

BAB III

ANALISIS DAN PERANCANGAN

3.1 Analisis Sistem (*System Analysis*)

Analisis sistem merupakan proses penguraian suatu sistem secara menyeluruh ke dalam bagian-bagian komponennya dengan tujuan untuk mengidentifikasi serta mengevaluasi berbagai permasalahan atau kendala yang terdapat di dalam sistem. Melalui tahapan ini, diharapkan dapat dirumuskan langkah-langkah penanggulangan, perbaikan, serta pengembangan sistem secara lebih terarah dan efektif (Mulyono, 2020).

3.1.1 Analisis Masalah

Analisis masalah adalah proses untuk mengidentifikasi, memahami, dan menganalisis sebuah masalah atau isu untuk menemukan akar penyebabnya dan mengusulkan solusi yang tepat. Ini adalah proses sistematis yang melibatkan langkah-langkah seperti mengidentifikasi masalah, mengumpulkan data, menganalisis penyebab, dan mengembangkan rencana tindakan untuk mengatasi masalah tersebut.

Berdasarkan hasil wawancara dengan Bapak Suparno selaku pemilik Toko Ikan & Aquarium Cilimus, ditemukan permasalahan bahwa pembeli mengalami kesulitan dalam membedakan jenis ikan cupang. Kesulitan ini disebabkan oleh keterbatasan informasi yang diperoleh pembeli, yang hanya disampaikan secara lisan oleh pemilik. Selain itu, proses membedakan jenis ikan cupang masih dilakukan secara manual, yang berpotensi menghambat efisiensi serta kualitas layanan pemilik, terutama ketika menghadapi jumlah pembeli yang banyak atau permintaan spesifik.

3.1.2 Analisis Kebutuhan Fungsional

Kebutuhan fungsional merupakan kebutuhan akan fasilitas yang dibutuhkan oleh sistem, termasuk aktivitas-aktivitas yang dijalankan secara umum dalam mendukung kinerja sistem. Kebutuhan ini mencakup proses-

proses yang akan dilaksanakan oleh sistem serta jenis informasi yang dibutuhkan maupun yang harus dihasilkan sebagai *output* dari sistem tersebut (Setiyani & Tjandra, 2021). Kebutuhan fungsional yang harus dipenuhi oleh aplikasi klasifikasi jenis ikan cupang dapat diuraikan sebagai

Perangkat Keras	Spesifikasi yang Dipakai
RAM	Intel Core i3-2328M Processor
Storage	418 GB
RAM	6 GB

berikut:

1. Aplikasi dapat menampilkan halaman *Login*, *Register*, Beranda, Pindai, Profil, tutorial aplikasi, dan Tentang Aplikasi.
2. Aplikasi dapat membuka kamera dan galeri untuk *input* gambar jenis ikan cupang.
3. Aplikasi dapat mengklasifikasikan ikan cupang jenis *Halfmoon*, *Plakat*, dan *Serit*.
4. Aplikasi dapat menampilkan informasi hasil pindai jenis ikan cupang.

3.1.3 Analisis Kebutuhan Non-Fungsional

Kebutuhan non-fungsional merupakan kebutuhan yang mendeskripsikan bagaimana sistem akan berjalan atau beroperasi di masa mendatang. Penentuan kebutuhan non-fungsional memerlukan pemahaman yang mendalam terhadap karakteristik serta batasan dari sistem yang dikembangkan. Kebutuhan ini dapat dianggap sebagai standar kualitas dan kinerja yang harus dicapai oleh perangkat lunak, sehingga menjadi elemen penting yang perlu diperhatikan secara serius selama proses perancangan dan pengembangan sistem (Aulia Aziiza & Nur Fadhilah, 2020). Kebutuhan non-fungsional pada aplikasi ini mencakup spesifikasi perangkat keras dan perangkat lunak yang dibutuhkan, serta karakteristik pengguna yang akan mengoperasikan aplikasi.

