

BAB III

ANALISIS DAN PERANCANGAN

3.1 Analisis

3.1.1 Identifikasi Masalah

Data pendukung pada penelitian ini didapatkan dari kuesioner penelitian yang merupakan hasil jawaban dari responden, dimana kuesioner tersebut berisi pertanyaan terkait dengan pembuatan thumbnail sebagai konten kreator. Pada bab ini akan dijelaskan secara detail pengolahan data dari kuesioner tersebut.

Data kuesioner akan dibagi menjadi dua kategori yaitu data diri dari responden serta jawaban dari pertanyaan yang menyangkut penelitian. Kategori pertama yang berupa data diri responden dapat dilihat pada Tabel 3.1 sampai dengan Tabel 3.5.

Salah satu kategori untuk data diri responden berupa jenis kelamin. Jenis kelamin akan dibedakan menjadi dua kategori yaitu laki - laki dan perempuan. Pada Tabel 3.1 memperlihatkan jenis kelamin dari responden yang menunjukkan bahwa mayoritas berjenis kelamin laki - laki sebanyak 8 responden.

Tabel 3.1 Jenis Kelamin Responden

Jenis Kelamin	Frekuensi	Presentase
Laki - Laki	8	53,3%
Perempuan	7	46,7%

Selanjutnya adalah kategori data diri responden yaitu usia. Pada kuesioner, responden memiliki kebebasan untuk mengetik usia mereka pada kolom yang telah disediakan. Tabel 3.2 menunjukkan bahwa kebanyakan responden berusia pada 22 tahun.

Tabel 3.2 Usia Responden

Usia	Frekuensi	Presentase
14	1	6,7%
15	1	6,7%
21	2	13,3%
22	7	46,7%
23	2	13,3%
24	1	6,7%
27	1	6,7%

Kategori data diri responden setelah usia adalah mereka termasuk konten kreator pemula atau konten kreator yang sudah lama memulai karir atau hobinya. Pada Tabel 3.3 menunjukkan responden mayoritas adalah konten kreator pemula yang memulai profesi atau hobinya kurang dari satu tahun.

Tabel 3.3 Kategori Konten Kreator Responden

Kategori	Frekuensi	Presentase
Konten Kreator Pemula (<1 tahun)	13	86,7%
Konten Kreator Lama (>1 tahun)	2	13,3%

Platform konten kreator yang responden gunakan juga menjadi salah satu kategori dalam data diri responden. Pada penelitian ini, platform akan hanya difokuskan pada platform media sosial bernama YouTube, Instagram, dan TikTok. Pada kuesioner kategori ini responden dapat memilih lebih dari satu pilihan platform. Tabel 3.4 memperlihatkan bahwa 66,7% responden menjadi konten kreator pada platform TikTok

Tabel 3.4 Platform Responden

Platform	Frekuensi	Presentase
YouTube	8	53,3%
Instagram	6	40%
TikTok	10	66,7%

Kategori terakhir untuk data diri responden adalah pengetahuan atau pengalaman dalam bidang desain grafis. Dikarenakan dengan adanya kategori tersebut peneliti dapat menganalisa apakah pengalaman dalam bidang desain grafis akan mempengaruhi seseorang dalam menciptakan sebuah *thumbnail*. Tabel 3.5 Mengindikasikan bahwa mayoritas responden tidak memiliki pengalaman ataupun pengetahuan dalam bidang tersebut.

Tabel 3.5 Pengetahuan Desain Grafis Responden

Pengalaman dalam Desain Grafis	Frekuensi	Presentase
Ya	5	33,3%
Tidak	10	66,7%

Pada kuesioner penelitian diajukan sebanyak delapan pertanyaan. Setiap pertanyaan mengacu apakah responden memiliki masalah dalam menciptakan sebuah *thumbnail*. Serta, apakah mereka membutuhkan software yang dapat membantu untuk membuat *thumbnail* secara efisien dan akurat. Tabel 3.6 pada dibawah ini menunjukkan hasil rincian dari bobot jawaban pada setiap pertanyaan dan jawaban dari responden yang diolah oleh peneliti. Berikut ini daftar delapan pertanyaan dari kuesioner penelitian yang telah disebarkan:

1. Menurut pendapat anda, apakah thumbnail termasuk salah satu elemen yang penting dalam konten media sosial?
2. Menurut anda, seberapa pengaruh apa thumbnail untuk menarik perhatian audiens?
3. Sebagai konten kreator, seberapa sering Anda merasa kesulitan dalam merancang thumbnail untuk konten Anda?
4. Seberapa banyak waktu yang Anda habiskan untuk membuat thumbnail yang sesuai dengan harapan Anda?
5. Seberapa puas Anda dengan hasil thumbnail yang Anda buat sendiri tanpa bantuan profesional?
6. Apakah hasil yang telah Anda buat memerlukan revisi?
7. Seberapa sering Anda merasa bahwa skill desain grafis Anda mempengaruhi kualitas thumbnail yang Anda buat?
8. Apakah anda membutuhkan sebuah software untuk membantu membuat thumbnail secara otomatis?

Tabel 3.6 Penghitungan Data Kuesioner

Pertanyaan	1		2		3		4		5		Total		Mean
	F	%	F	%	F	%	F	%	F	%	F	%	
P.1	1	6,67	0	0	0	0	6	40	8	53,33	15	100	4,33
P.2	0	0	0	0	0	0	5	33,33	10	66,67	15	100	4,67
P.3	0	0	0	0	4	26,67	9	60	2	13,33	15	100	3,87
P.4	2	13,33	6	40	4	26,67	2	13,3	1	6,67	15	100	2,6
P.5	1	6,67	1	6,67	3	20	6	40	4	26,67	15	100	3,73
P.6	0	0	10	66,67	2	13,33	3	20	0	0	15	100	2,53
P.7	0	0	0	0	1	6,67	5	33,33	9	60	15	100	4,53
P.8	2	13,33	0	0	1	6,67	6	40	6	40	15	100	3,93
Grand Mean													3,775

Beberapa aspek dari Tabel 3.6 perlu untuk diperhatikan. Salah satu aspek dari tabel tersebut adalah bobot poin dari jawaban responden. Respon atau jawaban responden memiliki lima macam skor atau skala sebagai berikut.

- Poin 1 = Sangat negatif
- Poin 2 = Negatif
- Poin 3 = Netral
- Poin 4 = Positif
- Poin 5 = Sangat positif

Untuk mengolah data dari jawaban responden, peneliti menghitung mean dari setiap pertanyaan terlebih dahulu dengan cara menjumlahkan semua bobot atau poin dari semua responden yang pada akhirnya dibagi dengan jumlah

responden. Selanjutnya penghitungan grand mean yang didapatkan dari penjumlahan semua mean pada setiap pertanyaan dan dibagi jumlah pertanyaan tersebut yaitu delapan pertanyaan yang menghasilkan 3,775. Menurut Stemple Jr (2004) menyatakan bahwa nilai rata – rata sebesar 3,4 s.d 4,2 memiliki interpretasi “setuju atau penting” dalam data distribusi frekuensi. Oleh karena itu, pada tabel 3.6 tersebut mengindikasikan bahwa rata-rata responden setuju atau memiliki respon positif untuk setiap pertanyaan pada kuesioner tersebut. Serta standar deviasi populasi yang didapatkan dari tabel tersebut sebanyak 0,76. Standar deviasi tersebut termasuk moderat atau sedang yang menunjukkan bahwa walaupun setuju dengan mayoritas pertanyaan, namun responden juga memiliki jawaban yang cukup berbeda. Bahkan jawaban mayoritas responden ada yang lebih merespon negatif pada pertanyaan kuesioner. Hal tersebut menandakan bahwa responden tidak memiliki perspektif bias.

Sehingga dari analisis data kuesioner tersebut menunjukkan bahwa pembuatan *thumbnail* bagi konten kreator yang tidak memiliki latar belakang desain grafis adalah sebuah permasalahan yang terjadi. Pertanyaan ke-delapan pada kuesioner yang berbunyi “Apakah anda membutuhkan sebuah software untuk membantu membuat *thumbnail* secara otomatis?” juga sangat mendukung permasalahan tersebut yang dimana 40% responden menjawab “Butuh” serta 40% responden lainnya menjawab “Sangat Butuh” maka jawaban tersebut sangat mendukung bahwasanya konten kreator membutuhkan solusi dari permasalahan diatas.

