidak, maka sistem akan mengubah status *campaign* menjadi “*Not Funded*” dan proses selesai.
- d. Sistem akan mengubah status *campaign* menjadi “*Funded*”.
- e. Sistem akan mengirim dana yang telah dikumpulkan *campaign* ke *fundraiser* dan proses selesai.

Ilustrasi *flowchart* proses penutupan voting proposal dapat dilihat pada Gambar 3.7.

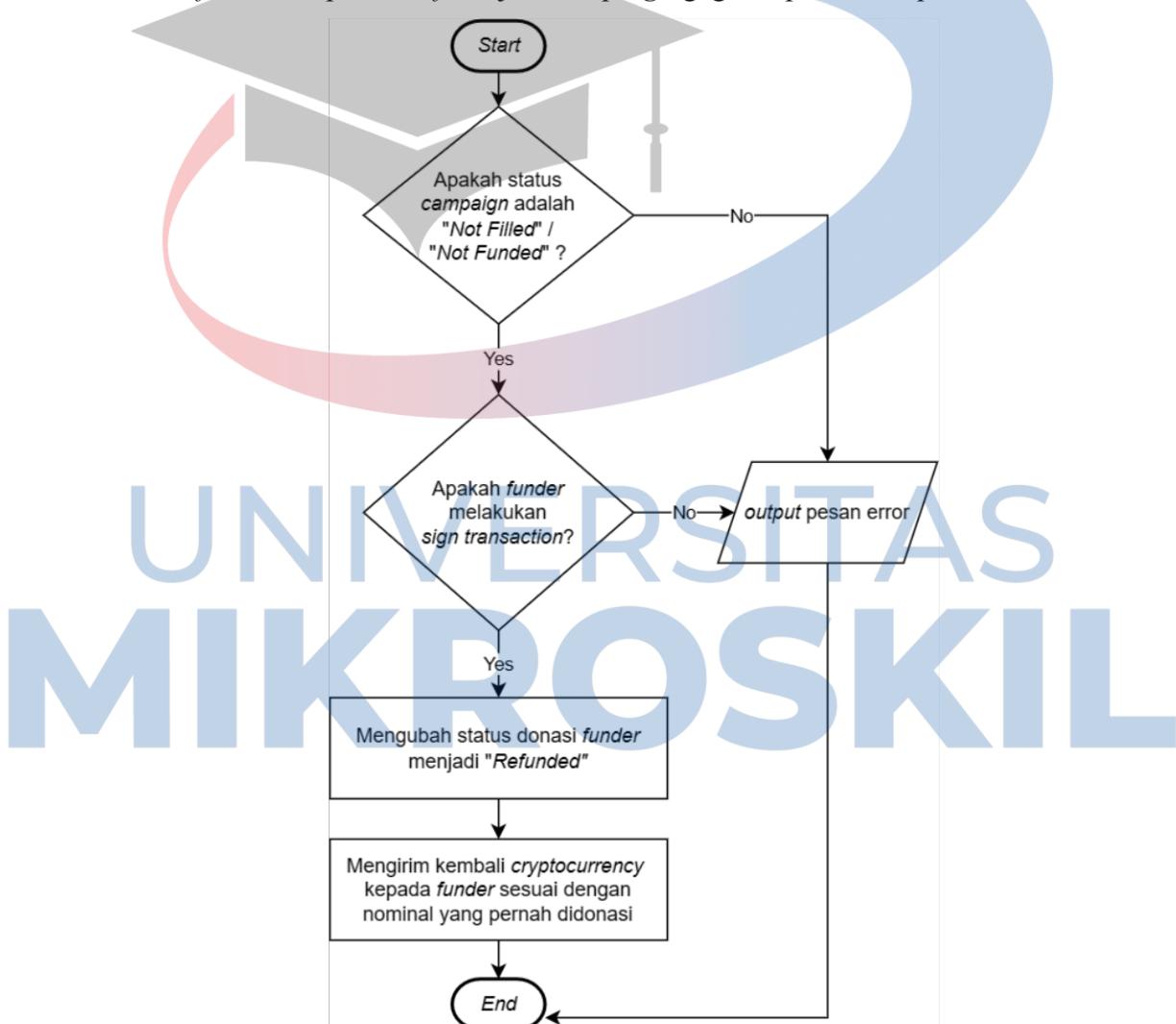


Gambar 3.7 *Flowchart* Penutupan Voting Proposal

6. Proses *refund* jika *campaign* gagal:

- a. Sistem akan melakukan pengecekan apakah status *campaign* adalah “*Not Filled*” atau “*Not Funded*”. Jika iya, maka lanjut ke tahapan berikutnya. Jika tidak, maka *output* pesan *error* dan proses selesai.
- b. Sistem akan melakukan pengecekan apakah *funder* melakukan *sign transaction*. Jika iya, maka lanjut ke tahapan berikutnya. Jika tidak, maka *output* pesan *error* dan proses selesai.
- c. Sistem akan mengubah status donasi *funder* menjadi “*Refunded*”.
- d. Sistem akan mengirimkan kembali *cryptocurrency* yang telah didonasikan kepada *funder* sesuai dengan nominal yang didonasikan dan proses selesai.

Ilustrasi *flowchart* proses *refund* jika *campaign* gagal dapat dilihat pada Gambar 3.8.



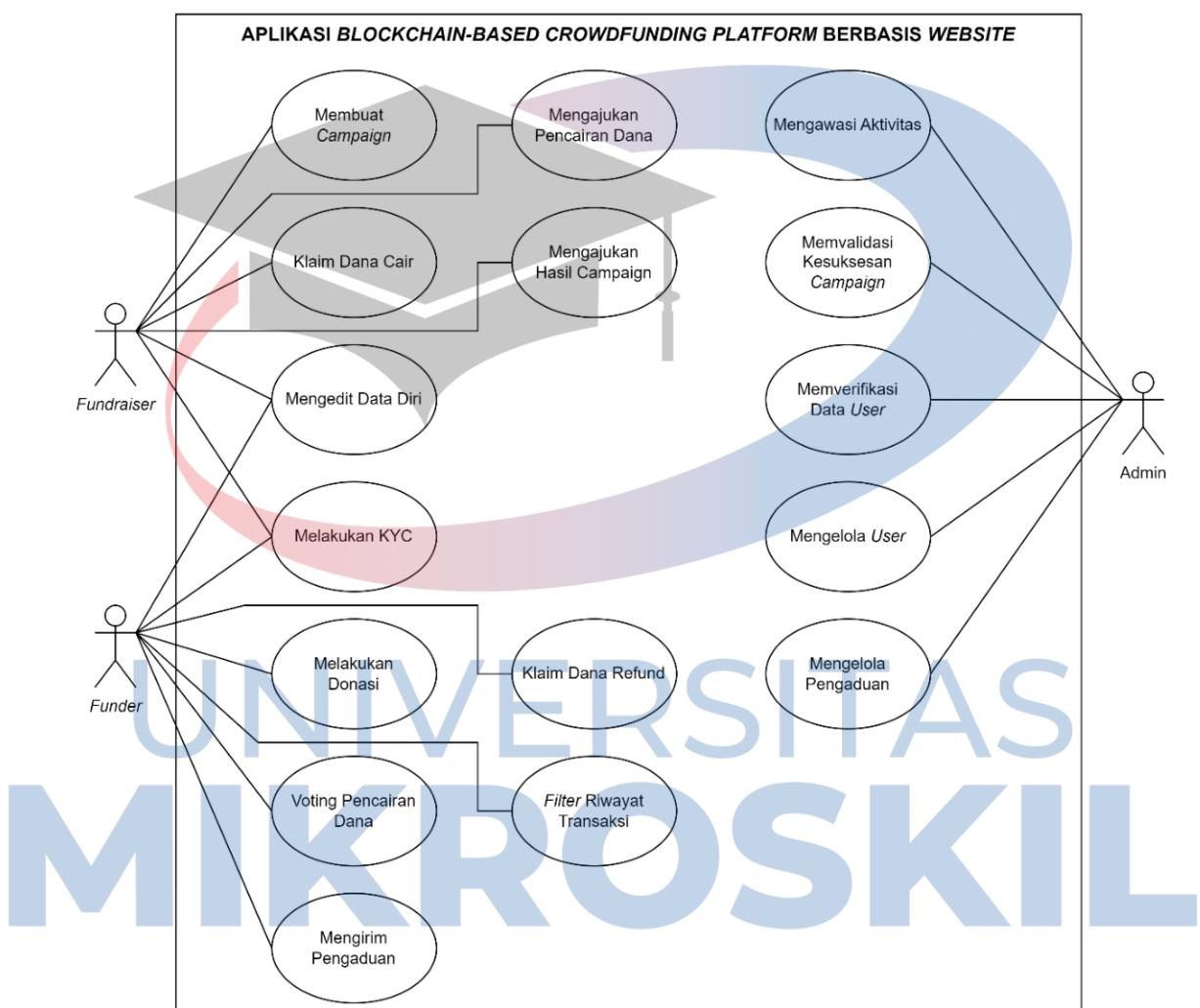
Gambar 3.8 *Flowchart Refund*

3.1.3 Analisis Kebutuhan

Analisis kebutuhan terdiri dari analisis kebutuhan fungsional yang akan dimodelkan dengan diagram *use case*, dan kebutuhan non fungsional akan dimodelkan dengan pemodelan PIECES (*Performance, Information, Economy, Control, Efficiency, Service*).

1. Analisis Kebutuhan Fungsional

Berikut diagram *use case* yang dapat menunjukkan interaksi antara sistem dengan *user* secara keseluruhan. *Use case* dapat dilihat pada Gambar 3.9.



Gambar 3.9 Diagram *Use Case* Aplikasi

Narasi dari deskripsi aktor dapat dilihat pada Tabel 3.1.

Tabel 3.1 Deskripsi Aktor

No.	Nama Aktor	Deskripsi
1.	Admin	Aktor yang menggunakan aplikasi untuk mengawasi aktivitas pada aplikasi, memverifikasi <i>user</i> , mengelola <i>user</i> , mengelola verifikasi keberhasilan <i>campaign</i> dan mengelola pengaduan.
2.	<i>Fundraiser (User)</i>	Aktor yang menggunakan aplikasi untuk membuat, mengelola dan melihat riwayat <i>campaign</i> , serta mengedit data diri dan mengajukan verifikasi data diri.
3.	<i>Funder (User)</i>	Aktor yang menggunakan aplikasi untuk melakukan donasi dan melihat riwayat donasi, serta mengedit data diri, mengajukan verifikasi data diri dan mengirim pengaduan.

Keseluruhan diagram *Use Case* dari aplikasi *blockchain-based crowdfunding platform* yang akan dirancang dapat dilihat pada Tabel 3.2.

Tabel 3.2 Deskripsi *Use Case* Aplikasi *Blockchain-Based Crowdfunding Platform*

No.	Use Case	Deskripsi
1.	<i>Login</i> dan <i>Logout</i>	<i>Use case</i> ini menjelaskan proses yang dilakukan Admin, <i>Fundraiser</i> , dan <i>Funder</i> untuk <i>login</i> dan <i>logout</i> ke aplikasi.
2.	Mengawasi aktivitas	<i>Use case</i> ini menjelaskan proses yang dilakukan Admin untuk mengawasi aktivitas yang terjadi pada aplikasi.
3.	Memverifikasi kesuksesan <i>campaign</i>	<i>Use case</i> ini menjelaskan proses yang dilakukan Admin untuk memverifikasi kesuksesan <i>campaign</i> .
4.	Memverifikasi data <i>user</i>	<i>Use case</i> ini menjelaskan proses yang dilakukan Admin untuk memverifikasi data <i>user</i> .
5.	Mengelola <i>user</i>	<i>Use case</i> ini menjelaskan proses yang dilakukan Admin untuk mengelola <i>user</i> .
6.	Mengelola pengaduan	<i>Use case</i> ini menjelaskan proses yang dilakukan Admin untuk mengelola pengaduan dari <i>user</i> .
7.	Membuat <i>campaign</i>	<i>Use case</i> ini menjelaskan proses yang dilakukan <i>Fundraiser</i> untuk membuat <i>campaign</i> .

8.	Mengajukan pencairan dana	<i>Use case</i> ini menjelaskan proses yang dilakukan <i>Fundraiser</i> untuk mengajukan pencairan dana jika sudah memenuhi syarat pengajuan.
9.	Klaim dana cair	<i>Use case</i> ini menjelaskan proses yang dilakukan <i>Fundraiser</i> untuk klaim dana yang berhasil dicairkan.
10.	Mengajukan hasil <i>campaign</i>	<i>Use case</i> ini menjelaskan proses yang dilakukan <i>Fundraiser</i> untuk mengajukan hasil penggerjaan <i>campaign</i> untuk verifikasi status kesuksesan <i>campaign</i> .
11.	Mengedit data diri	<i>Use case</i> ini menjelaskan proses yang dilakukan <i>Fundraiser</i> dan <i>Funder</i> untuk mengedit data diri.
12.	Melakukan KYC	<i>Use case</i> ini menjelaskan proses yang dilakukan <i>Fundraiser</i> dan <i>Funder</i> untuk mengajukan verifikasi data diri.
13.	Melakukan donasi	<i>Use case</i> ini menjelaskan proses yang dilakukan <i>Funder</i> untuk melakukan donasi.
14.	Klaim dana <i>refund</i>	<i>Use case</i> ini menjelaskan proses yang dilakukan <i>Funder</i> untuk klaim dana <i>refund</i> .
15.	Voting pencairan dana	<i>Use case</i> ini menjelaskan proses yang dilakukan <i>Funder</i> untuk melakukan voting proposal pencairan dana.
16.	<i>Filter</i> riwayat transaksi	<i>Use case</i> ini menjelaskan proses yang dilakukan oleh <i>Fundraiser</i> dan <i>Funder</i> untuk mem-filter riwayat transaksi.
17.	Mengirim pengaduan	<i>Use case</i> ini menjelaskan proses yang dilakukan oleh <i>Funder</i> untuk mengirim pengaduan.

Skenario dari masing-masing *use case* yang ada pada Tabel 3.2. dapat dilihat dari Tabel 3.3 hingga Tabel 3.19.

Tabel 3.3 Skenario *Use Case* : Login dan Logout

Nama Use Case	<i>Login</i> dan <i>Logout</i>	
Aktor	<i>Admin</i> , <i>Fundraiser</i> dan <i>Funder</i>	
Deskripsi	<i>Use case</i> ini menjelaskan proses yang dilakukan <i>Admin</i> , <i>Fundraiser</i> , dan <i>Funder</i> untuk <i>login</i> dan <i>logout</i> ke aplikasi.	
	Aksi Aktor	Respon Sistem

Normal Flow Event	<p>1. Menekan tombol <i>Select Wallet</i> atau Hubungkan <i>Wallet</i> yang ada pada halaman.</p> <p>3. Mengikuti arahan <i>browser extension wallet</i>.</p> <p>6. Menekan tombol <i>Disconnect</i></p>	<p>2. Menampilkan <i>browser extension wallet</i> Phantom.</p> <p>4. Melakukan autentikasi.</p> <p>5. Aksi sistem:</p> <ul style="list-style-type: none"> - Jika <i>User</i>, sistem menampilkan halaman Beranda <i>User</i>. - Jika Admin, sistem menampilkan halaman Beranda Admin. <p>7. <i>Wallet</i> berhasil <i>disconnect</i> dan kembali ke halaman Beranda</p>
Alternative Flow Event		<p>8. Jika ada kesalahan, tampilkan pesan kesalahan.</p>

Tabel 3.4 Skenario Use Case : Mengawasi Aktivitas

Nama Use Case	Mengawasi aktivitas	
Aktor	Admin	
Deskripsi	<i>Use case</i> ini menjelaskan proses yang dilakukan Admin untuk mengawasi aktivitas yang terjadi pada aplikasi.	
	Aksi Aktor	Respon Sistem
Normal Flow Event	<p>1. Masuk ke halaman <i>Dashboard</i> melalui menu <i>Dashboard</i>.</p>	<p>2. Menampilkan halaman <i>Dashboard</i>.</p>
Alternative Flow Event		<p>3. Jika ada kesalahan, tampilkan pesan kesalahan.</p>

Tabel 3.5 Skenario *Use Case* : Memverifikasi Kesuksesan *Campaign*

Nama Use Case	Memverifikasi kesuksesan <i>campaign</i>	
Aktor	Admin	
Deskripsi	<i>Use case</i> ini menjelaskan proses yang dilakukan Admin untuk memverifikasi kesuksesan <i>campaign</i> .	
	Aksi Aktor	Respon Sistem
<i>Normal Flow Event</i>	<p>1. Masuk ke halaman <i>Campaign</i> pada menu <i>Dashboard</i>.</p> <p>3. Memilih aksi yang dilakukan:</p> <ul style="list-style-type: none"> - Untuk melihat berkas, tekan tombol Lihat - Untuk setuju, tekan tombol Setuju. - Untuk menolak, tekan tombol Tolak. 	<p>2. Menampilkan halaman <i>Campaign</i>.</p> <p>4. Aksi sistem:</p> <ul style="list-style-type: none"> - Jika lihat berkas, tampilan diarahkan ke halaman berkas pada browser tab baru. - Jika setuju, status <i>campaign</i> menjadi "Sukses" - Jika ditolak, status <i>campaign</i> menjadi "Gagal"
<i>Alternative Flow Event</i>		<p>5. Jika ada kesalahan, tampilkan pesan kesalahan.</p>