1. Kebutuhan Perangkat Keras

Spesifikasi perangkat keras yang dimanfaatkan oleh peneliti dalam pelaksanaan penelitian ini dapat dijelaskan sebagai berikut:

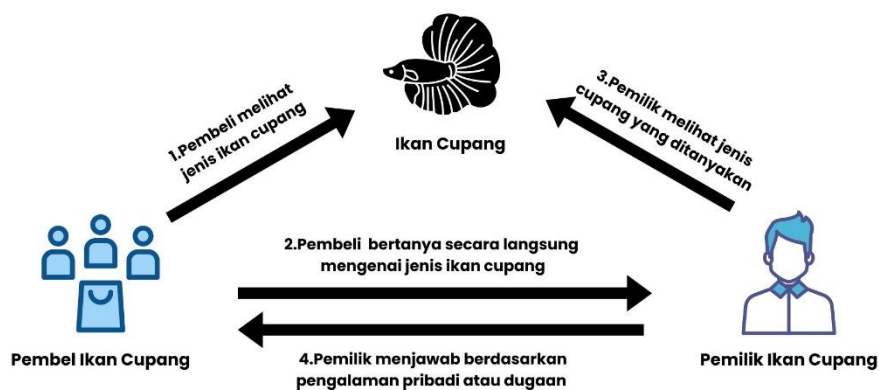
2. Kebutuhan Perangkat Lunak

Adapun spesifikasi perangkat lunak yang digunakan oleh peneliti dalam proses penelitian ini adalah sebagai berikut:

- a. *Operating System*: Windows 10 Pro
- b. *Development Tools*: Anaconda (Jupyter Notebook), Android Studio
- c. *System Design Tools*: Draw.IO
- d. *UI Design Tools*: Figma

3.1.4 Analisis Sistem yang Sedang Berjalan

Pada tahap ini dilakukan studi dan evaluasi terhadap sistem yang saat ini diterapkan di Toko Ikan & Aquarium Cilimus. Analisis terhadap sistem yang berjalan dilakukan dengan meneliti berbagai aspek yang berkaitan dengan proses pembelian di toko tersebut. Adapun hasil dari analisis sistem yang berjalan dapat dilihat pada *Rich Picture* yang disajikan pada Gambar 3.1.



Gambar 3. 1 sistem yang sedang berjalan

Pada gambar diatas, sistem yang sedang berjalan di Toko Ikan & Aquarium Cilimus sebagai berikut:

1. Pertama, calon pembeli datang dan melihat koleksi ikan cupang yang tersedia di Toko Ikan & Aquarium Cilimus.
2. Pembeli kemudian menanyakan jenis ikan cupang, seperti *Halfmoon*, *Plakat*, atau *Serit*, kepada pemilik karena kesulitan dalam membedakannya hanya berdasarkan tampilan fisik secara umum.
3. Pemilik memberikan informasi berdasarkan pengetahuan, pengalaman pribadi, atau dugaan, tanpa adanya sistem pendukung keputusan atau alat bantu identifikasi visual yang dapat memastikan jenis ikan secara akurat.
4. Setelah itu, terdapat dua kemungkinan. Pertama, pemilik berhasil mengidentifikasi jenis ikan dengan benar, sehingga ikan yang dibeli sesuai dengan yang diinginkan pembeli dan tidak menimbulkan permasalahan. Kedua, terjadi kesalahan identifikasi atau pelabelan jenis ikan (baik karena kekeliruan pemilik maupun dari pihak pemasok), yang menyebabkan pembeli memperoleh ikan cupang dengan jenis yang tidak sesuai.

3.1.5 Analisis Sistem Usulan

Dengan sistem yang diusulkan ini diharapkan dapat membantu pemilik dan pembeli di Toko Ikan & Aquarium Cilimus dalam membedakan jenis ikan cupang dengan lebih efektif. Berikut merupakan analisis terhadap sistem yang diusulkan.