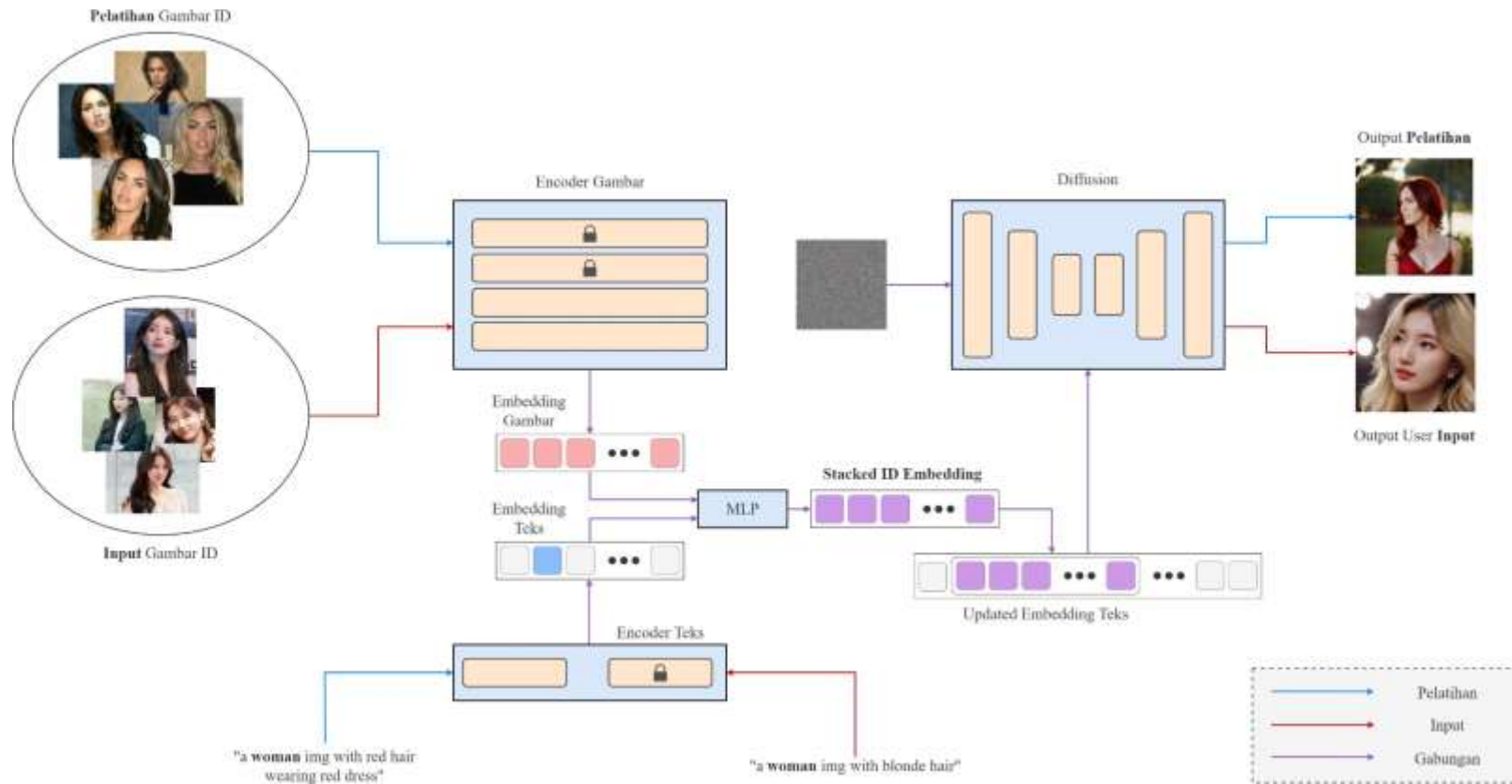
3.1.2 Pemecahan Masalah

Dengan membuat aplikasi ini yang menggunakan AI serta memanfaatkan penggunaan GPU, generasi gambar untuk *thumbnail* akan menjadi otomatis serta cepat. *User* hanya perlu memasukkan *prompt* sesuai yang diinginkan dan menunggu beberapa saat untuk gambar tersebut berhasil di generasi. Hal tersebut akan meningkatkan efektivitas *user* khususnya yang tidak memiliki pengalaman pada desain grafis dalam pembuatan *thumbnail*.

3.2 Perancangan

3.2.1 Alur Sistem

Alur pada Gambar 3.1 akan menjelaskan gambaran besar dari bagaimana cara kerja sistem untuk generasi gambar. Sehingga akan lebih mudah untuk dipahami sistem kerja dari aplikasi.

Gambar 3.1 Bagan *Image Generating*

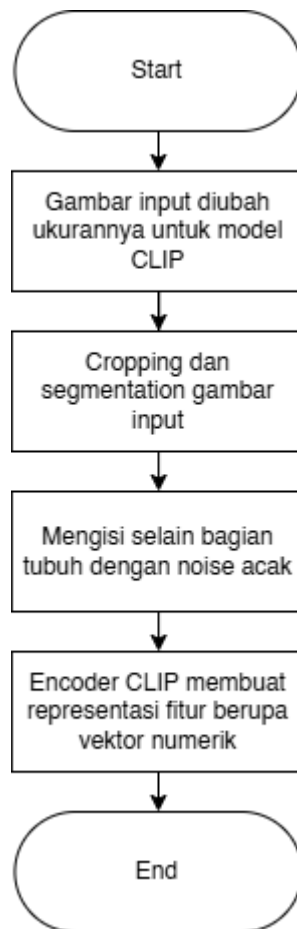
Pada alur yang ditampilkan, langkah awal dari proses melibatkan ekstraksi embedding gambar dan teks dari input maupun model pelatihan (pre-trained). Setelah memperoleh embedding untuk masing-masing, kedua embedding tersebut digabungkan menggunakan Multi-Layer Perceptron (MLP) untuk menghasilkan “*Stacked ID Embedding*”. Proses pembaruan Stacked ID Embedding merupakan tahap krusial yang bertujuan untuk meningkatkan akurasi gambar yang dihasilkan dan memastikan relevansi yang lebih tinggi (Li et al, 2023). Embedding yang telah diperbarui kemudian dimasukkan ke dalam model difusi, yang dimulai dari noise acak. Dalam proses ini, model difusi akan diperintahkan untuk “mengimajinasikan” gambar berdasarkan Stacked ID Embedding yang telah diperbarui, sehingga menghasilkan gambar yang diinginkan sesuai dengan embedding tersebut.

Untuk arti dari ikon kunci pada Gambar 3.2 adalah inputan tersebut tidak akan berubah selama proses. Seperti contoh untuk input teks selama proses tidak akan berubah dikarenakan jika ada perubahan dalam input teks maka hasil generasi tentunya tidak akan sesuai dengan ekspektasi user. Namun, ikon kunci pada pelatihan gambar menunjukkan bahwa beberapa parameter atau lapisan dalam encoder gambar tidak akan diubah selama proses pelatihan. Hal ini karena dataset tersebut digunakan untuk ekstraksi fitur yang stabil tanpa adanya pembaruan atau *fine-tuning*. Tujuannya adalah untuk menjaga agar fitur yang telah dipelajari tetap konsisten dan tidak terpengaruh oleh noise atau perubahan yang tidak relevan, sehingga memastikan bahwa karakteristik gambar ID dapat

diekstraksi dengan akurat dan stabil. Dengan kata lain, parameter yang dikunci membantu mempertahankan integritas dari fitur yang telah dipelajari dan mendukung kinerja model dalam menghasilkan representasi gambar yang berkualitas tinggi.

3.2.2 Flowchart Embedding Gambar

Berikut adalah *flowchart* yang menjelaskan secara rinci proses embedding gambar, menggambarkan langkah-langkah bagaimana informasi visual diolah menjadi representasi numerik dalam model.



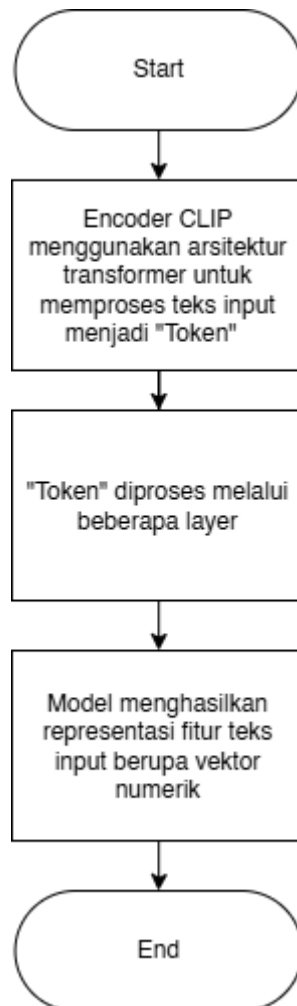
Gambar 3.2 *Flowchart Embedding Gambar*

Penelitian ini memanfaatkan Encoder CLIP sebagai encoder gambar, di mana proses dimulai dengan memasukkan gambar input ke dalam model. Gambar tersebut melalui tahap pre-processing yang mencakup resizing agar sesuai dengan spesifikasi model CLIP, sehingga meningkatkan efisiensi kerja model.

Selanjutnya, CLIP melakukan cropping dan segmentasi untuk fokus pada objek utama, mengurangi gangguan dari elemen lain seperti latar belakang dan objek yang tidak relevan. CLIP juga menambahkan noise acak pada area selain objek utama untuk meminimalisir gangguan dalam deteksi dan ekstraksi fitur. Setelah tahap *preprocessing* ini, CLIP menggunakan arsitektur Vision Transformer (ViT) untuk memproses gambar dalam patch, menghasilkan representasi fitur dalam bentuk vektor numerik yang dikenal sebagai “*Image Embedding*” yang merangkum informasi visual penting dari gambar tersebut.

3.2.3 Flowchart Embedding Teks

Flowchart berikut mengilustrasikan bagaimana sistem menghasilkan teks embedding yang kemudian dapat digabungkan dengan image embedding.



Gambar 3.3 *Flowchart Embedding Teks*

Pada penelitian ini, teks encoder juga menggunakan CLIP encoder. Proses untuk mendapatkan “*Text Embedding*” dimulai dengan penggunaan arsitektur transformer oleh CLIP untuk mengubah teks input menjadi unit-unit token. Setelah token terbentuk, token-token ini diproses melalui beberapa lapisan

transformer untuk menangkap konteks dan hubungan antar kata, yang penting untuk memahami makna teks secara keseluruhan. Proses ini memastikan bahwa model dapat memahami arti teks dengan lebih baik, sehingga hasil generasi menjadi lebih relevan. Akhirnya, CLIP mengubah teks yang telah diproses menjadi representasi fitur unik dalam bentuk vektor numerik, yang kemudian dapat digunakan dalam proses selanjutnya.

3.2.4 Flowchart MLP

Flowchart berikut menggambarkan proses penggabungan embedding gambar dan teks untuk menghasilkan “*Stacked ID Embedding*”.