Tabel 3.6 Skenario *Use Case* : Memverifikasi Data *User*

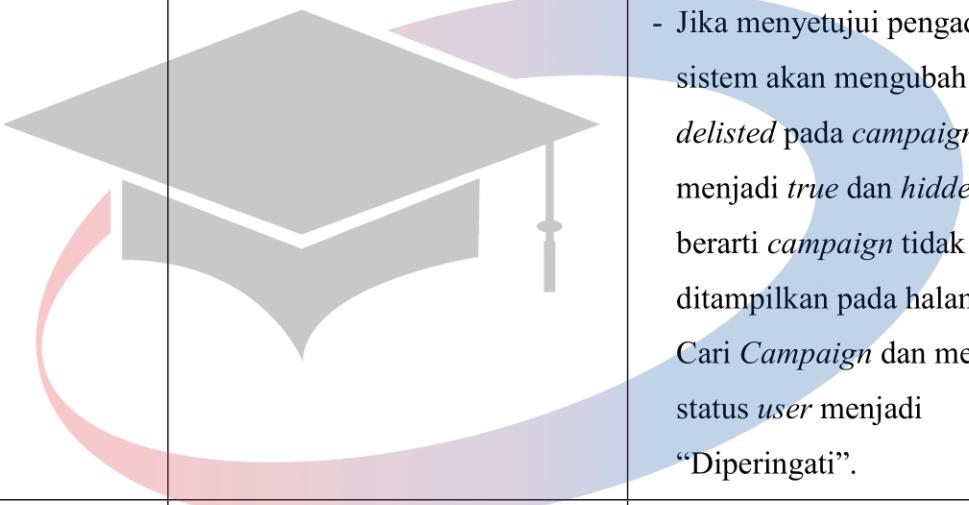
Nama Use Case	Memverifikasi data <i>user</i>	
Aktor	Admin	
Deskripsi	<i>Use case</i> ini menjelaskan proses yang dilakukan Admin untuk memverifikasi data <i>user</i> .	
	Aksi Aktor	Respon Sistem
<i>Normal Flow</i> <i>Event</i>	<p>1. Masuk ke halaman Verifikasi <i>User</i> pada menu <i>Dashboard</i>.</p> <p>3. Memilih aksi yang dilakukan:</p> <ul style="list-style-type: none"> - Untuk mengunduh berkas, tekan tombol Unduh. - Untuk menyetujui verifikasi <i>user</i>, tekan tombol Setuju. - Untuk menolak verifikasi <i>user</i>, tekan tombol Tolak. 	<p>2. Menampilkan halaman Verifikasi <i>User</i>.</p> <p>4. Aksi sistem:</p> <ul style="list-style-type: none"> - Jika mengunduh berkas, sistem mengunduh berkas. - Jika menyetujui verifikasi <i>user</i>, sistem mengubah status verifikasi <i>user</i> menjadi “Diterima”. - Jika menolak verifikasi <i>user</i>, sistem mengubah status verifikasi <i>user</i> menjadi “Ditolak”.
<i>Alternative Flow Event</i>		5. Jika ada kesalahan, tampilkan pesan kesalahan.

Tabel 3.7 Skenario Use Case : Mengelola User

Nama Use Case	Mengelola user	
Aktor	Admin	
Deskripsi	<i>Use case ini menjelaskan proses yang dilakukan Admin untuk mengelola user.</i>	
	Aksi Aktor	Respon Sistem
Normal Flow Event	<p>1. Masuk ke halaman Manajemen <i>User</i> pada menu Admin.</p> <p>3. Memilih aksi yang dilakukan:</p> <ul style="list-style-type: none"> - Untuk melihat detail profil <i>user</i>, tekan tombol Lihat. - Untuk menghapus status verifikasi <i>user</i>, tekan tombol Hapus. 	<p>2. Menampilkan halaman Manajemen <i>User</i>.</p> <p>4. Aksi sistem:</p> <ul style="list-style-type: none"> - Jika melihat detail profil <i>user</i>, sistem menampilkan halaman Detail Profil <i>User</i>. - Jika menghapus status verifikasi <i>user</i>, sistem mengubah status verifikasi <i>user</i> menjadi “Verifikasi Dicabut”
Alternative Flow Event		5. Jika ada kesalahan, tampilkan pesan kesalahan.

Tabel 3.8 Skenario Use Case : Mengelola Pengaduan User

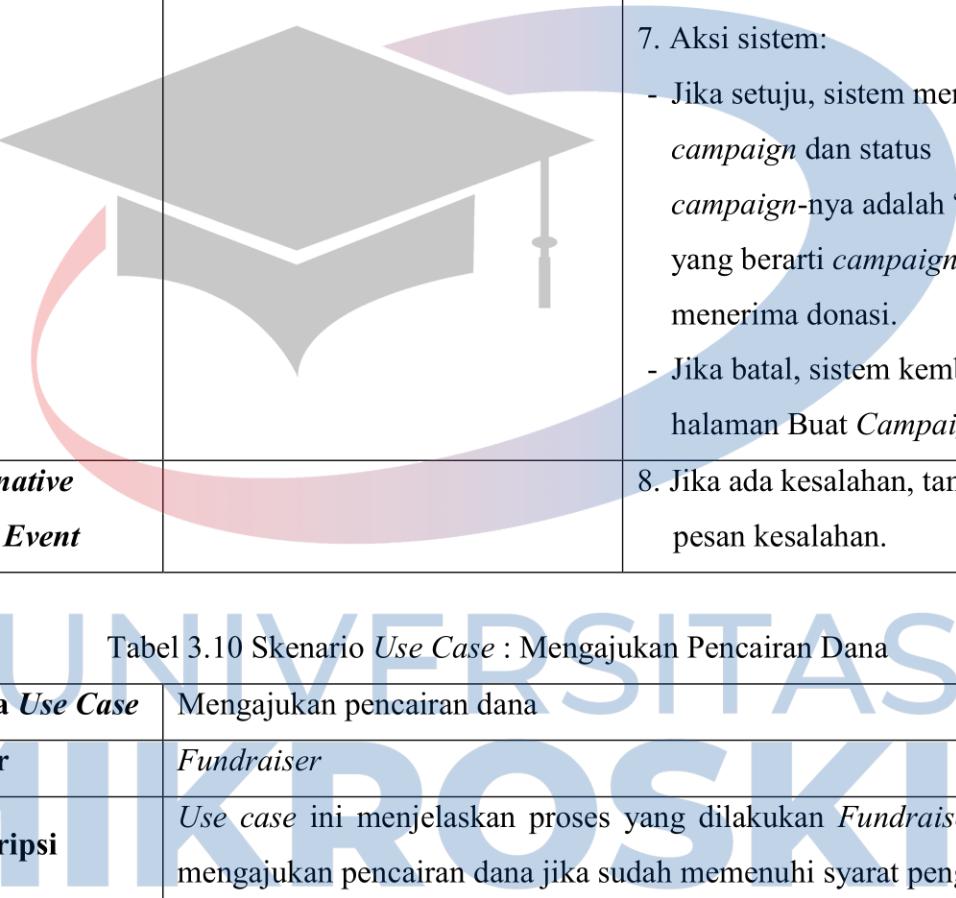
Nama Use Case	Mengelola pengaduan user	
Aktor	Admin	
Deskripsi	<i>Use case ini menjelaskan proses yang dilakukan Admin untuk mengelola pengaduan dari user.</i>	
	Aksi Aktor	Respon Sistem
<i>Normal Flow</i>	<p>1. Masuk ke halaman Pengaduan dari menu <i>Dashboard</i>.</p> <p>3. Memilih aksi yang dilakukan:</p> <ul style="list-style-type: none"> - Untuk melihat pengaduan, tekan tombol Lihat Detail Pengaduan. - Untuk menyetujui pengaduan, tekan tombol Terima Pengaduan. <p>5. Pada Halaman Pengaduan, pilih aksi yang dilakukan:</p>	<p>2. Menampilkan halaman Pengaduan.</p> <p>4. Aksi sistem:</p> <ul style="list-style-type: none"> - Jika melihat detail pengaduan, sistem menampilkan halaman Detail Pengaduan. - Jika menyetujui pengaduan, sistem akan mengubah status <i>delisted</i> pada <i>campaign</i> menjadi <i>true</i> yang berarti <i>campaign</i> tidak akan ditampilkan pada halaman dan mengubah status <i>user</i> menjadi “Diperingati”.
<i>Event</i>		



	<ul style="list-style-type: none"> - Untuk melihat detail <i>campaign</i>, tekan tombol Lihat Detail <i>Campaign</i>. - Untuk menyetujui pengaduan, tekan tombol Terima Pengaduan. 	
<i>Alternative Flow Event</i>		<p>6. Aksi sistem:</p> <ul style="list-style-type: none"> - Jika melihat detail <i>campaign</i>, sistem menampilkan halaman Detail <i>Campaign</i> pada <i>browser tab</i> baru. - Jika menyetujui pengaduan, sistem akan mengubah status <i>delisted</i> pada <i>campaign</i> menjadi <i>true</i> dan <i>hidden</i> yang berarti <i>campaign</i> tidak akan ditampilkan pada halaman Cari <i>Campaign</i> dan mengubah status <i>user</i> menjadi “Diperingati”. <p>7. Jika ada kesalahan, tampilkan pesan kesalahan.</p>

Tabel 3.9 Skenario Use Case : Membuat *Campaign*

Nama Use Case	Membuat <i>campaign</i>	
Aktor	<i>Fundraiser</i>	
Deskripsi	Use case ini menjelaskan proses yang dilakukan <i>Fundraiser</i> untuk membuat <i>campaign</i> .	
	Aksi Aktor	Respon Sistem
<i>Normal Flow Event</i>	1. Masuk ke halaman Buat <i>Campaign</i> . 3. Isi seluruh data yang diperlukan.	2. Menampilkan halaman Buat <i>Campaign</i> .



	<p>4. Tekan tombol Buat <i>Campaign</i>.</p> <p>6. Memilih aksi yang dilakukan:</p> <ul style="list-style-type: none"> - Untuk menyetujui transaksi, tekan tombol <i>Approve</i>. - Untuk batal, tekan tombol <i>Cancel</i>. 	<p>5. Menampilkan <i>browser extension wallet Phantom</i>.</p> <p>7. Aksi sistem:</p> <ul style="list-style-type: none"> - Jika setuju, sistem membuat <i>campaign</i> dan status <i>campaign</i>-nya adalah “<i>Active</i>” yang berarti <i>campaign</i> sedang menerima donasi. - Jika batal, sistem kembali ke halaman Buat <i>Campaign</i>.
<i>Alternative Flow Event</i>		8. Jika ada kesalahan, tampilkan pesan kesalahan.

Tabel 3.10 Skenario Use Case : Mengajukan Pencairan Dana

Nama Use Case	Mengajukan pencairan dana	
Aktor	<i>Fundraiser</i>	
Deskripsi	<i>Use case</i> ini menjelaskan proses yang dilakukan <i>Fundraiser</i> untuk mengajukan pencairan dana jika sudah memenuhi syarat pengajuan.	
	Aksi Aktor	Respon Sistem
<i>Normal Flow Event</i>	<p>1. Masuk ke halaman Detail <i>Campaign</i> yang ingin dicairkan dananya.</p>	<p>2. Sistem melakukan pengecekan apakah status <i>campaign</i> adalah “<i>Filled</i>” yang berarti target pengumpulan dana telah</p>

	<p>4. Menekan tombol Ajukan Pencairan Dana.</p> <p>6. <i>Upload</i> proposal pengajuan pencairan dana.</p> <p>7. Memilih aksi yang dilakukan:</p> <ul style="list-style-type: none"> - Untuk mengajukan pencairan, tekan tombol Ajukan. - Untuk batal, tekan tombol Batal. <p>9. Memilih aksi yang dilakukan:</p> <ul style="list-style-type: none"> - Untuk menyetujui transaksi, tekan tombol <i>Approve</i>. - Untuk batal, tekan tombol <i>Cancel</i>. 	<p>dicapai dan apakah <i>fundraiser</i> yang melakukan aksi adalah <i>owner</i> dari <i>campaign</i>.</p> <p>3. Menampilkan halaman Detail <i>Campaign</i>.</p> <p>5. Menampilkan <i>modal</i> Pengajuan Pencairan.</p> <p>8. Aksi sistem:</p> <ul style="list-style-type: none"> - Jika mengajukan pencairan, sistem menampilkan <i>browser extension wallet</i> Phantom. - Jika batal, sistem kembali ke halaman Detail <i>Campaign</i>. <p>10. Aksi sistem:</p> <ul style="list-style-type: none"> - Jika setuju, sistem mengubah status <i>campaign</i> dari “Active” menjadi “Voting” dan melampirkan file proposal
--	--	---

		pada halaman Detail <i>Campaign user</i> . - Jika batal, sistem kembali ke halaman Detail <i>Campaign</i> .
Alternative Flow Event		11. Jika ada kesalahan, tampilkan pesan kesalahan.

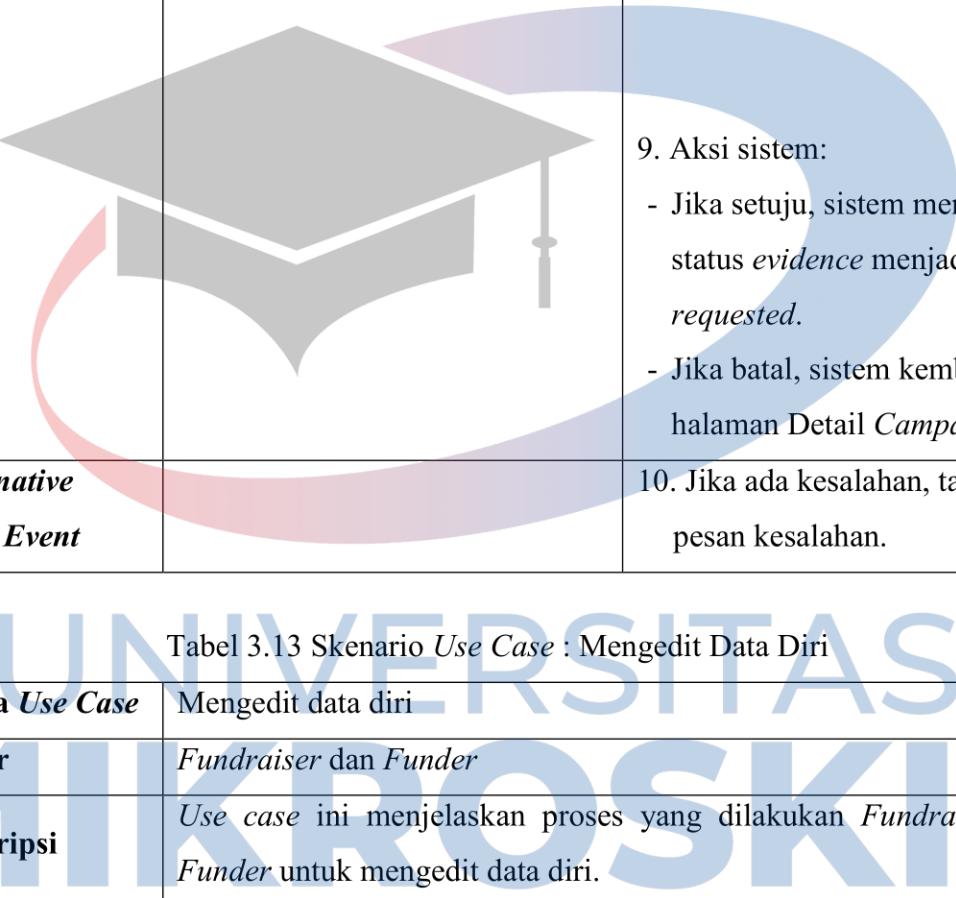
Tabel 3.11 Skenario Use Case : Klaim Dana Cair

Nama Use Case	Klaim dana cair	
Aktor	<i>Fundraiser</i>	
Deskripsi	Use case ini menjelaskan proses yang dilakukan <i>Fundraiser</i> untuk klaim dana yang berhasil dicairkan.	
	Aksi Aktor	Respon Sistem
Normal Flow Event	1. Masuk ke halaman Detail <i>Campaign</i> yang ingin diklaim dananya. 3. Untuk klaim dana, tekan tombol “Klaim Dana”.	2. Menampilkan halaman Detail <i>Campaign</i> . 4. Sistem melakukan pengecekan apakah status <i>campaign</i> adalah “Fund Claimable” yang berarti dana dapat dicairkan. 5. Menampilkan browser extension <i>wallet Phantom</i> .
	6. Memilih aksi yang dilakukan: - Untuk menyetujui transaksi, tekan tombol <i>Approve</i> . - Untuk batal, tekan tombol <i>Cancel</i> .	7. Aksi sistem:

		<ul style="list-style-type: none"> - Jika setuju, sistem memindahkan dana ke <i>fundraiser wallet</i>. - Jika batal, sistem kembali ke halaman Detail <i>Campaign</i>.
Alternative Flow Event		8. Jika ada kesalahan, tampilkan pesan kesalahan.