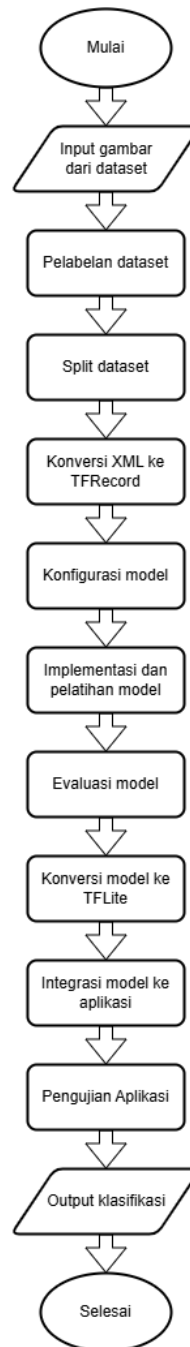
Gambar 3. 2 sistem usulan

Gambar di atas merupakan *Rich Picture* yang menggambarkan alur interaksi antara pemilik dan pembeli dengan sistem aplikasi klasifikasi ikan cupang berbasis citra digital. Proses dimulai ketika pengguna membuka aplikasi, yang kemudian menampilkan halaman *Login*. Apabila pemilik dan pembeli sudah memiliki akun, maka dapat langsung melakukan *Login* dengan memasukkan *username* dan *password*. Namun, jika belum terdaftar, pengguna diarahkan untuk melakukan proses registrasi terlebih dahulu dengan mengisi informasi yang diperlukan. Setelah berhasil *Login*, akan diarahkan ke halaman beranda aplikasi. Pada halaman ini, dapat memilih fitur pemindaian (*scan*) gambar ikan cupang. Setelah pemilik dan pembeli memilih menu *scan*, maka sistem akan membuka fitur pengambilan atau unggah gambar. Gambar tersebut kemudian diproses oleh sistem menggunakan model klasifikasi berbasis *Deep Learning*, dan hasil dari klasifikasi tersebut kemudian akan ditampilkan kepada pemilik dan pembeli dalam bentuk *output* yang menginformasikan jenis ikan cupang sesuai dengan citra yang telah dianalisis. Alur ini menggambarkan proses utama yang terjadi dalam aplikasi, yang terdiri dari proses autentikasi pengguna hingga ke tahap klasifikasi gambar ikan cupang.

Untuk mengatasi permasalahan yang ada, peneliti peneliti membangun aplikasi menggunakan algoritma *CNN* dengan menggunakan *framework TensorFlow*.

3.1.6 Analisis Penyelesaian Masalah

Metode yang digunakan untuk menyelesaikan masalah dalam penelitian ini adalah menerapkan algoritma *CNN* dengan pendekatan *Transfer Learning*. Dalam penelitian ini terdapat beberapa tahapan alur rancangan program sebagai berikut:



Gambar 3. 3 Alur metode penyelesaian masalah

a. Pengumpulan dataset

Penelitian ini diawali dengan tahap pengumpulan dataset yang digunakan untuk membangun sebuah model. Dataset tersebut terdiri dari gambar jenis ikan cupang *Halfmoon*, *Plakat*, dan *Serit* dengan per kelas mencakup 1033 gambar.

b. Pelabelan dataset

Dataset yang telah dikumpulkan, diunggah ke platform *RoboFlow* untuk pemberian anotasi atau label pada setiap gambar dalam dataset.

c. Split dataset

Dataset yang telah diberi label, akan dipisahkan secara acak dengan proporsi *training* 80%, *validation* 10%, dan *testing* 10%.

d. Konversi XML ke TFRecord

Melakukan konversi anotasi dari file XML ke TFRecord untuk mempersiapkan dataset agar bisa dibaca oleh *TensorFlow*

e. Konfigurasi model

Melakukan konfigurasi pada `pipeline.config` model *pre-trained SSDMobileNetV2 FPNLite*,

f. Implementasi dan pelatihan model

Pada tahap ini, desain model yang mencakup arsitektur *CNN* khususnya model *pre-trained SSDMobileNetV2 FPNLite* diterapkan pada dataset untuk mengarahkan checkpoint, menentukan jumlah kelas, menyesuaikan parameter pelatihan seperti batch size dan *Learning Rate*. Selanjutnya, model akan mulai dilatih agar mampu belajar dan mengenali pola dari gambar.

g. Evaluasi Model

Pada tahap ini, performa model akan dilatih dan diuji kembali menggunakan dataset validasi dan pengujian.

h. Konversi model ke *TFLite*

Model yang sudah dilatih akan dikonversi ke format *TensorFlowLite* sekaligus konversi ke *TFLite* metadata, dengan tujuan membuat model lebih ringan dan dapat digunakan di aplikasi *Android*.

i. Integrasi model ke aplikasi

Model *TensorFlow Lite* diterapkan ke dalam aplikasi *Android* dengan menambahkan *library TensorFlow Lite* ke proyek *Android Studio*.