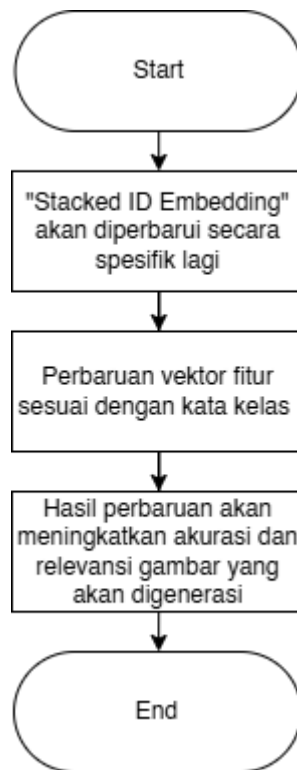
Gambar 3.4 Use Case Diagram MLP

Embedding gambar dan teks yang diperoleh dari proses sebelumnya digabungkan menggunakan *Multi-Layer Perceptron* (MLP). MLP melakukan operasi fusi dengan menumpuk lapisan-lapisan embedding dari gambar dan teks untuk menghasilkan representasi yang terpadu. Proses ini memungkinkan MLP untuk mengevaluasi kesamaan antara embedding teks dan gambar, sehingga meminimalisir kesalahan pasangan yang tidak tepat sehingga meningkatkan

akurasi. Hasil dari proses ini adalah “*Stacked ID Embedding*” yaitu representasi yang lebih kompleks dan kaya akan karakteristik sesuai dengan input.

3.2.5 Flowchart Pembaruan Stacked ID Embedding

Salah satu langkah atau proses krusial pada seluruh sistem ini adalah pembaruan dari “*Stacked ID Embedding*” yang telah didapatkan dari proses MLP. Proses tersebut dapat dilihat lebih jelas melalui ilustrasi flowchart berikut.



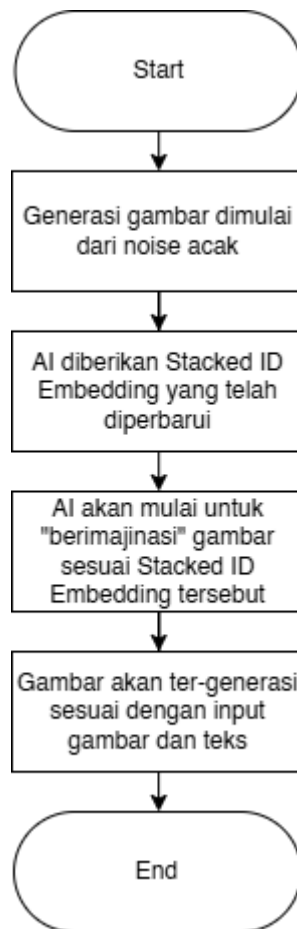
Gambar 3.5 *Flowchart Pembaruan Stacked ID Embedding*

Stacked ID Embedding yang diperoleh dari proses MLP kemudian diperbarui dalam embedding teks awal. Secara khusus, vektor fitur pada posisi yang sesuai dengan kata kelas (misalnya, “*man*” atau “*woman*”) dalam embedding teks asli digantikan dengan *Stacked ID Embedding*. Pembaruan ini bertujuan untuk meningkatkan kesesuaian gambar yang dihasilkan dengan ekspektasi pengguna, karena memungkinkan integrasi fitur identitas khusus dari *Stacked ID*

Embedding ke dalam embedding teks. Dengan cara ini, hasil generasi gambar menjadi lebih relevan dan akurat sesuai dengan deskripsi teks dan karakteristik identitas yang diinginkan. Ilustrasi dibawah ini menggambarkan proses dari langkah tersebut.

3.2.6 Flowchart Diffusion Model

Untuk langkah terakhir dalam generasi gambar yaitu melalui model difusi yang dimana akan menghasilkan gambar sesuai dengan “*Stacked ID Embedding*” yang telah diperbarui.



Gambar 3.6 *Flowchart Diffusion Model*

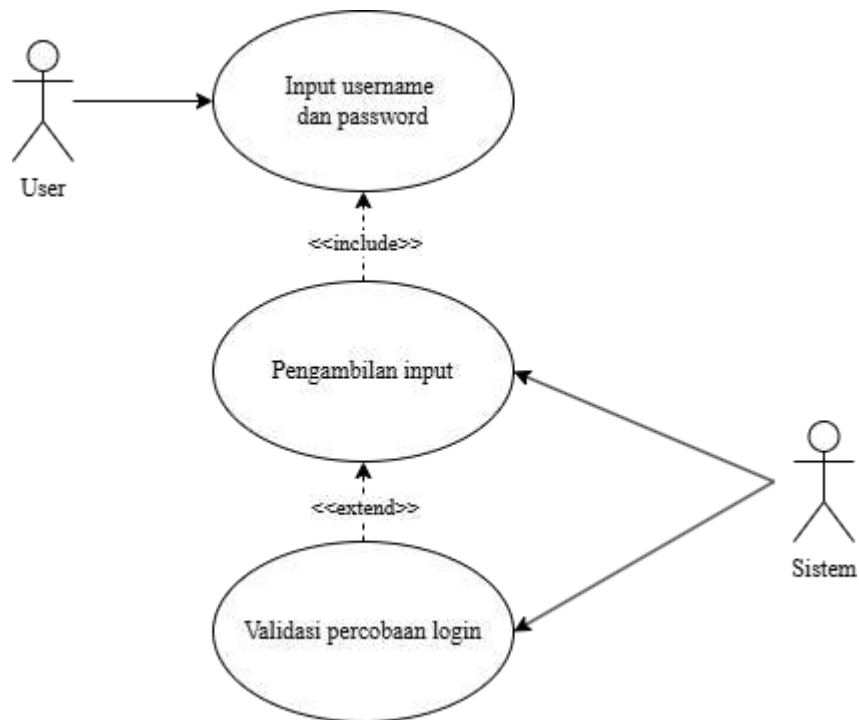
Diffusion model, sebelum dapat menghasilkan foto, memerlukan pelatihan menggunakan dataset khusus. Selama pelatihan, model ini secara bertahap mengubah gambar input menjadi noise sederhana melalui serangkaian langkah. Setelah proses transformasi ini, model diharuskan untuk merekonstruksi gambar

dari noise tersebut sehingga integrasi informasi gambar tetap terjaga. Proses ini berlanjut hingga gambar sepenuhnya menjadi noise. Model kemudian menggunakan vektor Stacked ID Embedding yang telah diperbarui untuk ‘mengimajinasikan’ dan menghasilkan gambar baru yang sesuai dengan input gambar dan teks dari pengguna. Proses ini melibatkan mekanisme cross-attention untuk menggabungkan informasi ID dari Stacked ID Embedding dengan embedding teks, memastikan bahwa gambar yang dihasilkan sesuai dengan identitas yang ditentukan dan deskripsi teks yang diberikan.

3.2.7 Use Case Diagram Login

Login adalah salah satu fitur dari PotionPix yang berfungsi untuk memisahkan profil *user* walaupun menggunakan dalam satu perangkat yang sama.

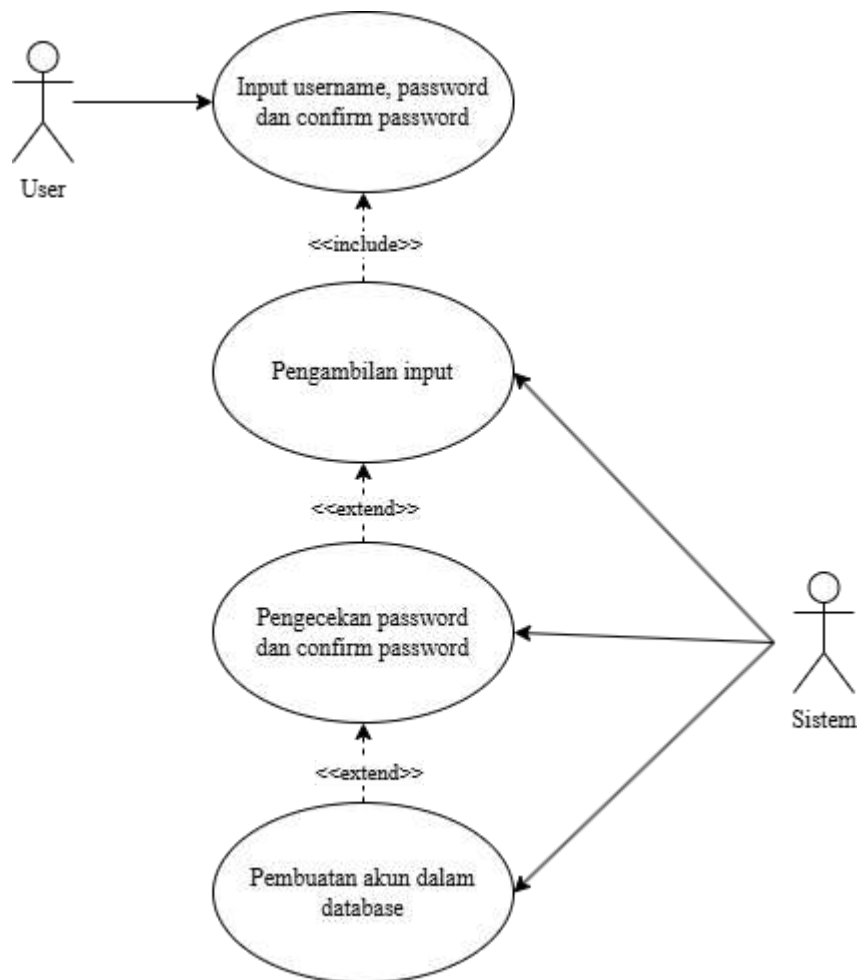
Gambar 3.7 menjelaskan *use case* dari fitur *login* tersebut.