Tabel 3.12 Skenario Use Case : Mengajukan Hasil *Campaign*

Nama Use Case	Mengajukan hasil <i>campaign</i>	
Aktor	<i>Fundraiser</i>	
Deskripsi	<p><i>Use case</i> ini menjelaskan proses yang dilakukan <i>Fundraiser</i> untuk mengajukan hasil penggeraan <i>campaign</i> untuk verifikasi status kesuksesan <i>campaign</i>.</p>	
Normal Flow Event	Aksi Aktor	Respon Sistem
	<ol style="list-style-type: none"> 1. Masuk ke halaman Detail <i>Campaign</i> yang ingin divalidasi kesuksesannya. 3. Menekan tombol Ajukan Bukti Keberhasilan <i>Campaign</i>. 5. <i>Upload</i> berkas keberhasilan proyek. 6. Memilih aksi yang dilakukan: <ul style="list-style-type: none"> - Untuk mengirimkan berkas , tekan tombol Kirim. - Untuk batal, tekan tombol Batal. 	<ol style="list-style-type: none"> 2. Menampilkan halaman Detail <i>Campaign</i>. 4. Menampilkan modal Pengiriman Bukti. 7. Aksi sistem:



	<p>8. Memilih aksi yang dilakukan:</p> <ul style="list-style-type: none"> - Untuk menyetujui transaksi, tekan tombol <i>Approve</i>. - Untuk batal, tekan tombol <i>Cancel</i>. 	<ul style="list-style-type: none"> - Untuk mengirim berkas, sistem menampilkan <i>browser extension wallet Phantom</i>. - Untuk batal, sistem kembali ke halaman Detail <i>Campaign</i>.
<i>Alternative Flow Event</i>		<p>9. Aksi sistem:</p> <ul style="list-style-type: none"> - Jika setuju, sistem mengubah status <i>evidence</i> menjadi <i>requested</i>. - Jika batal, sistem kembali ke halaman Detail <i>Campaign</i>. <p>10. Jika ada kesalahan, tampilkan pesan kesalahan.</p>

Tabel 3.13 Skenario *Use Case* : Mengedit Data Diri

Nama <i>Use Case</i>	Mengedit data diri	
Aktor	<i>Fundraiser</i> dan <i>Funder</i>	
Deskripsi	<i>Use case</i> ini menjelaskan proses yang dilakukan <i>Fundraiser</i> dan <i>Funder</i> untuk mengedit data diri.	
	Aksi Aktor	Respon Sistem
<i>Normal Flow Event</i>	<p>1. Masuk ke halaman Pengaturan Akun pada menu Profil.</p> <p>3. Isi atau edit data diri.</p> <p>4. Menekan tombol Simpan Data.</p>	<p>2. Menampilkan halaman Pengaturan Akun.</p>

		5. Sistem menyimpan data diri terbaru dari <i>user</i> .
Alternative Flow Event		6. Jika ada kesalahan, tampilkan pesan kesalahan.

Tabel 3.14 Skenario *Use Case* : Melakukan KYC

Nama Use Case	Melakukan KYC	
Aktor	<i>Fundraiser</i> dan <i>Funder</i>	
Deskripsi	<i>Use case</i> ini menjelaskan proses yang dilakukan <i>Fundraiser</i> dan <i>Funder</i> untuk mengajukan verifikasi data diri.	
	Aksi Aktor	Respon Sistem
Normal Flow Event	1. Masuk ke halaman Pengaturan Akun pada menu Profil. 3. Mengisi seluruh data yang dibutuhkan untuk verifikasi 4. Menekan tombol Verifikasi.	2. Menampilkan halaman Pengaturan Akun. 5. Sistem menyimpan data verifikasi dan mengubah status verifikasi menjadi “ <i>pending</i> ” yang berarti sedang menunggu verifikasi oleh Admin.
Alternative Flow Event		6. Jika ada kesalahan, tampilkan pesan kesalahan.

Tabel 3.15 Skenario *Use Case* : Melakukan Donasi

Nama Use Case	Melakukan donasi	
Aktor	<i>Funder</i>	
Deskripsi	<i>Use case</i> ini menjelaskan proses yang dilakukan <i>Funder</i> untuk melakukan donasi.	
	Aksi Aktor	Respon Sistem

<p>Normal Flow Event</p> 	<ol style="list-style-type: none"> 1. Masuk ke halaman Detail <i>Campaign</i> yang ingin didonasikan. 3. Mengisi jumlah USDC yang akan didonasikan. 4. Menekan tombol Donasi Sekarang. 7. Memilih aksi yang dilakukan: <ul style="list-style-type: none"> - Untuk menyetujui transaksi, tekan tombol <i>Approve</i>. - Untuk batal, tekan tombol <i>Cancel</i>. 	<ol style="list-style-type: none"> 2. Menampilkan halaman Detail <i>Campaign</i>. 5. Sistem melakukan pengecekan apakah saldo <i>funder</i> mencukupi dan status <i>campaign</i> adalah “Active” yang berarti sedang menerima donasi. 6. Menampilkan browser extension <i>wallet Phantom</i>. 8. Aksi sistem: <ul style="list-style-type: none"> - Jika setuju, sistem memindahkan saldo dari <i>wallet funder</i> ke <i>smart contract</i>. - Jika batal, sistem kembali ke halaman Detail <i>Campaign</i>. 9. Jika ada kesalahan, tampilkan pesan kesalahan.
<p>Alternative Flow Event</p>		

Tabel 3.16 Skenario Use Case : Klaim Dana Refund

Nama Use Case	Klaim dana <i>refund</i>	
Aktor	<i>Funder</i>	
Deskripsi	<i>Use case</i> ini menjelaskan proses yang dilakukan <i>Funder</i> untuk klaim dana <i>refund</i> .	
	Aksi Aktor	Respon Sistem
<i>Normal Flow</i>	<p>1. Masuk ke halaman Detail <i>Campaign</i>.</p> <p>3. Untuk klaim dana, tekan tombol “Klaim Refund”</p>	<p>2. Menampilkan halaman Detail <i>Campaign</i>.</p> <p>4. Sistem melakukan pengecekan apakah status <i>campaign</i> adalah “<i>Not Filled</i>” yang berarti target pengumpulan dana <i>campaign</i> gagal tercapai atau “<i>Not Funded</i>” yang berarti hasil voting menyatakan mayoritas <i>Funder</i> tidak setuju terhadap pencairan dana.</p>
<i>Event</i>	<p>6. Memilih aksi yang dilakukan:</p> <ul style="list-style-type: none"> - Untuk menyetujui transaksi, tekan tombol <i>Approve</i>. - Untuk batal, tekan tombol <i>Cancel</i>. 	<p>5. Menampilkan browser extension wallet Phantom.</p> <p>7. Aksi sistem:</p> <ul style="list-style-type: none"> - Jika setuju, sistem memindahkan saldo dari <i>smart contract</i> ke <i>wallet funder</i>.

		<ul style="list-style-type: none"> - Jika batal, sistem kembali ke halaman Detail <i>Campaign</i>.
Alternative Flow Event		8. Jika ada kesalahan, tampilkan pesan kesalahan.

Tabel 3.17 Skenario Use Case : Voting Pencairan Dana

Nama Use Case	Voting pencairan dana	
Aktor	<i>Funder</i>	
Deskripsi	<i>Use case</i> ini menjelaskan proses yang dilakukan <i>Funder</i> untuk melakukan voting proposal pencairan dana.	
Aksi Aktor		Respon Sistem
Normal Flow Event	<p>1. Masuk ke halaman Detail <i>Campaign</i> yang sudah pernah didonasikan, dan status <i>campaign</i>-nya adalah “Voting”.</p> <p>3. Untuk membaca berkas proposal, tekan <i>file</i> proposal yang dilampirkan.</p> <p>4. Memilih aksi yang dilakukan:</p> <ul style="list-style-type: none"> - Jika setuju, tekan tombol Setuju. - Jika tolak, tekan tombol Tolak. <p>6. Memilih aksi yang dilakukan:</p> <ul style="list-style-type: none"> - Untuk menyetujui transaksi, tekan tombol <i>Approve</i>. - Untuk batal, tekan tombol <i>Cancel</i>. 	<p>2. Menampilkan halaman Detail <i>Campaign</i>.</p> <p>5. Menampilkan <i>browser extension wallet Phantom</i>.</p>

		<p>7. Aksi sistem:</p> <ul style="list-style-type: none"> - Jika setuju, sistem mencatat isi <i>vote</i>. - Jika batal, sistem kembali ke halaman Detail <i>Campaign</i>.
Alternative Flow Event		8. Jika ada kesalahan, tampilkan pesan kesalahan.

Tabel 3.18 Skenario Use Case : Filter Riwayat Transaksi

Nama Use Case	<i>Filter</i> riwayat transaksi	
Aktor	<i>Fundraiser</i> dan <i>Funder</i>	
Deskripsi	<i>Use case</i> ini menjelaskan proses yang dilakukan oleh <i>Fundraiser</i> dan <i>Funder</i> untuk mem- <i>filter</i> riwayat transaksi.	
Normal Flow Event	<p>Aksi Aktor</p> <ol style="list-style-type: none"> 1. Masuk ke halaman Riwayat Transaksi. 3. Mengatur <i>filter</i> agar riwayat transaksi yang muncul hanya yang diinginkan. 	<p>Respon Sistem</p> <ol style="list-style-type: none"> 2. Menampilkan halaman Riwayat Transaksi. 4. Menampilkan riwayat transaksi sesuai <i>filter</i>. 5. Jika ada kesalahan, tampilkan pesan kesalahan.
Alternative Flow Event		

Tabel 3.19 Skenario Use Case : Mengirim Pengaduan

Nama Use Case	Mengirim pengaduan	
Aktor	<i>Funder</i>	
Deskripsi	<i>Use case</i> ini menjelaskan proses yang dilakukan <i>Funder</i> untuk mengirim pengaduan.	
	Aksi Aktor	Respon Sistem

Normal Flow Event	<p>1. Masuk ke halaman Detail <i>Campaign</i> yang ingin diadukan.</p> <p>3. Menekan tombol Pengaduan.</p> <p>5. Mengisi pengaduan.</p> <p>6. Memilih aksi yang dilakukan:</p> <ul style="list-style-type: none"> - Untuk mengirim pengaduan, tekan tombol Kirim. - Untuk batal, tekan tombol Batal. 	<p>2. Menampilkan halaman Detail <i>Campaign</i>.</p> <p>4. Menampilkan <i>modal</i> Buat Pengaduan.</p> <p>7. Aksi sistem:</p> <ul style="list-style-type: none"> - Jika mengirim pengaduan, sistem akan menyimpan pengaduan. - Jika batal, sistem kembali ke halaman Detail <i>Campaign</i>.
Alternative Flow Event		8. Jika ada kesalahan, tampilkan pesan kesalahan.

2. Analisis Kebutuhan Non Fungsional

Analisis kebutuhan *non fungsional* dilakukan dengan menggunakan metode PIECES

yang dapat dilihat pada Tabel 3.20.

Tabel 3.20 Analisis PIECES

Parameter	Penjelasan
<i>Performance</i>	Waktu yang dibutuhkan sistem untuk mengeksekusi transaksi atau proses yang terjadi
<i>Information</i>	Informasi data transaksi tercatat lengkap dan bersifat transparan karena dapat diakses oleh publik

<i>Economy</i>	Sistem berbasis web sehingga dapat diakses kapanpun dan dimanapun selama memiliki koneksi internet
<i>Control</i>	<ol style="list-style-type: none"> 1. Sistem mampu menangani kesalahan input dari <i>user</i> dengan menampilkan pesan <i>error</i> 2. Sistem dapat membedakan aktor yang melakukan <i>login</i> berdasarkan <i>wallet address</i>
<i>Efficiency</i>	Proses yang terjadi pada sistem seluruhnya bersifat otomatis kecuali bagian <i>admin</i> sehingga lebih menghemat waktu
<i>Service</i>	Sistem dirancang dengan sifat <i>user-friendly</i> sehingga <i>user</i> mudah menggunakannya

3.2 Perancangan

Pada perancangan akan dilampirkan gambaran aplikasi yang akan dirancang yang terdiri dari perancangan tampilan dan perancangan basis data.

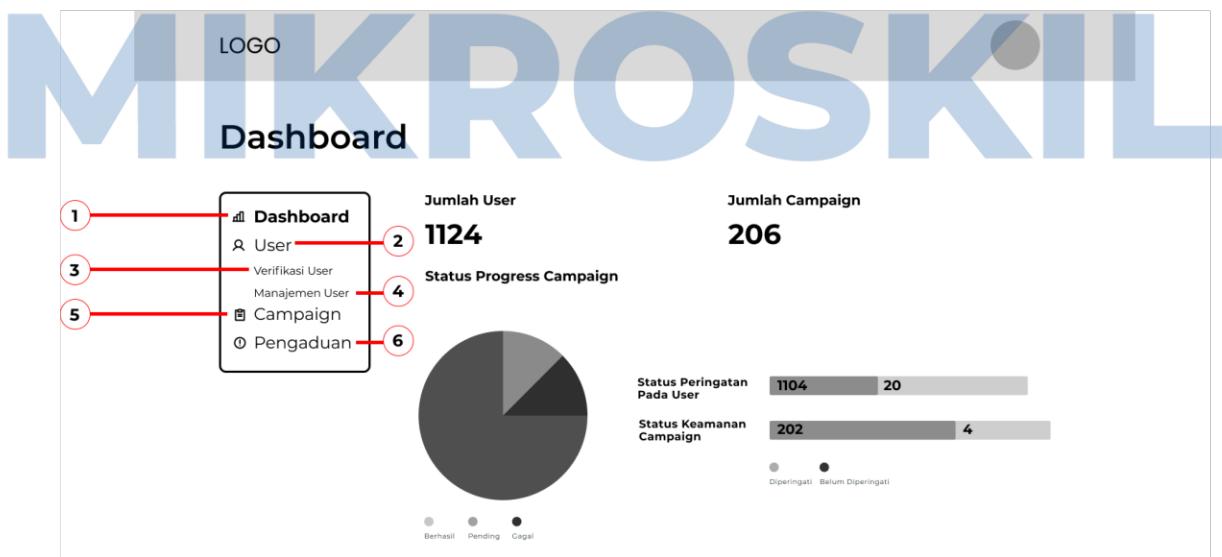
3.2.1 Perancangan Tampilan

Perancangan tampilan digunakan untuk menampilkan gambaran tampilan dari aplikasi yang akan dirancang. Berikut tampilan aplikasi:

1. Admin

a. Halaman *Dashboard*

Gambar 3.10 merupakan rancangan tampilan halaman yang digunakan admin untuk mengawasi aktivitas pada aplikasi.



Gambar 3.10 Tampilan *Dashboard*

Keterangan:

- (1) *Text link*, untuk mengarahkan admin ke halaman *Dashboard*.
- (2) *Dropdown menu*, untuk menampilkan opsi *text link* Verifikasi *User* dan Manajemen *User*.
- (3) *Text link*, untuk mengarahkan admin ke halaman Verifikasi *User*.
- (4) *Text link*, untuk mengarahkan admin ke halaman Manajemen *User*.
- (5) *Text link*, untuk mengarahkan admin ke halaman Verifikasi Kesuksesan *Campaign*.
- (6) *Text link*, untuk mengarahkan admin ke halaman Pengaduan.

b. Halaman

Verifikasi

User

Gambar 3.11 merupakan rancangan tampilan halaman yang digunakan admin untuk memverifikasi *user* yang mengajukan verifikasi data diri.

No	Nama	Wallet Address	Berkas	Aksi
1	User 1	Wallet Address 1		
2	User 2	Wallet Address 2		
3	User 3	Wallet Address 3		
4	User 4	Wallet Address 4		
5	User 5	Wallet Address 5		

Gambar 3.11 Tampilan Verifikasi *User*

Keterangan:

- (1) *Button*, untuk mengunduh berkas foto untuk keperluan verifikasi.
- (2) *Button*, untuk menyetujui verifikasi *user*.
- (3) *Button*, untuk menolak verifikasi *user*.
- (4) *Pagination*, untuk navigasi tabel *user* yang mengajukan verifikasi data diri.

c. Halaman

Manajemen

User

Gambar 3.12 merupakan rancangan tampilan halaman yang digunakan admin untuk manajemen *user*.

No	Nama	Wallet Address	Aksi
1	User 1	Wallet Address 1	<input type="checkbox"/> <input checked="" type="checkbox"/> Cabut Verifikasi
2	User 2	Wallet Address 2	<input type="checkbox"/> <input checked="" type="checkbox"/>
3	User 3	Wallet Address 3	<input type="checkbox"/> <input checked="" type="checkbox"/>
4	User 4	Wallet Address 4	<input type="checkbox"/> <input checked="" type="checkbox"/>
5	User 5	Wallet Address 5	<input type="checkbox"/> <input checked="" type="checkbox"/>

Hal 1 dari 10 < >

Gambar 3.12 Tampilan Manajemen User

Keterangan :

- (1) *Button*, untuk mengarahkan admin ke halaman Detail Profil *User*.
- (2) *Button*, untuk mencabut status verifikasi *user* dan hanya tampil jika *user* sudah terverifikasi.

d. Halaman Detail Profil *User*

Gambar 3.13 merupakan rancangan tampilan halaman yang digunakan admin untuk melihat detail profil *user*.

Status Verifikasi
Nama User
Status Peringatan

E-mail
Address

Total Donasi	Total Pendapatan
15 USDC	2.000 USDC
Campaign Yang Didonasi	Campaign Yang Dibuat
2 Campaign	2 Campaign

1

Gambar 3.13 Tampilan Detail Profil User

Keterangan :

- (1) *Button*, untuk menangguhkan status verifikasi *user* jika *user* sudah terverifikasi.

e. Halaman Validasi *Campaign*

Gambar 3.14 merupakan rancangan tampilan halaman yang digunakan admin untuk memvalidasi keberhasilan proyek dari sebuah *campaign*.