j. Pengujian aplikasi

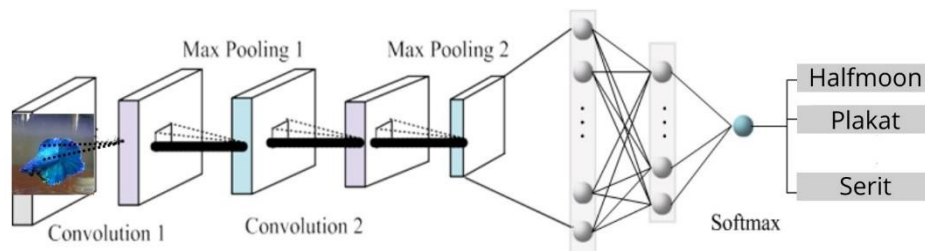
Aplikasi yang sudah dibangun akan diuji untuk memastikan semua fitur, seperti klasifikasi gambar, klasifikasi *Realtime*, dan antarmuka pengguna berjalan dengan lancar.

k. *Output* aplikasi

Output untuk klasifikasi gambar akan berupa nama kelas dan informasi ciri jenis ikan cupang. *Output* untuk klasifikasi *Realtime* akan menampilkan *bounding box* dengan kotak-kotak yang menandakan objek itu dikenali.