Gambar 3.7 Use Case Diagram Login

3.2.8 Use Case Diagram Sign Up

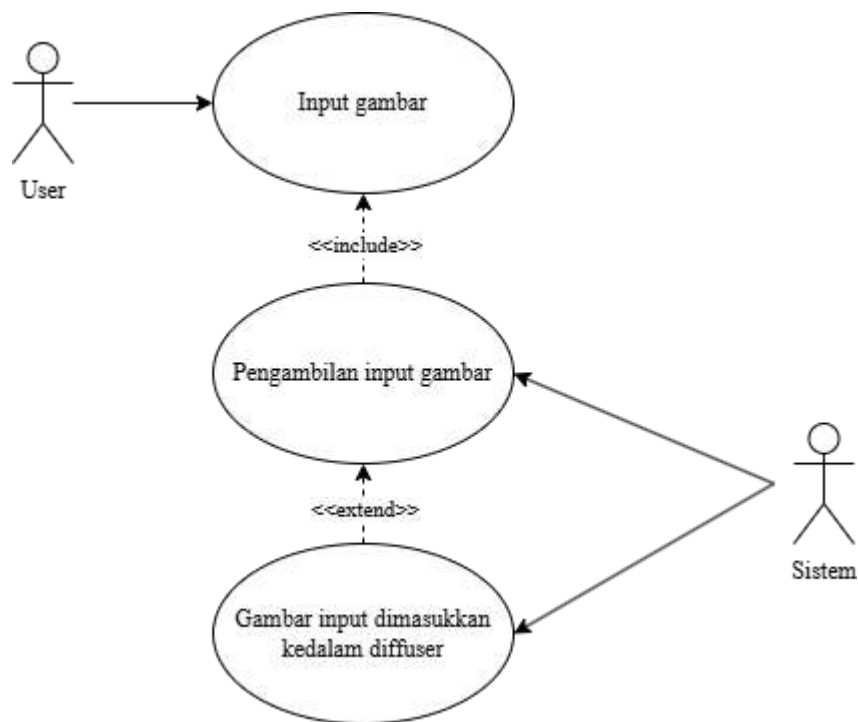
Sign up memiliki fungsi untuk *user* membuat sebuah akun baru. Sehingga *user* dapat melanjutkan proses dari *Login*. Penjelasan *use case* dari fitur ini dapat dilihat pada Gambar 3.8.



Gambar 3.8 Use Case Diagram *Sign Up*

3.2.9 Use Case Diagram Select Image

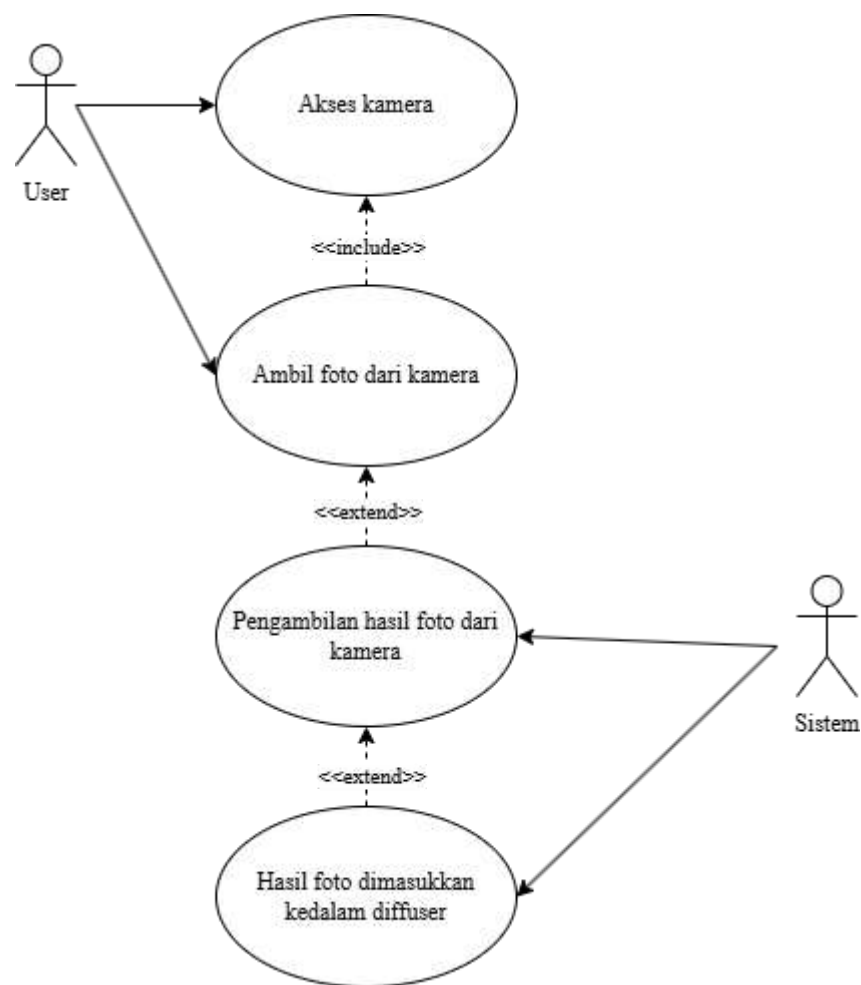
Pemilihan gambar pada fitur ini berfungsi sebagai *input* gambar yang akan dimasukkan kedalam *diffuser*. Sehingga *diffuser* akan dapat memulai proses *encoding* gambar dan mengekstrak fitur unik dari gambar tersebut. *Use case* dapat ditinjau pada Gambar 3.9.



Gambar 3.9 Use Case Diagram Select Image

3.2.10 Use Case Diagram Kamera

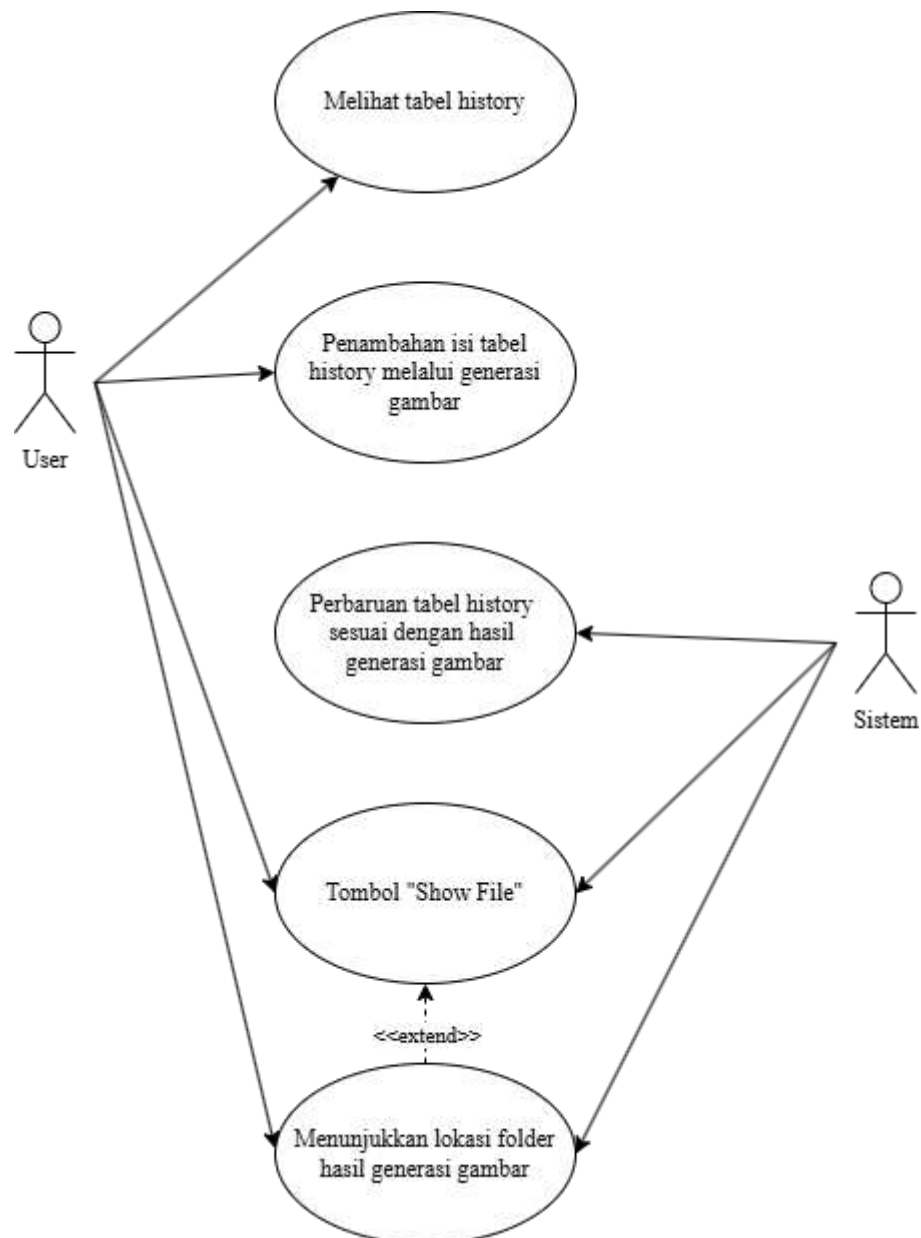
Fitur ini memanfaatkan perangkat kamera yang terhubung pada perangkat yang digunakan oleh *user*. Kamera tersebut dapat mengambil foto yang akan disimpan dalam sebuah folder. Sehingga hasil foto tersebut dapat dijadikan sebagai *input* gambar kedalam *diffuser*. Untuk *use case* diagram nya dapat dilihat pada Gambar 3.10.