The screenshot shows a user interface titled 'Verifikasi Kesuksesan Campaign'. At the top left is a 'LOGO' placeholder. On the left, a sidebar menu includes 'Dashboard', 'User', 'Verifikasi User', 'Manajemen User', 'Campaign' (which is selected), and 'Pengaduan'. The main content area displays a table with columns: No, Campaign Address, Wallet Address Pemilik, Berkas, and Aksi. The table contains five rows of data:

No	Campaign Address	Wallet Address Pemilik	Berkas	Aksi
1	Address Campaign 1	Wallet Address 1	<input type="button" value="↓"/>	<input checked="" type="button"/> Projek Berhasil
2	Address Campaign 2	Wallet Address 2	<input type="button" value="↓"/>	<input checked="" type="button"/> X
3	Address Campaign 3	Wallet Address 3	<input type="button" value="↓"/>	<input checked="" type="button"/> X
4	Address Campaign 4	Wallet Address 4	<input type="button" value="↓"/>	<input checked="" type="button"/> X
5	Address Campaign 5	Wallet Address 5	<input type="button" value="↓"/>	<input checked="" type="button"/> X

At the bottom right of the table, it says 'Hal 1 dari 10 < >'.

Gambar 3.14 Tampilan Validasi *Campaign*

Keterangan :

- (1) *Button*, untuk mengambil berkas bukti bahwa projek berhasil dilaksanakan.
- (2) *Button*, untuk mengubah status *campaign* menjadi *Success*.
- (3) *Button*, untuk mengubah status *campaign* menjadi *Failed*.

f. Halaman Pengaduan

Gambar 3.15 merupakan rancangan tampilan halaman yang digunakan admin untuk melihat daftar tiap *campaign* yang memiliki pengaduan.

The screenshot shows a user interface with a large watermark 'UNIVERSITAS MIKROSKIL' at the top. At the top left is a 'LOGO' placeholder. On the left, a sidebar menu includes 'Dashboard', 'User', 'Verifikasi User', 'Manajemen User', 'Campaign', and 'Pengaduan' (which is selected). The main content area displays a table with columns: No, Campaign, Wallet Address Pemilik, Jumlah Laporan, and Aksi. The table contains five rows of data:

No	Campaign	Wallet Address Pemilik	Jumlah Laporan	Aksi
1	Nama Campaign 1	Wallet Address 1	100	<input type="button"/> <input type="button"/> Penyelesaian Laporan
2	Nama Campaign 2	Wallet Address 2	100	<input type="button"/> <input type="button"/> 1
3	Nama Campaign 3	Wallet Address 3	100	<input type="button"/> <input type="button"/> 2
4	Nama Campaign 4	Wallet Address 4	100	<input type="button"/> <input type="button"/>
5	Nama Campaign 5	Wallet Address 5	100	<input type="button"/> <input type="button"/>

At the bottom right of the table, it says 'Hal 1 dari 10 < >'.

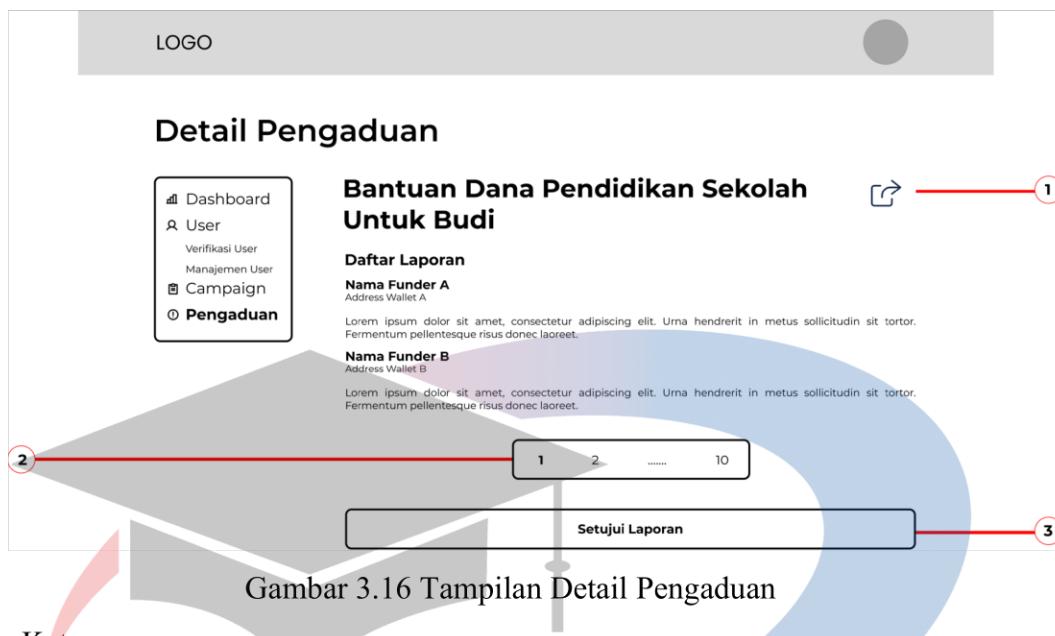
Gambar 3.15 Tampilan Pengaduan

Keterangan :

- (1) *Button*, untuk mengarahkan admin ke halaman Detail Pengaduan.
- (2) *Button*, untuk menyetujui pengaduan.

g. Halaman Detail Pengaduan

Gambar 3.16 merupakan rancangan tampilan halaman yang digunakan admin untuk melihat daftar pengaduan dari *campaign*.



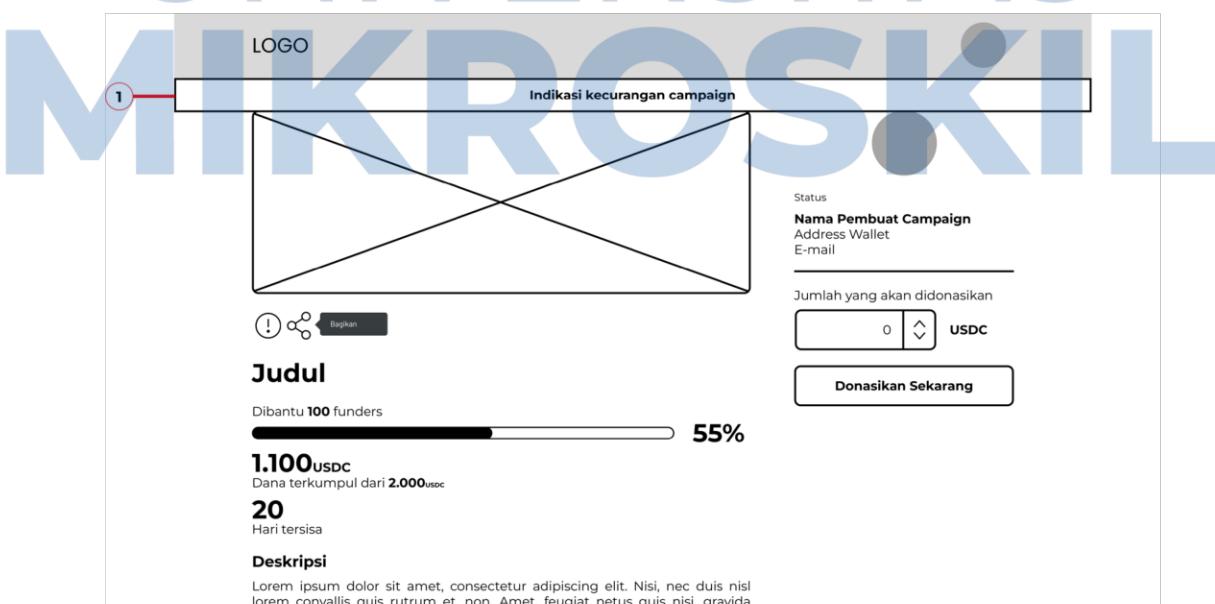
Gambar 3.16 Tampilan Detail Pengaduan

Keterangan :

- (1) *Button*, untuk mengarahkan admin ke halaman Detail *Campaign*.
- (2) *Pagination*, untuk navigasi daftar pengaduan yang ada.
- (3) *Button*, untuk menyetujui pengaduan.

h. Halaman Detail Campaign Yang Dilaporkan

Gambar 3.17 merupakan rancangan tampilan halaman yang digunakan admin untuk melihat detail *campaign* yang memiliki pengaduan.



Gambar 3.17 Tampilan Detail Campaign Yang Dilaporkan

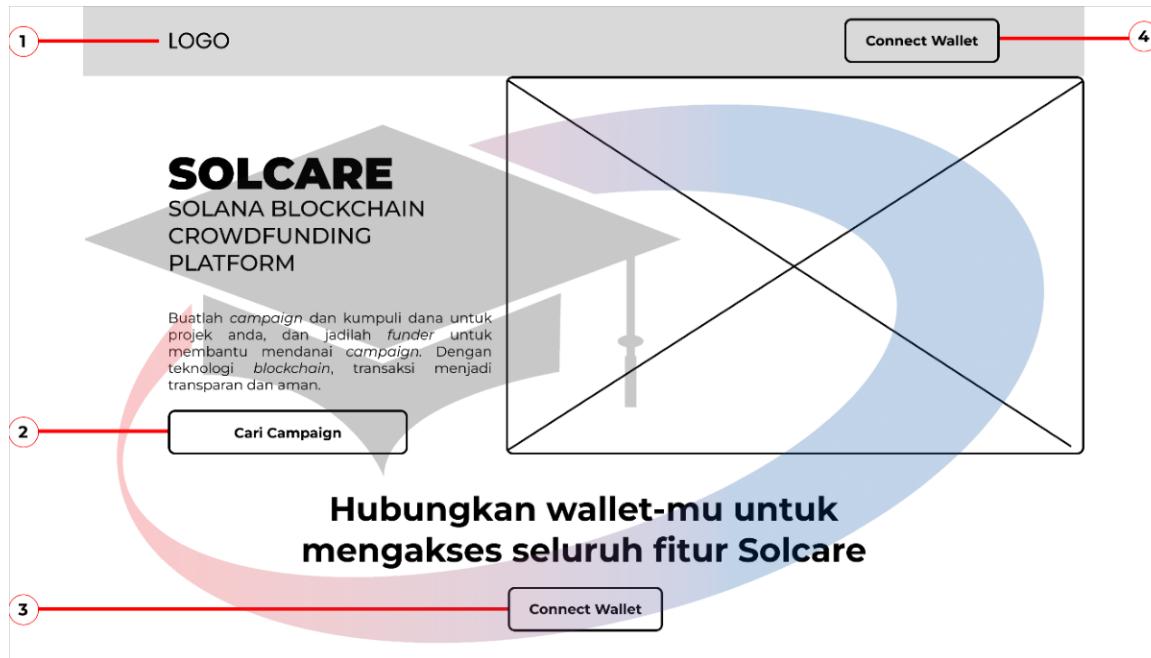
Keterangan :

- (1) *Pagination*, untuk navigasi menuju ke halaman daftar pengaduan.

2. User

a. Halaman Beranda sebelum Login

Gambar 3.18 merupakan rancangan tampilan halaman utama yang dilihat *user* ketika mengakses aplikasi.



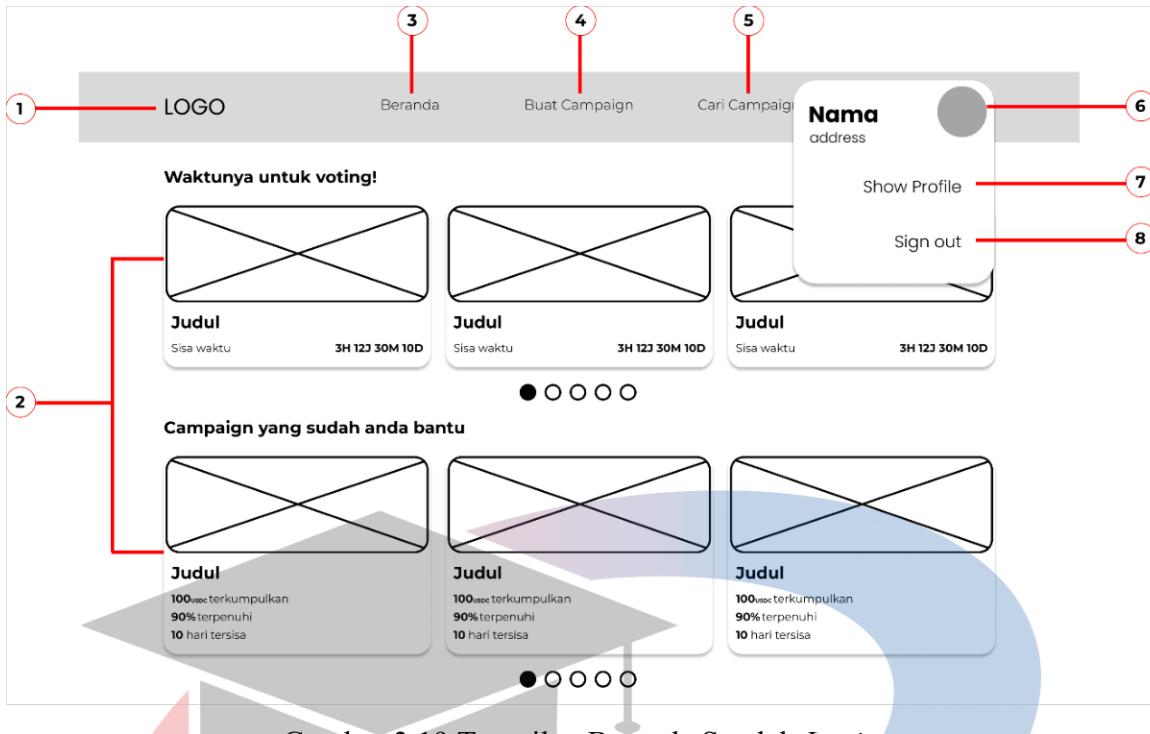
Gambar 3.18 Tampilan Beranda Sebelum Login

Keterangan :

- (1) Button, untuk mengarahkan *user* ke halaman Beranda.
(2) Button, untuk mengarahkan *user* ke halaman Cari Campaign.
(3) Button, untuk mengarahkan *user* agar menghubungkan wallet terlebih dahulu.
(4) Button, untuk mengarahkan *user* agar menghubungkan wallet terlebih dahulu.

b. Halaman Beranda setelah Login

Gambar 3.19 merupakan rancangan tampilan halaman utama yang tampil jika *user* telah melakukan login dengan connect wallet. Halaman berisi kumpulan campaign yang dikelompokkan per bagian untuk kemudahan aksesnya.



Gambar 3.19 Tampilan Beranda Setelah Login

Keterangan :

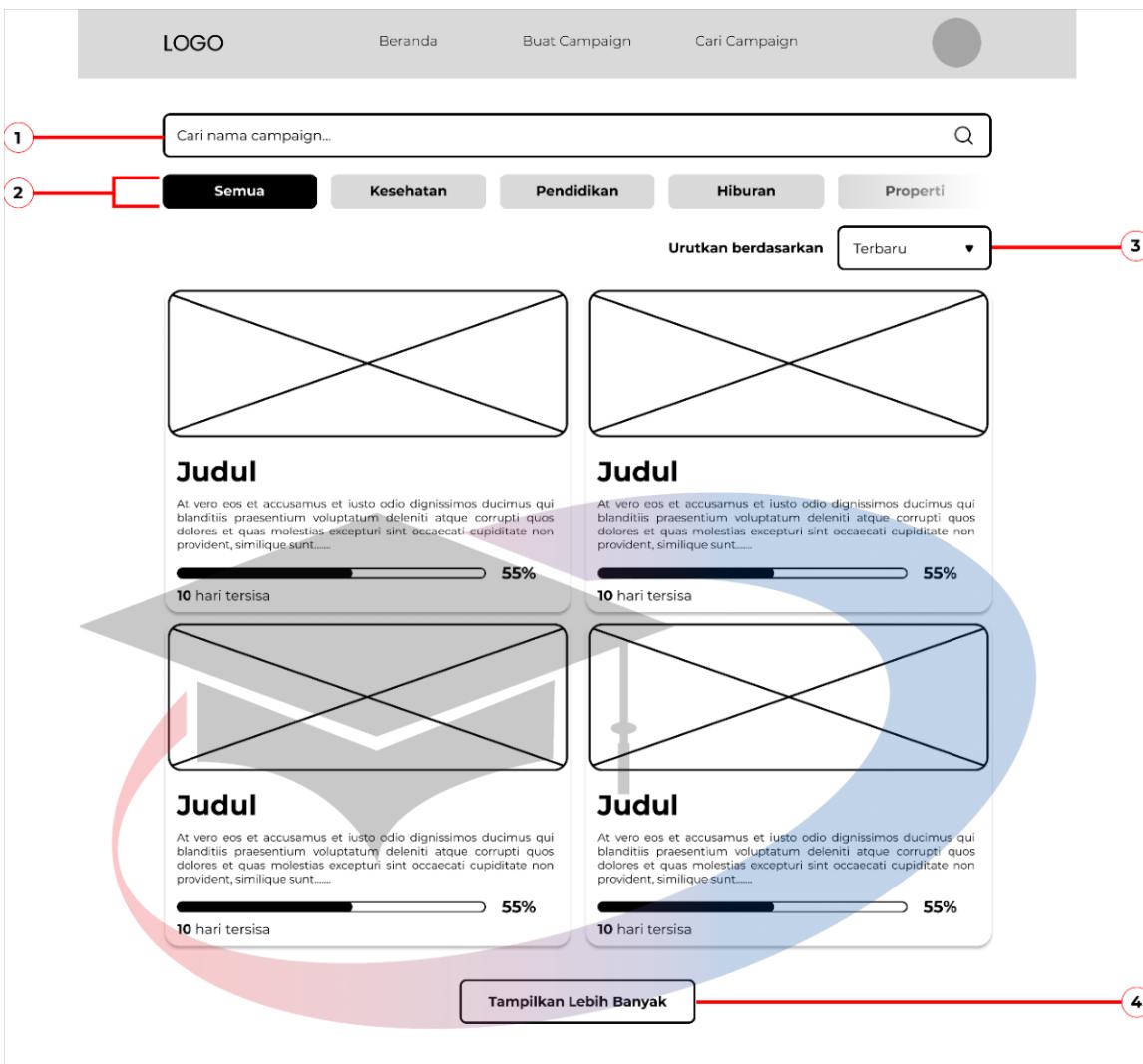
- (1) *Button*, untuk mengarahkan *user* ke halaman Beranda.
- (2) *Card*, berisi informasi *campaign* dan akan mengarahkan *user* ke halaman Detail *Campaign* jika diklik.
- (3) *Text link*, untuk mengarahkan *user* ke halaman Beranda.
- (4) *Text link*, untuk mengarahkan *user* ke halaman Buat *Campaign*.
- (5) *Text link*, untuk mengarahkan *user* ke halaman Cari *Campaign*.
- (6) *Button*, untuk membuka *menu overlay Profil*.
- (7) *Button*, untuk mengarahkan *user* ke halaman Profil.
- (8) *Button*, untuk *disconnect wallet* dan *sign out*.

c. Halaman

Cari

Campaign

Gambar 3.20 merupakan rancangan tampilan halaman yang digunakan oleh *user* untuk mencari *campaign* pada aplikasi. Pencarian dapat dilakukan melalui kolom pencarian, filter kategori ataupun filter waktu.