Gambar 3.10 Use Case Diagram Kamera

3.2.11 Use Case Diagram History

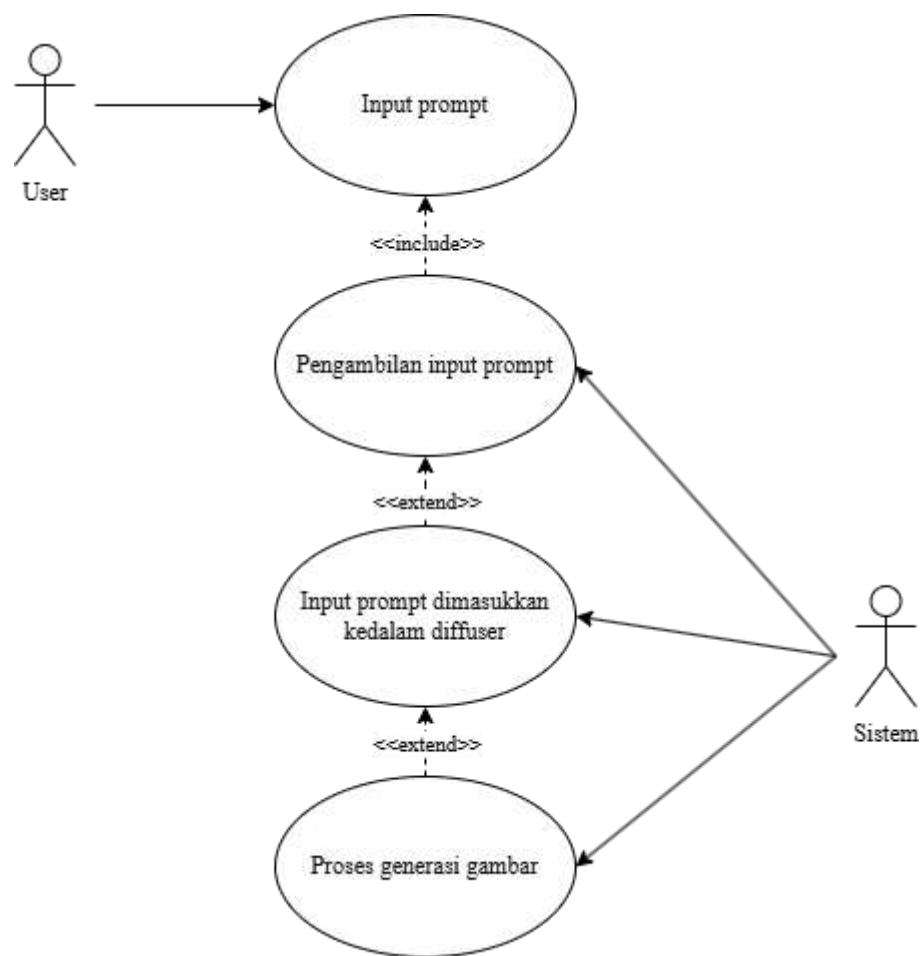
Pada halaman *history*, *user* dapat melihat sebuah tabel yang menunjukkan informasi sejarah generasi yang telah mereka lakukan. Dari *prompt* yang telah dimasukkan, kategori yang dipilih, hingga *path* hasil gambar tersebut ada pada tabel *history*. Gambar 3.11 menjelaskan *use case* dari halaman *history* tersebut.



Gambar 3.11 Use Case Diagram History

3.2.12 Use Case Diagram Prompt

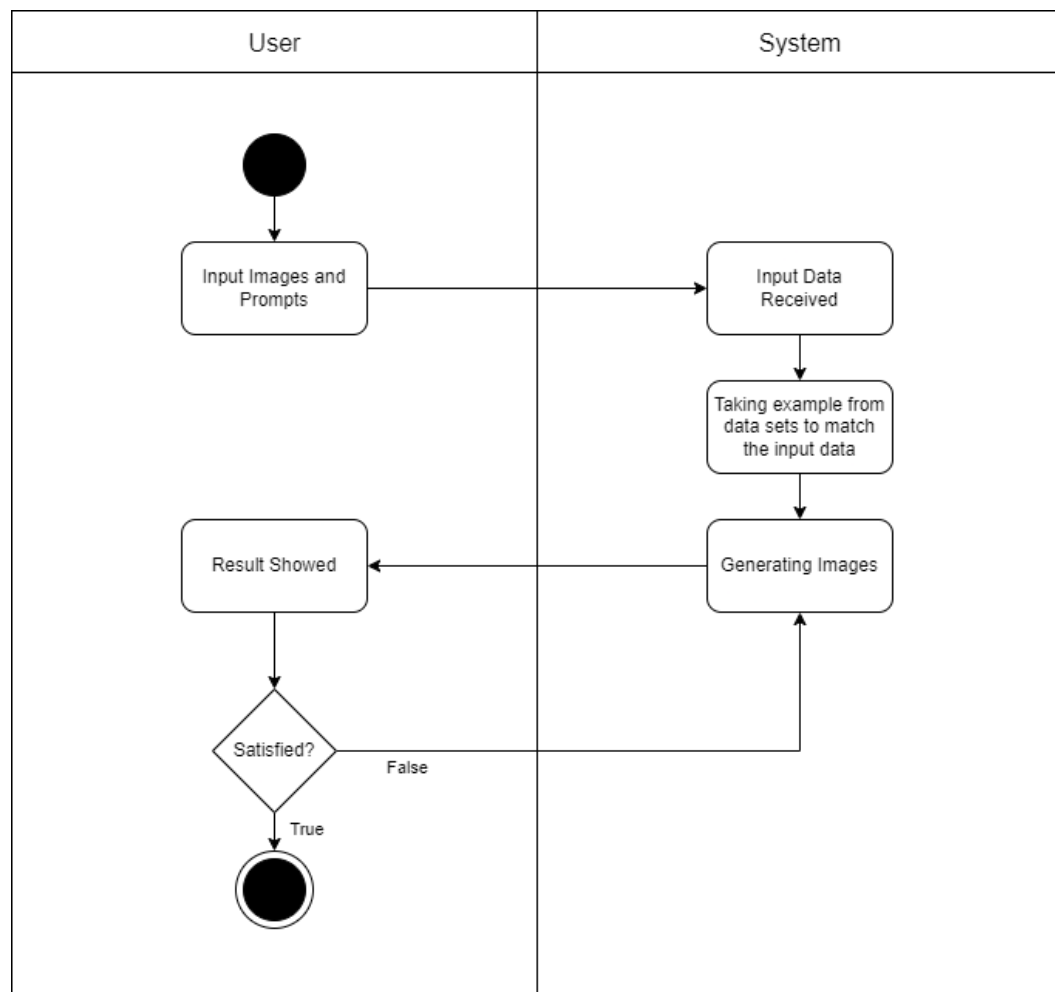
User pada halaman *prompt* akan memiliki akses berupa *input* pada *field* teks yang sudah disediakan, serta memilih pilihan yang telah disediakan. Hasil dari *input* dan pilihan *user* tersebut akan otomatis menjadi satu kalimat yang akan dimasukkan kedalam diffuser. *Use case* diagram untuk halaman *prompt* ini dapat dilihat pada Gambar 3.12.



Gambar 3.12 *Use Case Diagram Prompt*

3.2.14 Activity Diagram

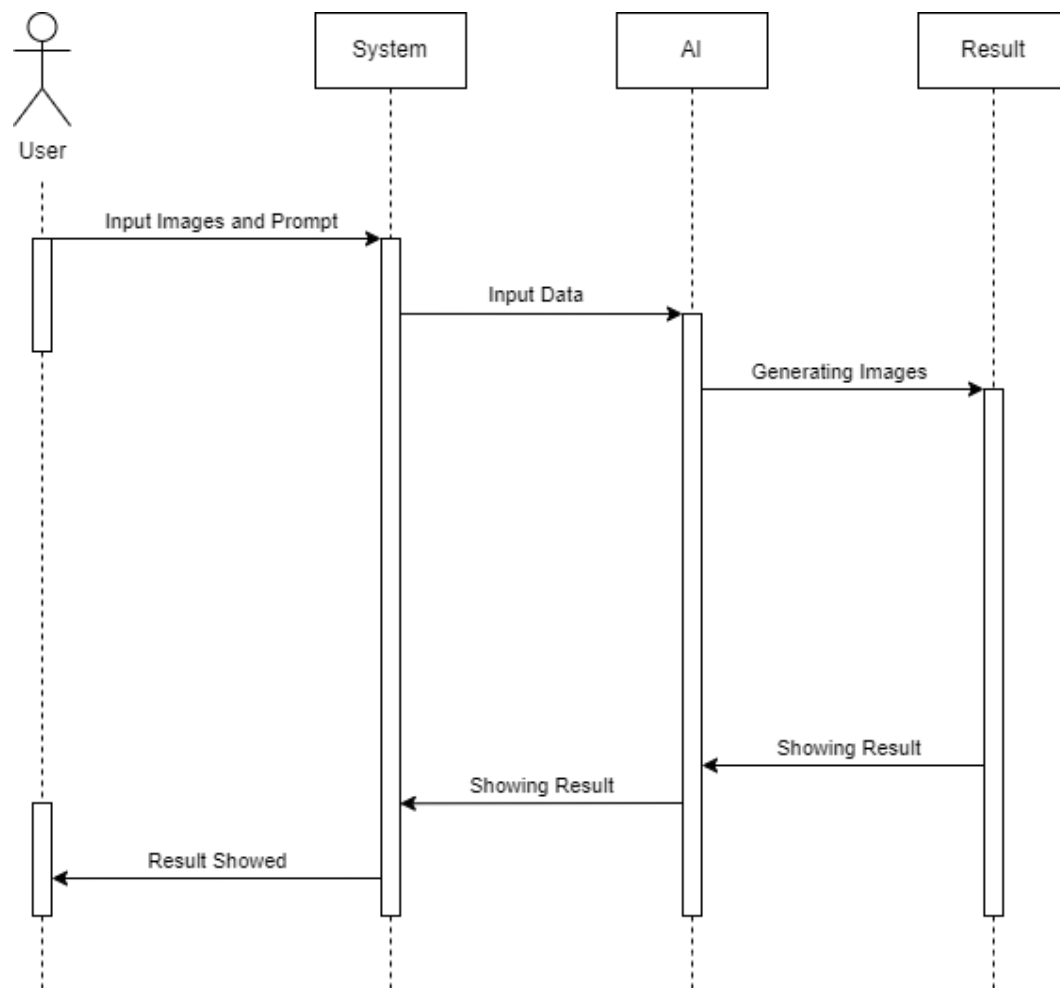
Activity diagram dibawah ini menjelaskan lebih detail dari use case diagram. Bagaimana aktivitas dari sebuah proses dan aktor dari use case diagram sebelumnya dijelaskan. Hal tersebut dapat dilihat pada Gambar 3.14.