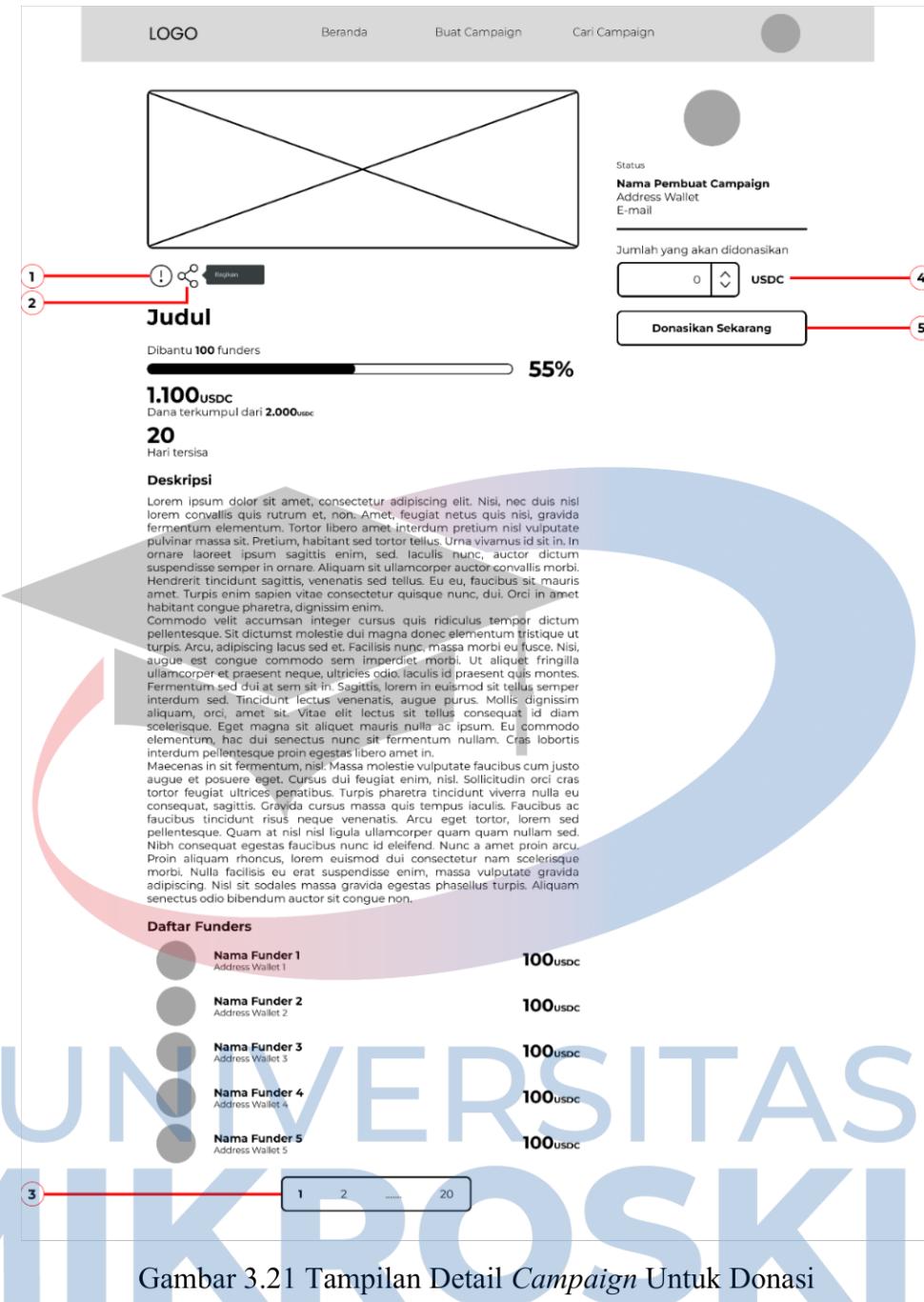
Gambar 3.20 Tampilan Cari Campaign

Keterangan :

- (1) *Text input*, untuk menerima inputan *text* pencarian *campaign*.
- (2) *Category*, untuk memfilter *campaign* berdasarkan *category* yang dipilih.
- (3) *Select input*, untuk memilih filter pengurutan *campaign* yang ditampilkan.
- (4) *Button*, untuk memuat *campaign* agar ditampilkan lebih banyak.

d. Halaman Detail *Campaign* untuk melakukan donasi

Gambar 3.21 merupakan rancangan tampilan halaman yang digunakan *user* untuk melihat informasi detail dari *campaign* dan melakukan donasi.



Gambar 3.21 Tampilan Detail *Campaign* Untuk Donasi

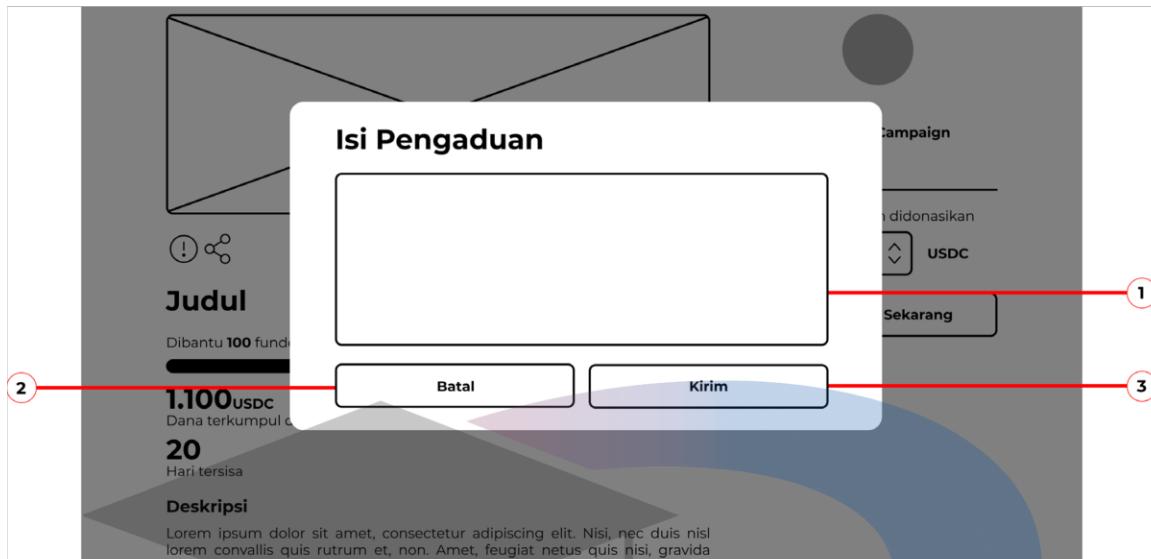
Keterangan :

- (1) *Button*, untuk menampilkan *modal* pengaduan *campaign*.
- (2) *Button*, untuk menyalin link *campaign*.
- (3) *Pagination*, untuk navigasi halaman Daftar *Funders*.
- (4) *Numeric input*, untuk menerima inputan nominal donasi.
- (5) *Button*, untuk melakukan donasi.

e. *Modal* Pengaduan *Campaign*

Gambar 3.22 merupakan rancangan tampilan *modal* yang merupakan bagian dari

halaman Detail *Campaign* yang digunakan *user* untuk mengadukan *campaign* yang dianggap mencurigakan.



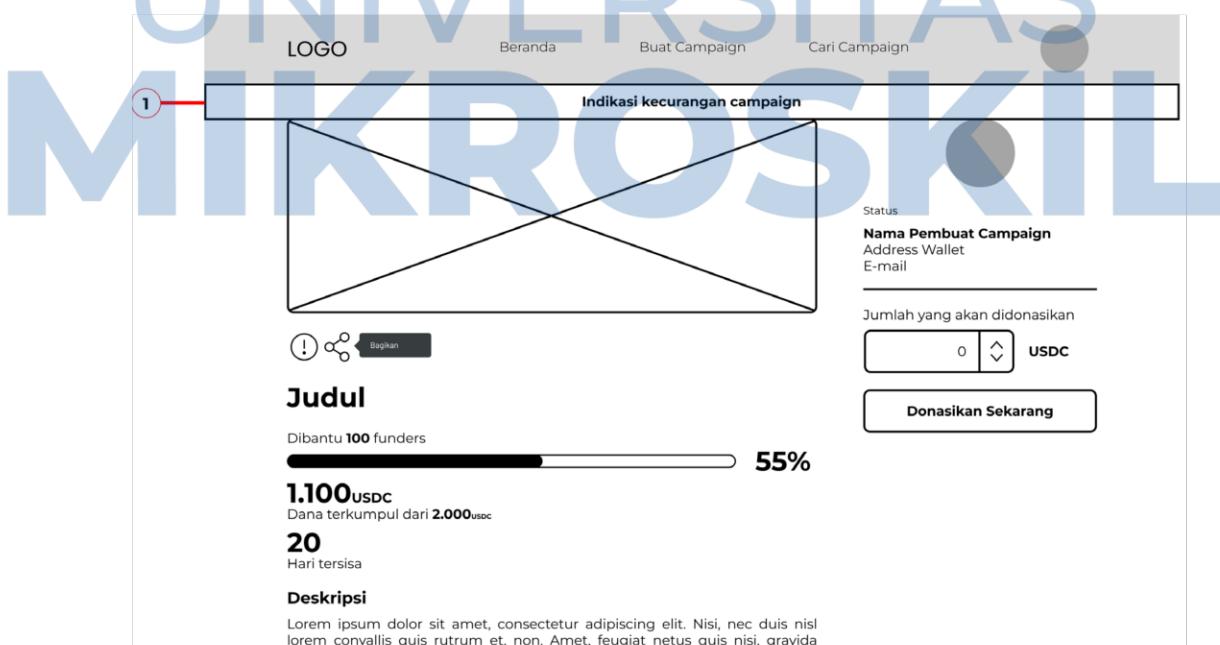
Gambar 3.22 Tampilan Modal Pengaduan Campaign

Keterangan :

- (1) *Text input*, untuk menerima inputan *text* pengaduan.
- (2) *Button*, untuk membatalkan pengaduan dan menutup *modal*.
- (3) *Button*, untuk mengirimkan pengaduan dan menutup *modal*.

f. Halaman Detail *Campaign* untuk *campaign* yang terindikasi pelanggaran.

Gambar 3.23 merupakan rancangan tampilan halaman yang digunakan *user* untuk melihat detail *campaign* dari *campaign* yang memiliki label terindikasi pelanggaran.



Gambar 3.23 Tampilan Halaman Detail Campaign Terindikasi Pelanggaran

Keterangan :

- (1) *Text link*, untuk membuka halaman isi pengaduan yang telah diadukan oleh *user* lain.

g. Halaman pengaduan *campaign*

Gambar 3.24 merupakan rancangan tampilan halaman yang digunakan *user* untuk melihat daftar pengaduan dari *campaign* yang memiliki label terindikasi pelanggaran.



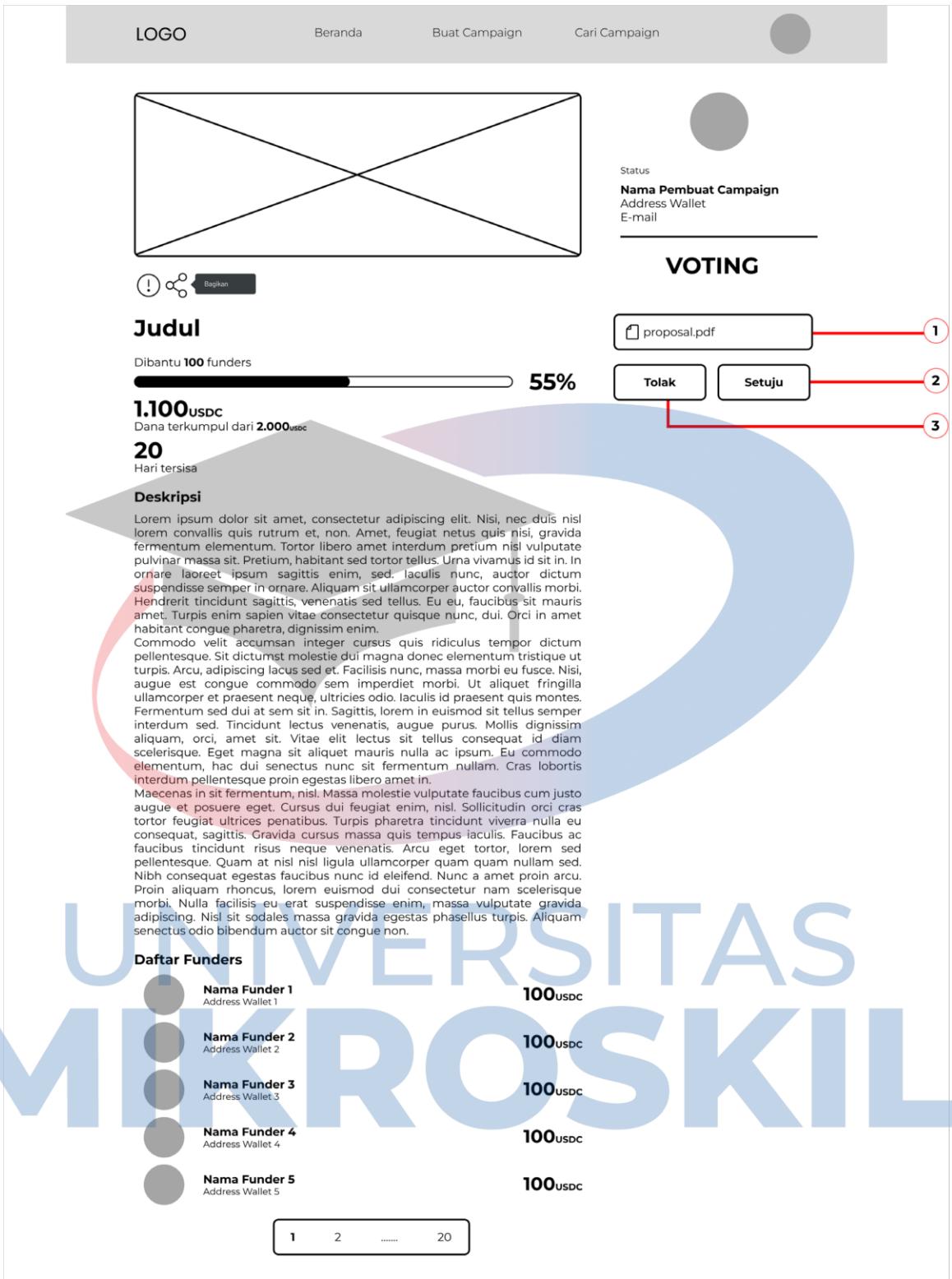
Gambar 3.24 Tampilan Halaman Pengaduan *Campaign*

Keterangan :

- (1) *Pagination*, untuk navigasi halaman pengaduan *campaign*.

h. Halaman Detail *Campaign* untuk melakukan voting

Gambar 3.25 merupakan rancangan tampilan halaman yang digunakan user untuk melihat detail *campaign* dari *campaign* yang sedang mengajukan permintaan pencairan dana dan untuk melakukan voting.