Gambar 3.14 Activity Diagram *Image Generating*

3.2.15 Sequential Diagram

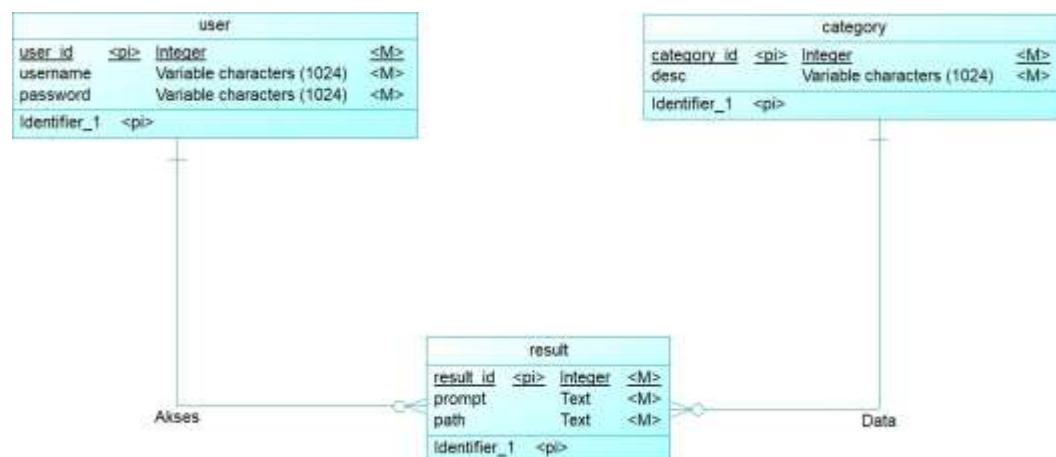
Sequential diagram dibawah ini memiliki tujuan untuk menjelaskan bagaimana urutan yang terjadi untuk mendapatkan hasil yang diperoleh. Berikut pada Gambar 3.15 digambarkan.



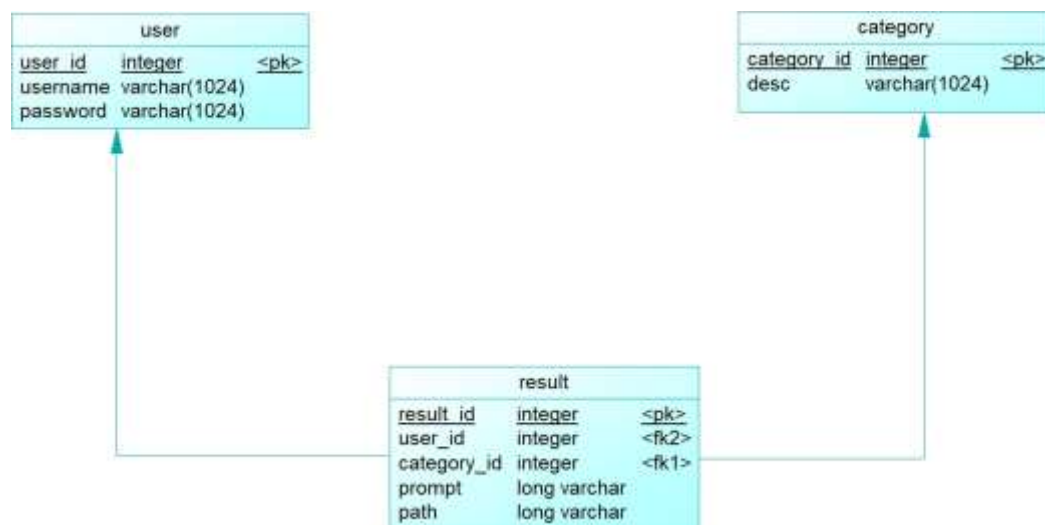
Gambar 3.15 *Sequential Diagram Image Generating*

3.2.16 Entity Relationship Diagram

Diagram yang ditampilkan pada Gambar 3.16 dan Gambar 3.17 menggambarkan struktur basis data yang digunakan dalam aplikasi atau sistem PotionPix. Basis data ini terdiri dari tiga tabel, yaitu tabel pengguna (*user*), tabel kategori (*category*), dan tabel hasil (*result*). Tabel pengguna berfungsi untuk menyimpan kredensial *login* setiap pengguna, sementara tabel kategori menyimpan informasi mengenai kategori *platform* yang digunakan untuk generasi gambar. Tabel hasil, di sisi lain, berperan dalam menyimpan informasi terkait hasil generasi gambar, yang mencakup *user_id* dan *category_id* sebagai *foreign key*, *prompt* yang dimasukkan oleh pengguna, serta *path* yang menunjukkan lokasi *folder* hasil generasi gambar.



Gambar 3.16 CDM Database PotionPix

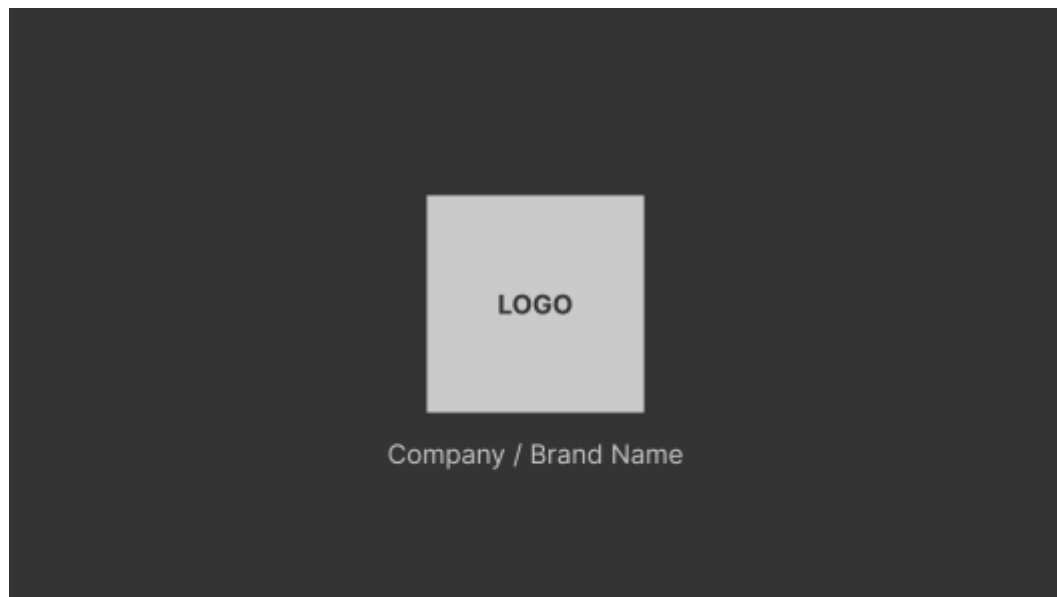


Gambar 3.17 PDM Database PotionPix

3.3 Rancangan Antarmuka

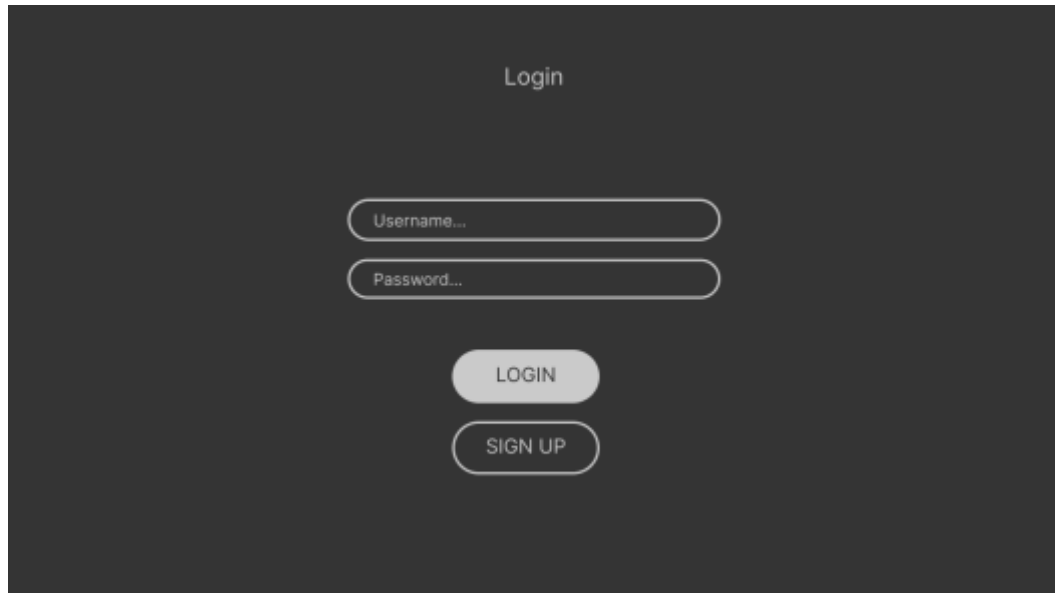
Desain rancangan antarmuka adalah salah satu hal yang penting untuk membuat suatu program atau aplikasi. Berikut ini adalah *prototype* rancangan antarmuka *low fidelity* yang dapat dilihat pada Gambar 3.18 hingga Gambar 3.27 dibawah ini.

Gambar 3.18 adalah UI pertama yang akan diperlihatkan saat menjalankan aplikasi. Splash screen tersebut akan menampilkan nama dari aplikasi serta *icon* atau *logo* nya.



Gambar 3.18 Desain Antarmuka *Splash Screen*

Setelah halaman *splash screen*, halaman *login* yang dapat dilihat pada Gambar 3.19 yang akan ditampilkan. Halaman tersebut memiliki fungsi berupa verifikasi akun dari pengguna yang telah tersimpan pada database.



The image shows a dark-themed login screen. At the top center, the word "Login" is displayed in a light gray font. Below it, there are two rounded rectangular input fields. The first field is labeled "Username..." and the second field is labeled "Password...". Both labels are in a light gray font. Below the input fields, there are two rounded rectangular buttons. The top button is labeled "LOGIN" and the bottom button is labeled "SIGN UP". Both buttons are in a light gray font. The entire screen has a dark gray background.

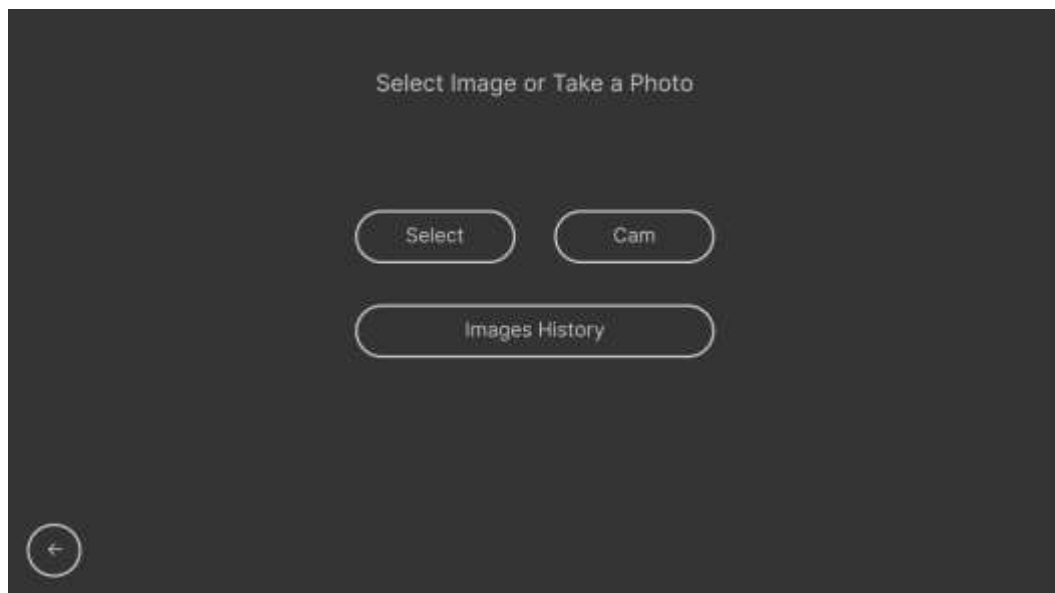
Gambar 3.19 Desain Antarmuka *Login*

Sebelum pengguna dapat mencoba untuk *login*, tentunya mereka diharuskan untuk mendaftarkan akun terlebih dahulu kedalam database. Sehingga setelah akun telah terdaftar, maka pengguna dapat mencoba untuk *login*. Untuk desain antarmuka halaman pendaftaran akun dapat dilihat pada Gambar 3.20.

The image shows a dark-themed user interface for a 'Sign Up' page. At the top center, the text 'Sign Up' is displayed in a light gray font. Below this, there are three vertically stacked input fields, each with a light gray border and rounded corners. The first field is labeled 'Username...', the second 'Password...', and the third 'Confirm Password...'. Below these fields is a rounded rectangular button with the text 'SIGN UP' in all caps. In the bottom left corner, there is a small circular icon containing a left-pointing arrow.

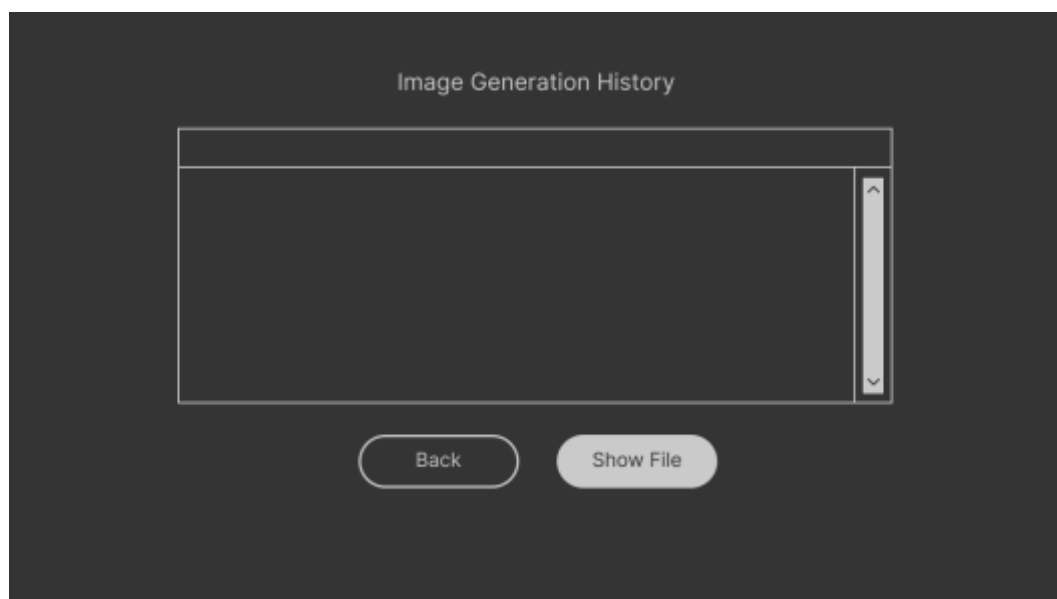
Gambar 3.20 Desain Antarmuka *Sign Up*

Jika pengguna telah berhasil dalam melakukan percobaan *login*, mereka akan langsung diarahkan menuju halaman *menu* yang dapat ditinjau pada Gambar 3.21. Pada halaman ini akan disediakan 3 macam pilihan tombol berupa, tombol untuk memilih gambar yang sudah tersimpan atau ada. Tombol untuk mengambil gambar menggunakan perangkat kamera. Serta tombol untuk melihat hasil generasi gambar pada sebelumnya jika pengguna pernah melakukan generasi gambar.



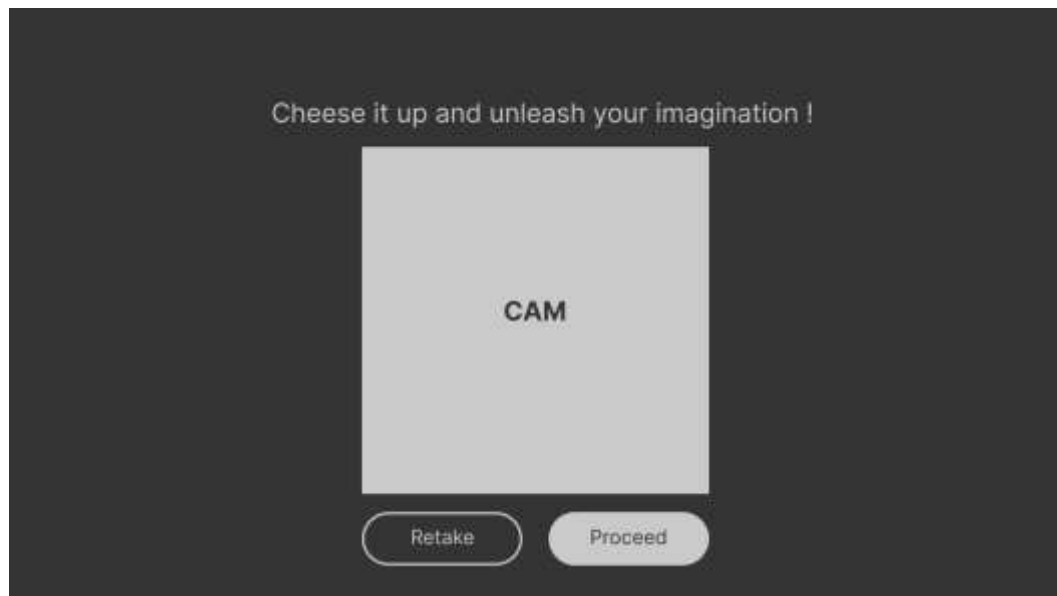
Gambar 3.21 Desain Antarmuka *Menu*

Pada halaman *history*, pengguna akan disajikan informasi berupa hasil generasi gambar pada sebelum-sebelumnya. Pada tabel tersebut akan ditampilkan prompt apa yang telah pengguna ketik serta hasil gambarnya. Sehingga pengguna dapat memanfaatkan fitur tersebut untuk referensi pada generasi gambar kedepannya. Tentunya untuk desain antarmuka halaman *history* dapat dilihat pada Gambar 3.22.



Gambar 3.22 Desain Antarmuka *History*

Pada Gambar 3.23 akan menampilkan halaman pengguna untuk mengambil gambar menggunakan perangkat kamera. Hal tersebut akan mempermudah pengguna jika dia tidak memiliki gambar yang sudah tersedia untuk dijadikannya *input* gambar.