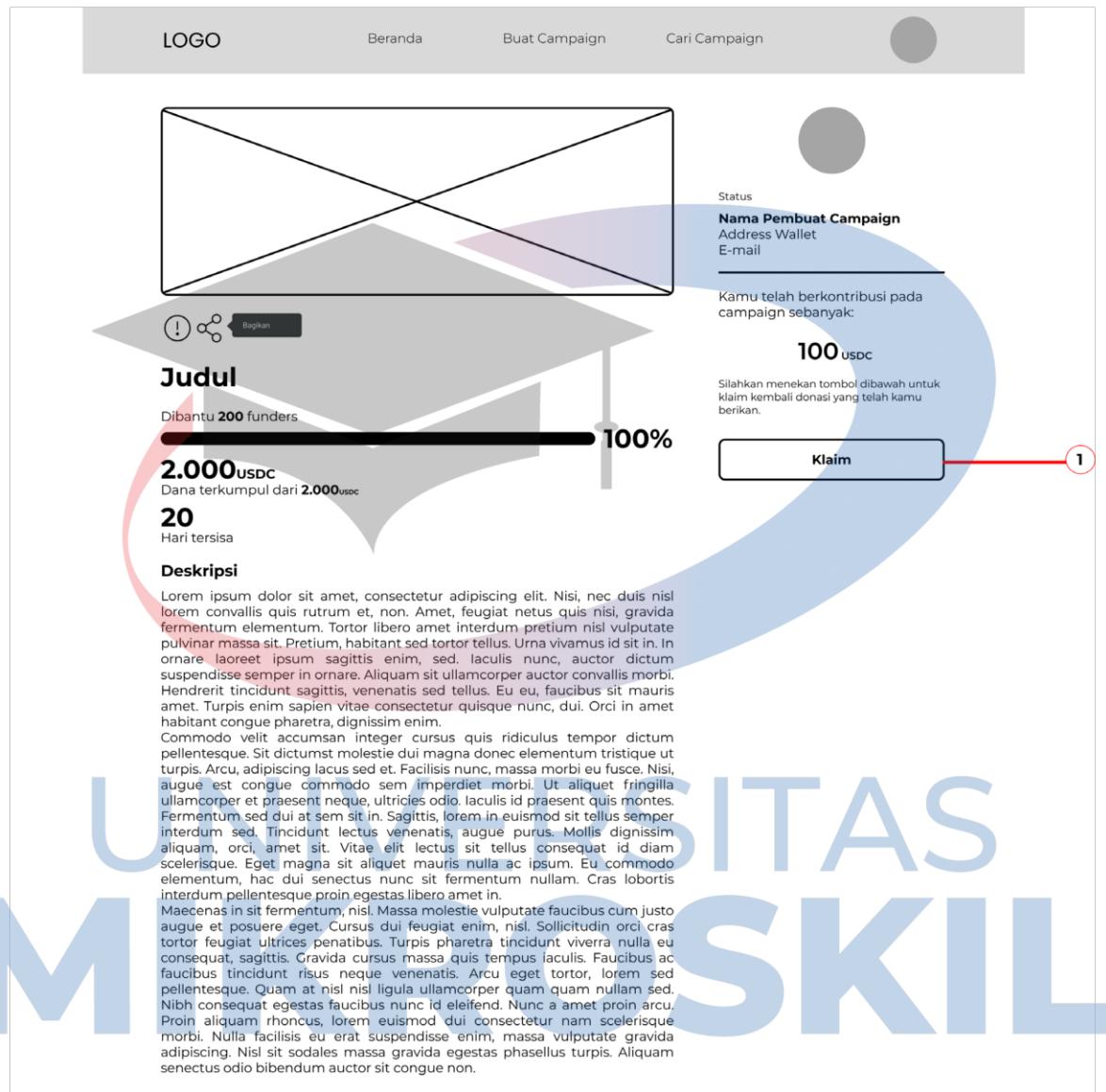
Gambar 3.25 Tampilan Detail Campaign Untuk Voting

Keterangan :

- (1) *File*, untuk melihat *file* proposal.
- (2) *Button*, untuk menyetujui proposal.
- (3) *Button*, untuk menolak proposal.

i. Halaman Detail *Campaign* untuk klaim dana *refund*

Gambar 3.26 merupakan rancangan tampilan halaman yang digunakan *user* untuk melihat detail *campaign* dari *campaign* yang gagal, yaitu *campaign* dengan status *Not Filled* dan *Not Funded*, kemudian digunakan untuk melakukan klaim dana *refund*.



Gambar 3.26 Tampilan Detail *Campaign* Untuk *Refund*

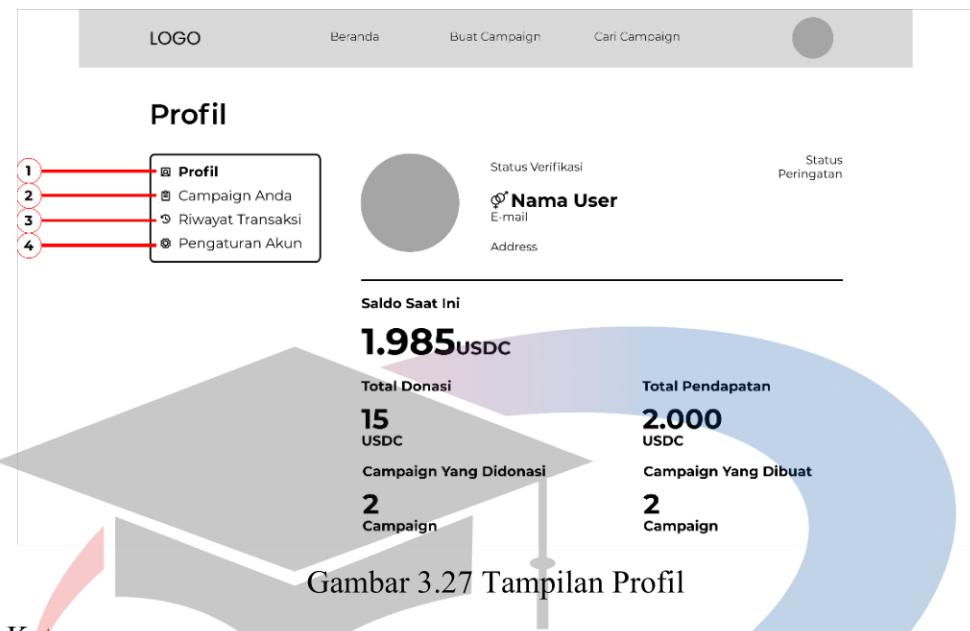
Keterangan :

- (1) *Button*, untuk mengklaim dana yang di-*refund*

j. Halaman

Profil

Gambar 3.27 merupakan rancangan tampilan halaman yang digunakan *user* untuk melihat profil akunnya.



Gambar 3.27 Tampilan Profil

Keterangan :

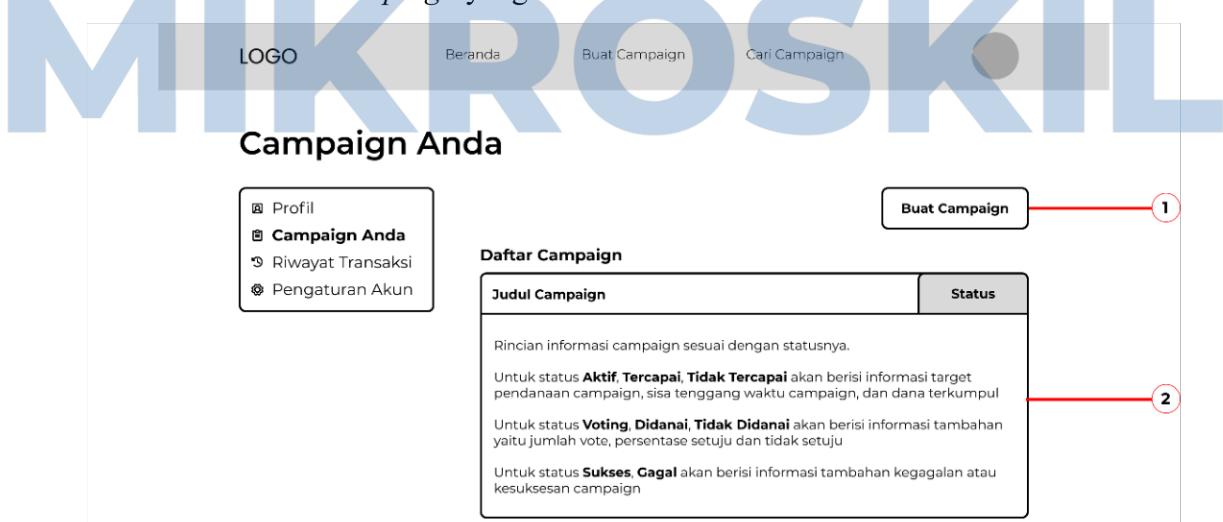
- (1) *Text link*, untuk mengarahkan *user* ke halaman Profil.
- (2) *Text link*, untuk mengarahkan *user* ke halaman Campaign Anda.
- (3) *Text link*, untuk mengarahkan *user* ke halaman Riwayat Transaksi.
- (4) *Text link*, untuk mengarahkan *user* ke halaman Pengaturan Akun.

k. Halaman

Campaign

Anda

Gambar 3.28 merupakan rancangan tampilan halaman yang digunakan *user* untuk melihat daftar *campaign* yang telah dibuat.



Gambar 3.28 Tampilan Campaign Anda

Keterangan :

- (1) *Button*, untuk mengarahkan *user* ke halaman Buat Campaign.
- (2) *Button*, untuk mengarahkan *user* ke halaman Detail Campaign.

1. Halaman Buat *Campaign*

Gambar 3.29 merupakan rancangan tampilan halaman yang digunakan *user* untuk membuat *campaign*.

The screenshot shows the 'Buat Campaign' form with various input fields and a sidebar menu. Red numbers 1 through 6 are overlaid on the interface to indicate specific user interactions:

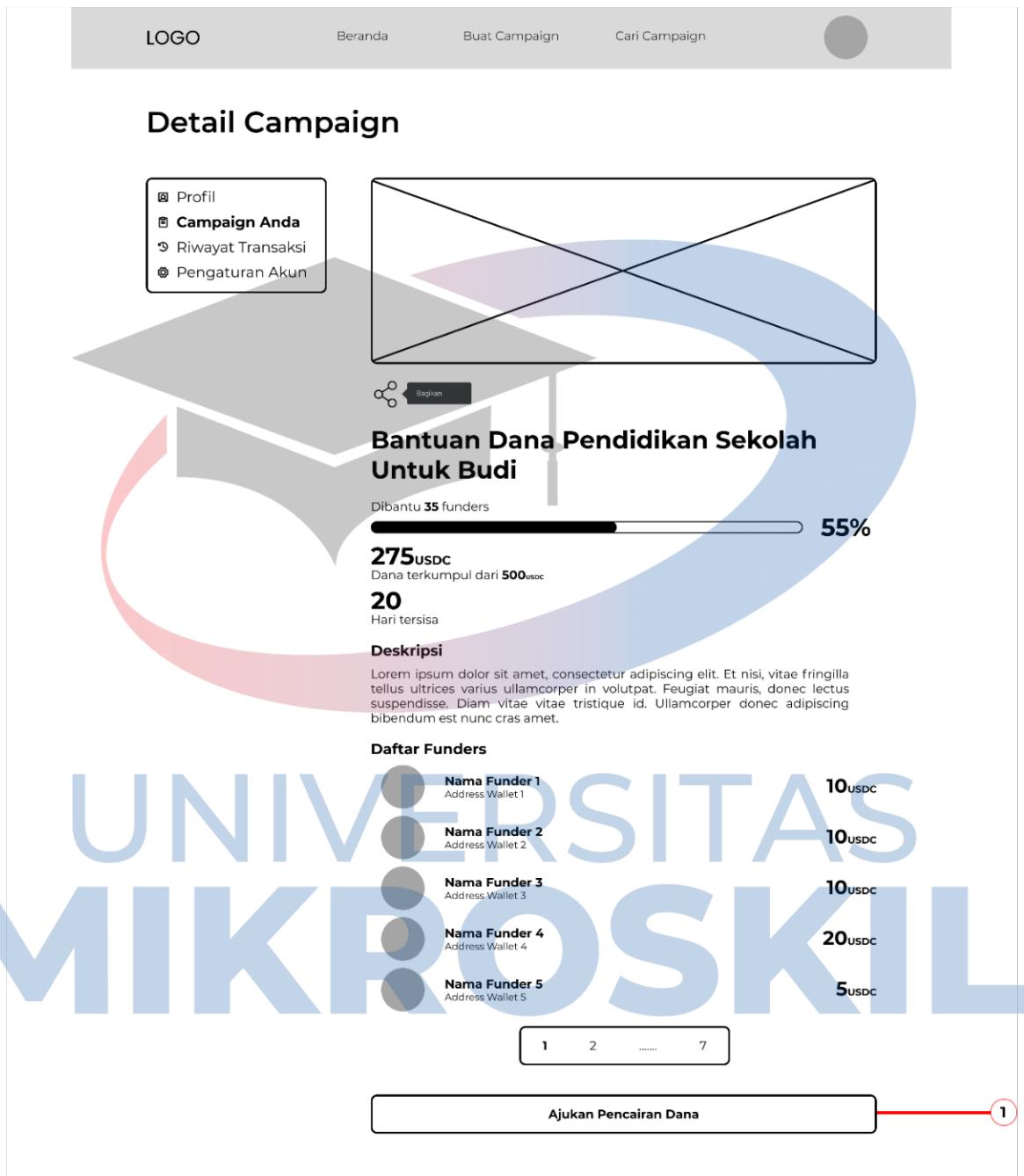
- 1. Points to the 'Kategori' dropdown menu.
- 2. Points to the 'Target Pendanaan' numeric input field.
- 3. Points to the 'Upload Gambar' file input field, which is crossed out with a large 'X'.
- 4. Points to the 'Judul' text input field.
- 5. Points to the 'Deskripsi' text input field.
- 6. Points to the 'Buat Campaign' button at the bottom.

Gambar 3.29 Tampilan Buat Campaign

Keterangan :

- (1) *Select input*, untuk memilih kategori *campaign*.
- (2) *Numeric input*, untuk menerima inputan nominal target pendanaan.
- (3) *File*, untuk menerima inputan berupa *file* gambar.
- (4) *Text input*, untuk menerima inputan judul *campaign*.
- (5) *Text input*, untuk menerima inputan deskripsi *campaign*.
- (6) *Button*, untuk membuat *campaign*.

m. Halaman Detail *Campaign* untuk pengajuan pencairan dana. Gambar 3.30 merupakan rancangan tampilan halaman yang digunakan *user* untuk melihat detail *campaign* dari *campaign* yang telah berhasil mencapai target pendanaan.



Gambar 3.30 Tampilan Detail *Campaign* Untuk Pengajuan Pencairan Dana

Keterangan :

- (1) *Button*, untuk menampilkan modal Pengajuan Pencairan.

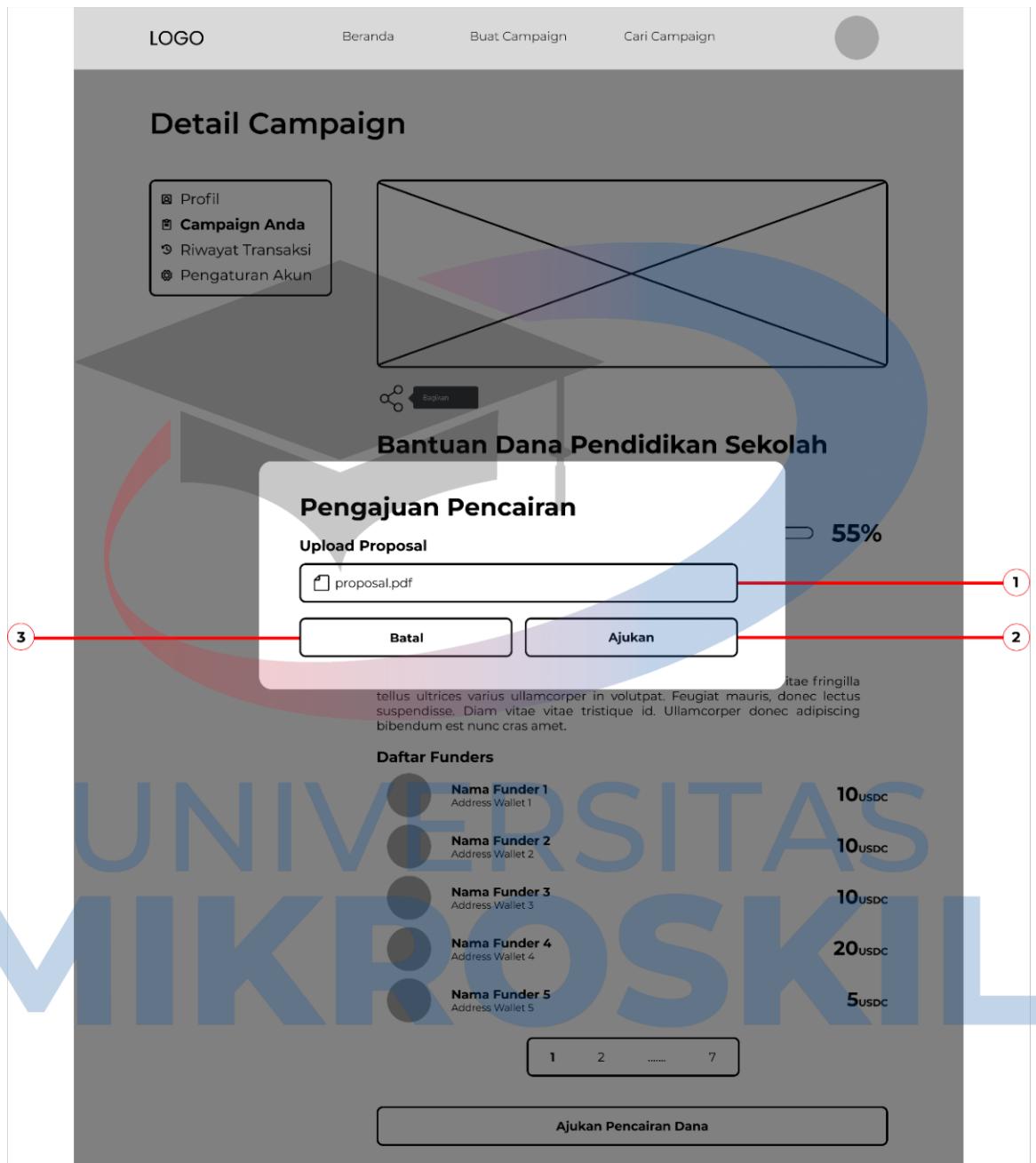
n. *Modal*

Pencairan

Dana

Campaign

Gambar 3.31 merupakan rancangan tampilan *modal* yang merupakan bagian dari halaman Detail *Campaign* yang digunakan *user* untuk mengajukan pencairan dana *campaign*.