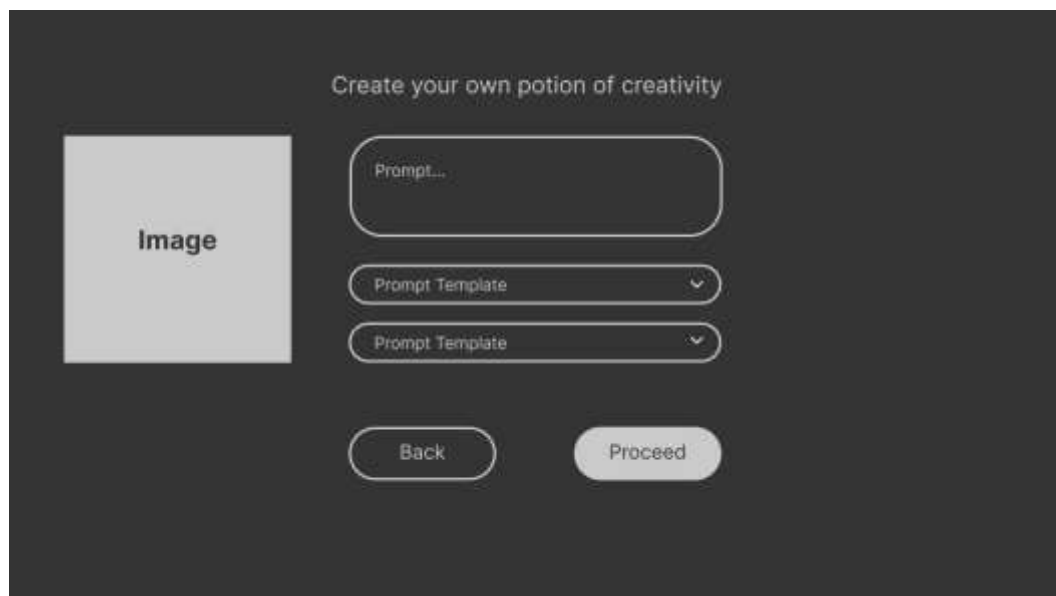
Gambar 3.23 Desain Antarmuka *Input Camera*

Sedangkan pilihan lainnya untuk *input* gambar adalah pemilihan *file* yang pengguna sudah siapkan. Pengguna dapat memilih gambar dengan ekstensi jpg, jpeg, serta png. Dengan adanya pilihan halaman tersebut, pengguna akan memiliki kebebasan untuk memilih diantara memanfaatkan kamera ataupun menyediakan gambar sebagai *input*. Gambar 3.24 akan memperlihatkan rancangan desain antarmuka untuk *input* gambar.



Gambar 3.24 Desain Antarmuka *Input Image*

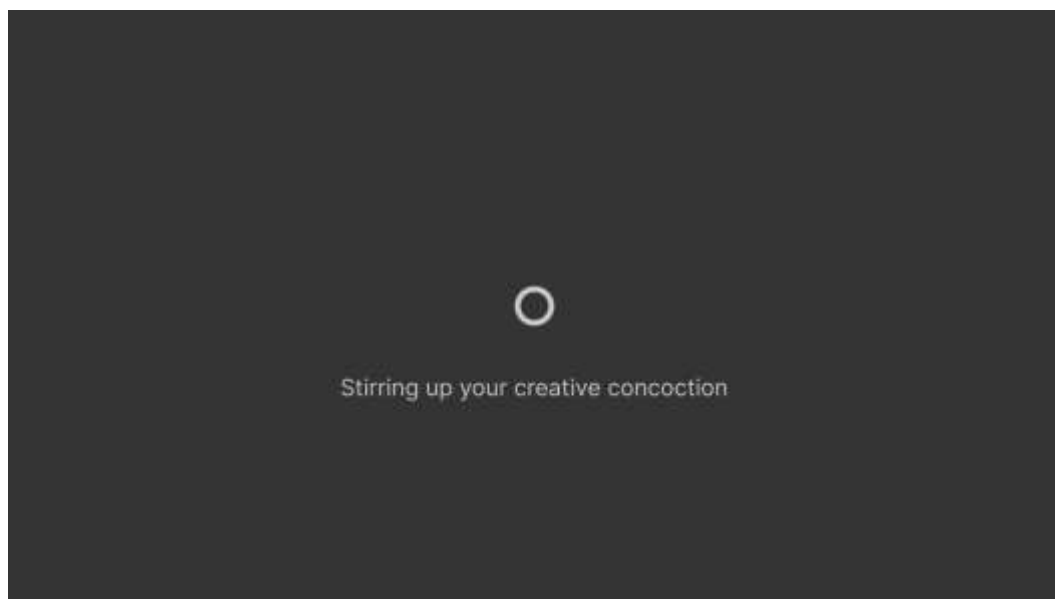
Setelah pengguna *input* sebuah gambar, maka mereka dapat melanjutkan menuju halaman selanjutnya yang dapat dilihat pada Gambar 3.25. Pada halaman tersebut, pengguna dapat memasukkan input teks sesuai dengan kreatifitas yang diinginkan. Tentunya ada beberapa *prompt* yang dimana pengguna disediakan untuk memilih beberapa opsi. Hal tersebut dikarenakan faktor hasil kualitas gambar salah satunya dipengaruhi oleh kualitas *prompt* itu sendiri. Sehingga dengan adanya pilihan tersebut, kualitas *input* teks dari pengguna akan lebih konsisten dan tinggi dibandingkan jika pengguna mengetikkan tanpa adanya struktur.



The image shows a dark-themed user interface for creating prompts. At the top, the text "Create your own potion of creativity" is displayed. On the left, there is a square placeholder labeled "Image". To the right of the image placeholder, there is a large, rounded rectangular text input field labeled "Prompt...". Below this field are two identical dropdown menus, each labeled "Prompt Template" with a downward arrow icon. At the bottom of the interface, there are two rounded rectangular buttons: "Back" on the left and "Proceed" on the right.

Gambar 3.25 Desain Antarmuka *Prompt*

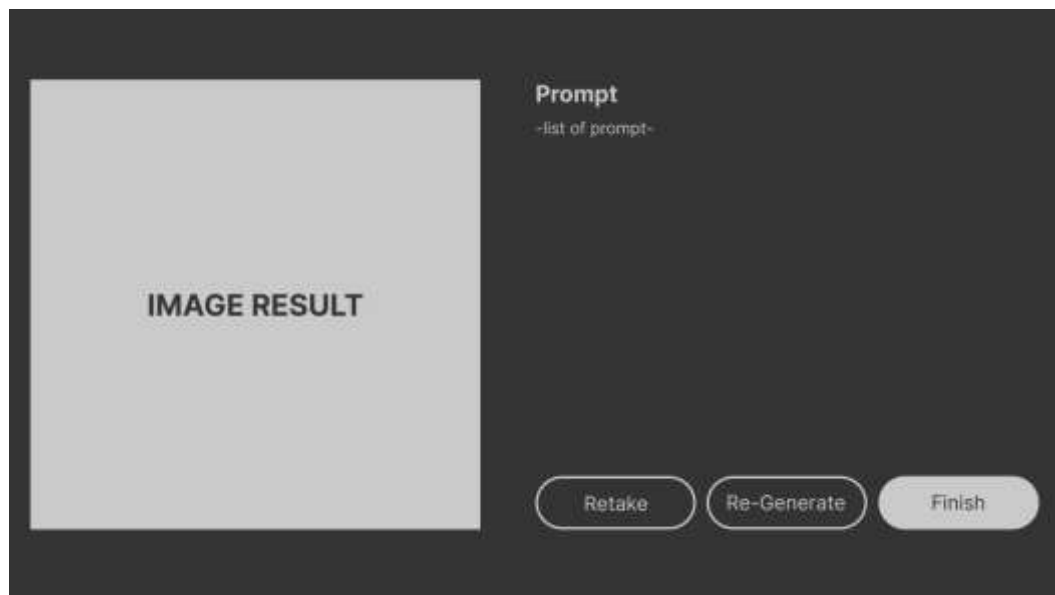
Tentunya disaat pengguna memulai proses generasi gambar, aplikasi akan menampilkan sebuah *loading screen*. Hal tersebut memiliki tujuan untuk berkomunikasi kepada pengguna bahwa hasil gambar masih dalam tahap pemrosesan. Untuk rancangan desain *loading screen* tersebut dapat dilihat pada Gambar 3.26.



Gambar 3.26 Desain Antarmuka *Loading Screen*

Saat proses dari generasi gambar sudah selesai, maka pengguna akan diarahkan pada halaman hasil atau rekap dari pilihan dan *input* pengguna. Pada halaman tersebut akan ditampilkan hasil generasi gambar serta *prompt* yang mereka ketik dan pilih. Selain itu, halaman tersebut akan menyajikan tombol untuk melakukan generasi gambar ulang yang dimana semua pilihan dan masukan yang sama. Sehingga jika pengguna tidak puas dengan hasil dari generasi gambar

pertama, mereka selalu bisa mengulangi generasi gambar tanpa memerlukan untuk proses *input* dari awal. Lalu untuk tombol lainnya memiliki fungsi untuk kembali kepada *menu* awal serta keluar dari aplikasi. Rancangan desain halaman tersebut dikaji pada Gambar 3.27.



Gambar 3.27 Desain Antarmuka Hasil *Image Generation*

3.4 Rancangan Pengujian

Metode pengujian yang akan digunakan pada aplikasi ini adalah metode pengujian Blackbox. Pengujian Blackbox merupakan bentuk pengujian aplikasi secara keseluruhan untuk melihat apakah berfungsi dengan baik dan sesuai rencana.