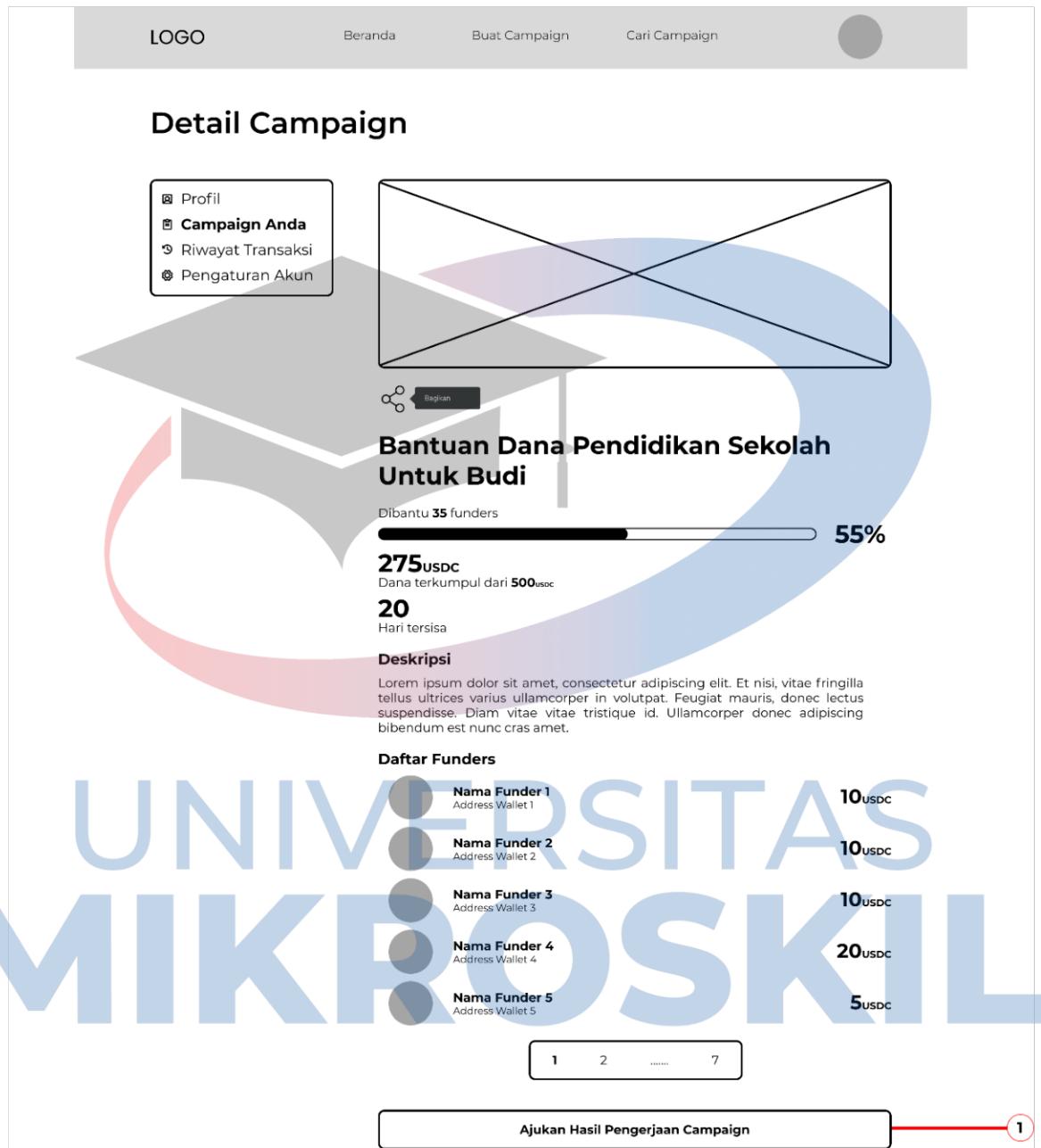


Gambar 3.31 Tampilan *Modal* Pencairan Dana *Campaign*

Keterangan :

- (1) *File input*, untuk meng-upload file proposal.
- (2) *Button*, untuk mengajukan pencairan dana.
- (3) *Button*, untuk batal mengajukan pencairan dana.

- o. Halaman Detail *Campaign* pengajuan verifikasi kesuksesan *campaign*
- Gambar 3.32 merupakan rancangan tampilan halaman yang digunakan *user* untuk melihat detail *campaign* dari *campaign* yang telah mendapatkan pendanaan, dan untuk mengajukan verifikasi kesuksesan *campaign*.



Gambar 3.32 Tampilan Detail *Campaign* Untuk Pengajuan Verifikasi *Campaign*

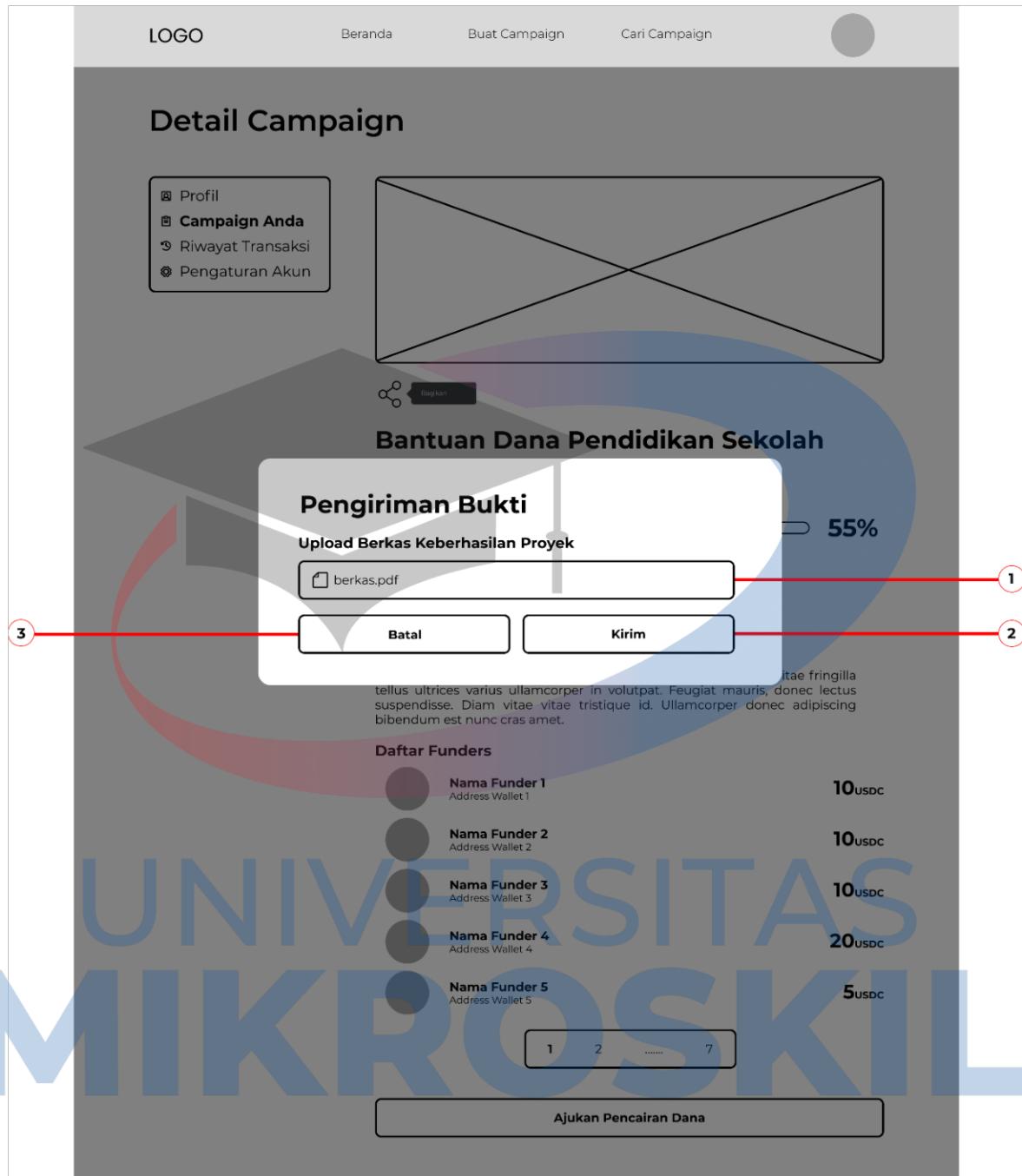
Keterangan :

- (1) *Button*, untuk untuk mengarahkan *user* ke halaman Validasi *Campaign*.

- p. *Modal* Verifikasi Kesuksesan *Campaign*

Gambar 3.33 merupakan rancangan tampilan *modal* yang merupakan bagian dari

halaman Detail *Campaign* yang digunakan *user* untuk mengajukan berkas verifikasi kesuksesan *campaign*.



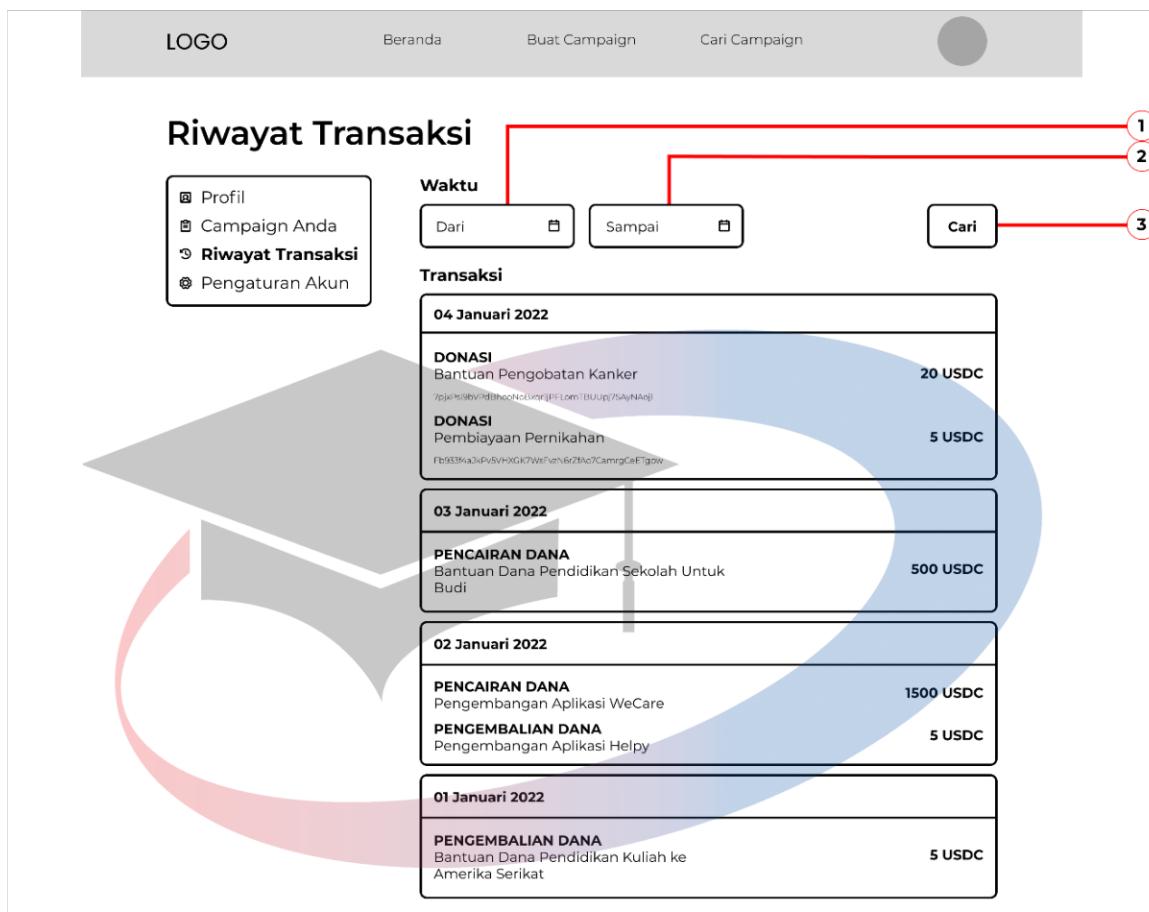
Gambar 3.33 Tampilan Modal Verifikasi Kesuksesan *Campaign*

Keterangan :

- (1) *File input*, untuk meng-*upload* berkas.
- (2) *Button*, untuk mengirimkan berkas.
- (3) *Button*, untuk batal mengirimkan berkas.

q. Halaman Riwayat Transaksi

Gambar 3.34 merupakan rancangan tampilan halaman yang digunakan *user* untuk melihat riwayat transaksi.



Gambar 3.34 Tampilan Riwayat Transaksi

Keterangan :

- (1) *Select date*, untuk menentukan batas awal tanggal pada filter.
- (2) *Select date*, untuk menentukan batas akhir tanggal pada filter.
- (3) *Button*, untuk mencari riwayat transaksi berdasarkan filter.

r. Halaman Pengaturan Akun

Gambar 3.35 merupakan rancangan tampilan halaman yang digunakan *user* untuk mengelola keperluan informasi akun.

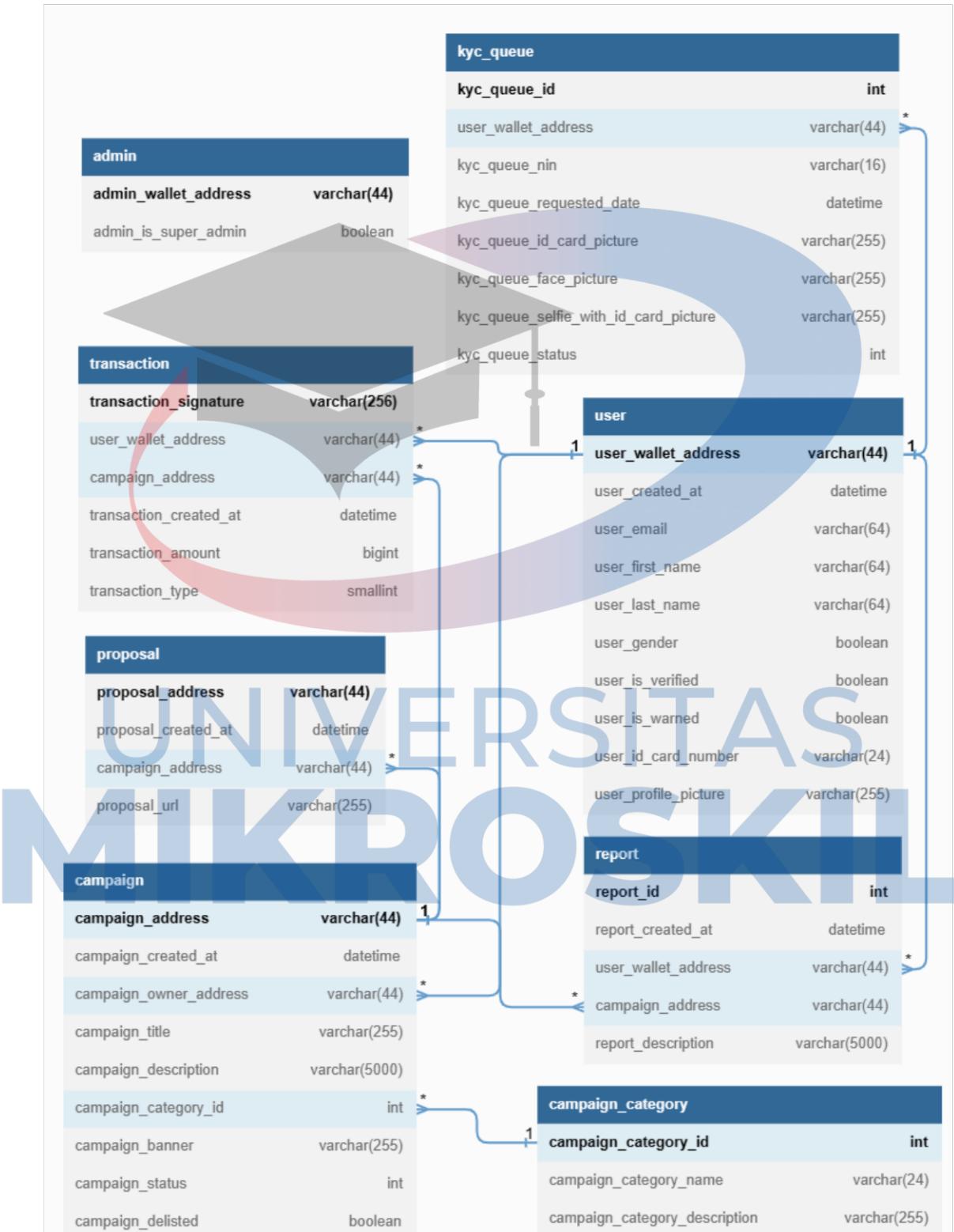
Gambar 3.35 Tampilan Pengaturan Akun

Keterangan :

- (1) *Text input*, untuk menerima inputan nama depan.
- (2) *Text input*, untuk menerima inputan nama belakang.
- (3) *Text input*, untuk menerima inputan *email*.
- (4) *Radio button*, untuk menerima inputan gender.
- (5) *Button*, untuk menyimpan data.
- (6) *Text input*, untuk menerima inputan nomor KTP.
- (7) *File input*, untuk menerima inputan berupa KTP *user*
- (8) *Text link*, untuk membuka petunjuk pengambilan foto KYC
- (9) *File input*, untuk menerima inputan berupa foto *selfie* dari *user*
- (10) *File input*, untuk menerima inputan berupa foto *selfie* dan foto KTP *user*
- (11) *Button*, untuk mengajukan verifikasi.

3.2.2 Perancangan Basis Data

Basis data yang digunakan pada aplikasi ini dirancang menggunakan PostgreSQL yang merupakan basis data SQL. Perancangan basis data dimodelkan dengan menggunakan *Entity Relationship Diagram* (ERD) yang dapat dilihat pada Gambar 3.36.



Gambar 3.36 ERD Rancangan Basis Data

Setiap atribut dari tiap tabel *database* akan dijelaskan lebih detail dari Tabel 3.21 hingga Tabel 3.28.

1. Tabel *User*

Tabel 3.21 Struktur Tabel *User*

Nama Atribut	Tipe Data	Keterangan
<i>user_wallet_address</i> [PK]	<i>varchar</i> (44)	<i>Wallet address</i> dari <i>user</i>
<i>user_created_at</i>	<i>datetime</i>	Waktu data <i>user</i> dibuat
<i>user_email</i>	<i>varchar</i> (64)	Email <i>user</i>
<i>user_first_name</i>	<i>varchar</i> (64)	Nama depan <i>user</i>
<i>user_last_name</i>	<i>varchar</i> (64)	Nama belakang <i>user</i>
<i>user_gender</i>	<i>boolean</i>	Gender <i>user</i> : <i>true</i> = pria <i>false</i> = wanita
<i>user_is_verified</i>	<i>boolean</i>	Apakah <i>user</i> sudah terverifikasi
<i>user_is_warned</i>	<i>boolean</i>	Apakah <i>user</i> pernah di-warning
<i>user_id_card_number</i>	<i>varchar</i> (24)	Nomor identitas <i>user</i>
<i>user_profile_picture</i>	<i>varchar</i> (255)	URL foto profil <i>user</i>

2. Tabel *Campaign*

Tabel 3.22 Struktur Tabel *Campaign*

Nama Atribut	Tipe Data	Keterangan
<i>campaign_address</i> [PK]	<i>varchar</i> (44)	Address dari <i>campaign</i>
<i>campaign_created_at</i>	<i>datetime</i>	Waktu <i>campaign</i> dibuat
<i>campaign_owner_address</i> [FK]	<i>varchar</i> (44)	<i>Wallet address</i> dari pemilik <i>campaign</i>
<i>campaign_title</i>	<i>varchar</i> (255)	Judul <i>campaign</i>
<i>campaign_description</i>	<i>varchar</i> (5000)	Deskripsi <i>campaign</i>
<i>campaign_category_id</i> [FK]	<i>int</i>	ID dari <i>campaign category</i>
<i>campaign_banner</i>	<i>varchar</i> (255)	URL foto <i>campaign banner</i>
<i>campaign_status</i>	<i>int</i>	Status <i>evidence</i> dari <i>campaign</i> :

		0 = <i>Waiting</i> , campaign belum mengajukan <i>request</i> untuk memverifikasi kesuksesan campaign 1 = <i>Requested</i> , bukti pelaksanaan campaign telah diajukan dan dalam masa review oleh admin. 2 = <i>Success</i> , admin telah memverifikasi kesuksesan campaign, dan campaign dinyatakan sukses 3 = <i>Failed</i> , admin menolak bukti yang diajukan dan campaign dinyatakan gagal
<i>campaign_delisted</i>	<i>boolean</i>	Apakah campaign sudah <i>delisted</i>

3. Tabel *Campaign Category*

Tabel 3.23 Struktur Tabel *Campaign Category*

Nama Atribut	Tipe Data	Keterangan
<i>campaign_category_id</i> [PK]	<i>int</i>	<i>ID</i> dari campaign category
<i>campaign_category_name</i>	<i>varchar</i> (24)	Nama category
<i>campaign_category_description</i>	<i>varchar</i> (255)	Deskripsi category

4. Tabel *Transaction*

Tabel 3.24 Struktur Tabel *Transaction*

Nama Atribut	Tipe Data	Keterangan
<i>transaction_signature</i> [PK]	<i>varchar</i> (256)	<i>Signature</i> dari transaksi yang dilakukan
<i>user_wallet_address</i> [FK]	<i>varchar</i> (44)	<i>Wallet address</i> dari user yang melakukan transaksi
<i>campaign_address</i> [FK]	<i>varchar</i> (44)	<i>Address</i> dari campaign yang berkaitan
<i>transaction_created_at</i>	<i>datetime</i>	Waktu transaksi
<i>transaction_amount</i>	<i>bigint</i>	Nominal transaksi
<i>transaction_type</i>	<i>smallint</i>	Tipe transaksi: 0 = donasi 1 = refund 2 = klaim dana <i>fundraising</i>

5. Tabel KYC (*Know Your Customer*) Queue

Tabel 3.25 Struktur Tabel KYC Queue

Nama Atribut	Tipe Data	Keterangan
<i>kyc_queue_id</i> [PK]	<i>int</i>	<i>ID</i> dari antrian verifikasi KYC
<i>user_wallet_address</i> [FK]	<i>varchar (44)</i>	<i>Wallet address</i> dari <i>user</i> yang mengajukan verifikasi
<i>kyc_queue_nin</i>	<i>varchar (16)</i>	<i>National Identity Number</i> dari <i>user</i>
<i>kyc_queue_requested_date</i>	<i>datetime</i>	Waktu pengajuan verifikasi
<i>kyc_queue_id_card_picture</i>	<i>varchar (255)</i>	<i>URL</i> foto kartu identitas <i>user</i>
<i>kyc_queue_face_picture</i>	<i>varchar (255)</i>	<i>URL</i> foto <i>user</i>
<i>kyc_queue_selfie_with_id_card_picture</i>	<i>varchar (255)</i>	<i>URL</i> foto <i>user</i> bersama kartu identitasnya
<i>kyc_queue_status</i>	<i>int</i>	Status KYC: 0 = <i>pending</i> 1 = <i>accepted</i> 2 = <i>declined</i> 3 = <i>removed</i>

6. Tabel Report

Tabel 3.26 Struktur Tabel Report

Nama Atribut	Tipe Data	Keterangan
<i>report_id</i> [PK]	<i>int</i>	<i>ID</i> dari pengaduan
<i>report_created_at</i>	<i>datetime</i>	Waktu pengaduan dibuat
<i>user_wallet_address</i> [FK]	<i>varchar (44)</i>	<i>Wallet address</i> dari <i>user</i> pengadu
<i>campaign_address</i> [FK]	<i>varchar (44)</i>	<i>Address</i> dari <i>campaign</i> yang diadukan
<i>report_description</i>	<i>varchar (5000)</i>	Deskripsi pengaduan

7. Tabel Admin

Tabel 3.27 Struktur Tabel Admin

Nama Atribut	Tipe Data	Keterangan
<i>admin_wallet_address</i> [PK]	<i>varchar (44)</i>	<i>Wallet address</i> dari admin

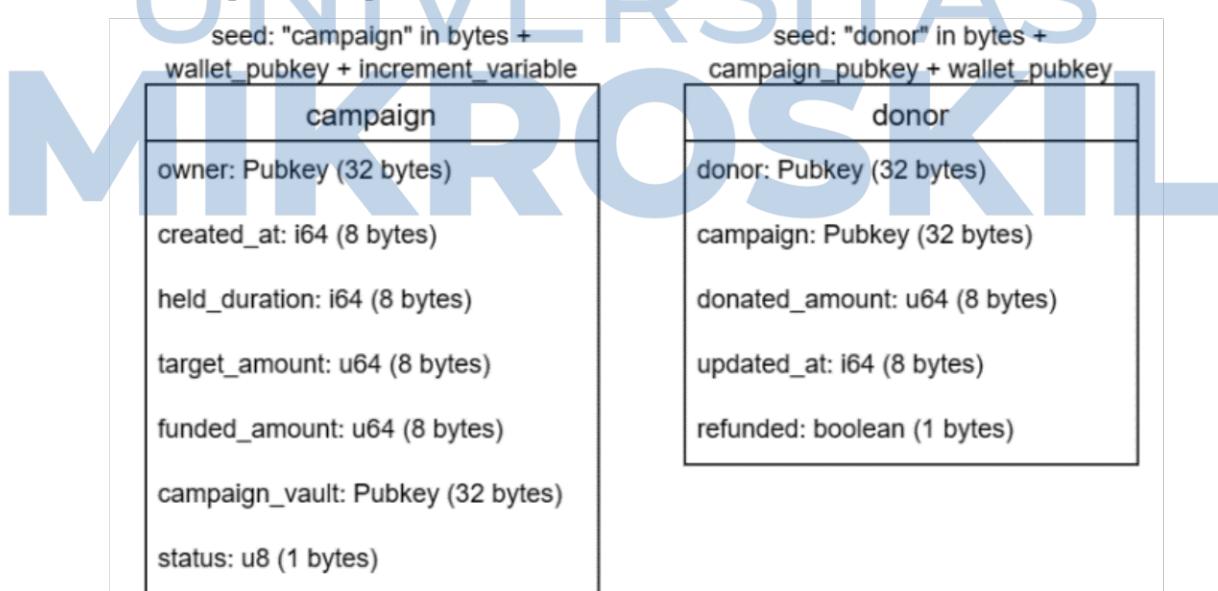
<i>admin_is_super_admin</i>	<i>boolean</i>	Apakah status <i>admin</i> adalah <i>super admin</i>
-----------------------------	----------------	--

8. Tabel Proposal

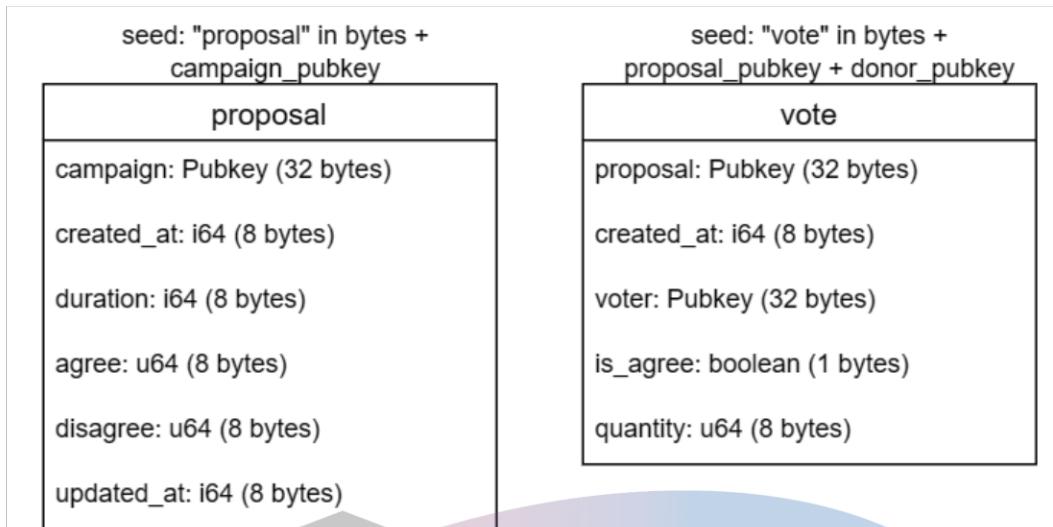
Tabel 3.28 Struktur Tabel Proposal

Nama Atribut	Tipe Data	Keterangan
<i>proposal_address</i> [PK]	<i>varchar</i> (44)	Address dari proposal
<i>proposal_created_at</i>	<i>datetime</i>	Waktu proposal dibuat
<i>campaign_address</i> [FK]	<i>varchar</i> (44)	Address dari <i>campaign</i>
<i>proposal_url</i>	<i>varchar</i> (255)	URL dari file proposal

Selain memanfaatkan *database* aplikasi untuk penyimpanannya, aplikasi ini juga memanfaatkan penyimpanan *blockchain*. Tidak seperti *database* pada umumnya, *blockchain* mencatat data dalam sebuah *ledger*. Agar dapat tersimpan dalam *blockchain*, data harus dikonversi ke dalam bentuk *bytes* terlebih dahulu. Dalam Solana, ada sebuah istilah yang digunakan yaitu *account*. *Account* merupakan sebuah wadah dalam *ledger* yang dapat digunakan untuk menyimpan data maupun *coin* yang juga disebut *lamports*. Tiap *account* memiliki *address*-nya masing – masing yang di-generate menggunakan *seed*. *Seed* merupakan kombinasi *bytes* yang digunakan untuk di-*hashing*. Hasil *hashing* tersebut akan digunakan sebagai *address* dari sebuah *account*. Rancangan penyimpanan dengan *blockchain* dapat dilihat pada Gambar 3.37 dan Gambar 3.38.



Gambar 3.37 Penyimpanan *Blockchain* - 1



Gambar 3.38 Penyimpanan Blockchain - 2

Setiap atribut pada tiap *account* di dalam *ledger* akan dijelaskan lebih detail pada Tabel 3.29 hingga Tabel 3.32.

1. Campaign

Tabel 3.29 Account Campaign

Nama Atribut	Tipe Data	Keterangan
owner	Pubkey (32 bytes)	Wallet address dari fundraiser
created_at	i64 (8 bytes)	Waktu saat <i>campaign</i> dibuat
held_duration	i64 (8 bytes)	Durasi penyelenggaraan <i>campaign</i>
target_amount	u64 (8 bytes)	Target dana yang ingin dikumpulkan
funded_amount	u64 (8 bytes)	Jumlah dana yang sudah dikumpulkan
campaign_vault	Pubkey (32 bytes)	Address dari <i>token account</i> dimana dana <i>campaign</i> akan disimpan
status	u8 (1 bytes)	Status dari <i>campaign</i> : 0 = <i>Active</i> , <i>campaign</i> menerima donasi 1 = <i>Filled</i> , <i>campaign</i> tidak menerima donasi karena telah mencapai target pendanaan 2 = <i>Not Filled</i> , <i>campaign</i> tidak menerima donasi karena gagal mencapai target pendanaan dan tenggang waktu habis

		<p>3 = Voting, <i>campaign</i> yang sebelumnya memiliki status <i>Filled</i> dan saat ini telah mengajukan pencairan dana</p> <p>4 = <i>Funded</i>, <i>campaign</i> yang sebelumnya memiliki status <i>Fund Claimable</i> dan dana telah diklaim pemilik <i>campaign</i></p> <p>5 = <i>Not Funded</i>, <i>campaign</i> yang sebelumnya memiliki status <i>Voting</i> dan gagal mendapatkan pencairan dana</p> <p>6 = <i>Fund Claimable</i>, <i>campaign</i> yang sebelumnya memiliki status <i>Voting</i> dan berhasil mendapatkan pencairan dana</p>
--	--	---

2. Donor

Tabel 3.30 *Account Donor*

Nama Atribut	Tipe Data	Keterangan
donor	Pubkey (32 bytes)	<i>Address</i> dari <i>funder</i>
<i>campaign</i>	Pubkey (32 bytes)	<i>Address</i> dari <i>campaign</i> yang didonasikan
<i>donated_amount</i>	u64 (8 bytes)	Jumlah uang yang didonasikan
<i>updated_at</i>	i64 (8 bytes)	Waktu saat terjadi penambahan jumlah donasi
<i>refunded</i>	boolean (1 bytes)	Untuk mengetahui apakah uang donatur telah di-refund atau belum

3. Proposal

Tabel 3.31 *Account Proposal*

Nama Atribut	Tipe Data	Keterangan
<i>campaign</i>	Pubkey (32 bytes)	<i>Address</i> dari <i>campaign</i> dimana proposal dibuat
<i>created_at</i>	i64 (8 bytes)	Waktu saat proposal dibuat
<i>duration</i>	i64 (8 bytes)	Durasi penyelenggaraan voting
<i>agree</i>	u64 (8 bytes)	Total jumlah suara yang setuju

<i>disagree</i>	u64 (8 bytes)	Total jumlah suara yang tidak setuju
<i>updated_at</i>	i64 (8 bytes)	Waktu saat terjadi perubahan suara voting yang dilakukan <i>funder</i>

4. Vote

Tabel 3.32 Account Vote

Nama Atribut	Tipe Data	Keterangan
<i>proposal</i>	<i>Pubkey (32 bytes)</i>	<i>Address</i> dari proposal yang di-vote
<i>created_at</i>	i64 (8 bytes)	Waktu saat pemberian suara
<i>voter</i>	<i>Pubkey (32 bytes)</i>	<i>Address</i> dari <i>funder</i> yang melakukan <i>vote</i>
<i>is_agree</i>	<i>boolean (1 bytes)</i>	Apakah <i>voter</i> setuju atau tidak
<i>quantity</i>	u64 (8 bytes)	Besar suara yang dimiliki oleh <i>voter</i>

3.3 Perancangan Pengujian Pada Smart Contract

Pada subbab ini akan dilampirkan rancangan pengujian keamanan yang akan diuji ke *smart contract* dari setiap transaksi pada aplikasi dikarenakan titik lemah Solana berada pada *smart contract* yang dibuat. Berikut pengujian keamanan yang dilakukan pada tiap transaksi:

1. Pengujian transaksi melakukan donasi pada *campaign*:
 - a. Pengecekan kecukupan jumlah donasi
 - b. Pengecekan akun donor (*funder*)
2. Pengujian transaksi pengajuan proposal pencairan dana:
 - a. Pengecekan status *campaign*
 - b. Pengecekan *fundraiser address*
3. Pengujian transaksi melakukan voting pada proposal:
 - a. Pengecekan akun donor (*funder*)
 - b. Pengecekan *authority* akun donor
4. Pengujian transaksi melakukan klaim dana *campaign*:
 - a. Pengecekan *fundraiser address*
 - b. Pengecekan status *campaign* yang akan diklaim
5. Pengujian transaksi melakukan klaim dana *refund*:
 - a. Pengecekan akun donor (*funder*)
 - b. Pengecekan status *refund*