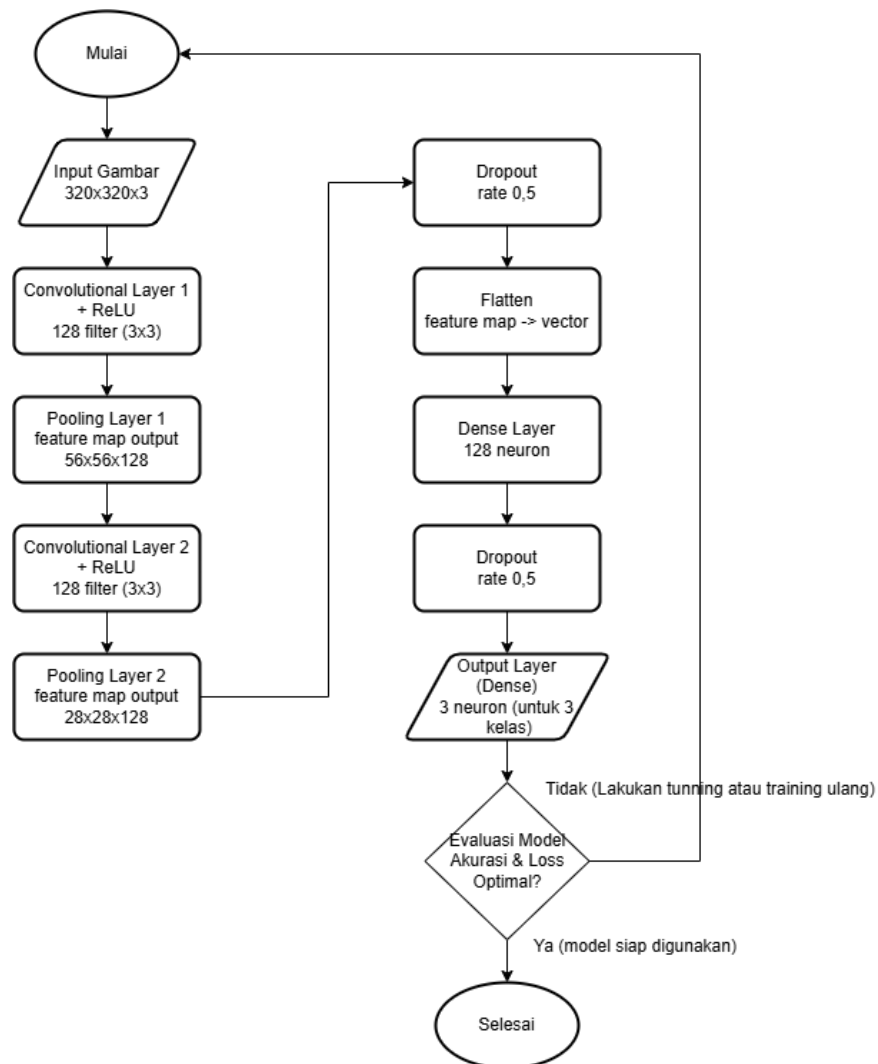
3.1.5.1. Arsitektur Algoritma CNN

Convolutional Neural network (CNN) merupakan suatu metode dalam *Deep Learning* yang dirancang untuk memproses data berbentuk array multidimensi, seperti halnya citra berwarna yang terdiri atas tiga array dua dimensi yang merepresentasikan intensitas piksel pada tiga kanal warna (merah, hijau, dan biru). Dalam penelitian ini, arsitektur *CNN* yang digunakan terdiri dari beberapa komponen utama. Pertama, *Convolutional Layer* berfungsi untuk melakukan proses konvolusi terhadap citra *input* guna mengekstraksi fitur-fitur penting. Selanjutnya, fungsi aktivasi ReLU diterapkan untuk memperkenalkan non-linearitas dalam jaringan. Setelah itu, *Pooling Layer* digunakan untuk meReduksi dimensi spasial dari fitur yang dihasilkan oleh proses konvolusi, sehingga dapat menurunkan beban komputasi dan mencegah *overfitting*. Lapisan berikutnya adalah *fully-connected layer*, yang menghubungkan seluruh neuron dari lapisan sebelumnya untuk menghasilkan representasi akhir dari fitur yang telah dipelajari. Terakhir, fungsi aktivasi *Softmax* digunakan untuk mengeluarkan nilai probabilitas terhadap masing-masing kelas, sehingga memungkinkan model dalam melakukan klasifikasi terhadap *input* yang diberikan. Arsitektur dari *CNN* secara umum dapat dilihat pada Gambar 3.4



Gambar 3. 4 Arsitektur Algoritma CNN

Gambar 3.5 berikut menyajikan *Flowchart* yang menggambarkan alur proses pelatihan model *Convolutional Neural network (CNN)* dalam penelitian ini, dimulai dari tahap pemasukan data hingga evaluasi model untuk menentukan apakah model tersebut telah memenuhi kriteria kelayakan untuk digunakan atau belum.



Gambar 3. 5 *Flowchart* Algoritma CNN

Berikut merupakan uraian dari setiap tahapan dalam *Flowchart* proses pelatihan model *Convolutional Neural network (CNN)* yang digunakan dalam penelitian ini:

1. *Input* gambar

Tahap awal dimulai dengan memasukkan data citra ke dalam model. Dalam penelitian ini, citra yang digunakan memiliki resolusi

320x320 piksel dengan tiga saluran warna (RGB), sehingga direpresentasikan dalam dimensi 320x320x3.

2. *Convolutional Layer + ReLU*

Pada tahap ini, citra diproses oleh lapisan konvolusional yang berfungsi mengekstraksi fitur-fitur penting seperti tepi, tekstur, dan pola bentuk melalui penggunaan filter atau kernel. Selanjutnya, fungsi aktivasi ReLU diterapkan untuk memungkinkan model mempelajari hubungan non-linear pada data.

3. *Pooling Layer*

Lapisan *pooling* berfungsi untuk mereduksi dimensi *feature map* hasil konvolusi, dengan tujuan mempertahankan informasi penting sekaligus mengurangi kompleksitas komputasi.

4. Pengulangan Proses Konvolusi dan *Pooling*

Tahapan konvolusi dan *pooling* diulang sebanyak dua kali. Pengulangan ini bertujuan agar model mampu mengenali fitur yang lebih kompleks dan abstrak dari citra *input* melalui tahapan bertingkat.

5. *Dropout*

Sebelum masuk ke tahap klasifikasi, diterapkan lapisan *Dropout* yang berfungsi untuk menonaktifkan secara acak sejumlah neuron selama proses pelatihan guna mencegah terjadinya *overfitting*.

6. *Flatten*

Keluaran dari proses ekstraksi fitur yang masih berbentuk matriks dua dimensi diubah menjadi vektor satu dimensi melalui proses *Flattening*, agar dapat diproses oleh lapisan dense berikutnya.

7. *Dense Layer*

Lapisan ini merupakan lapisan *fully-connected* di mana setiap neuron terhubung dengan semua neuron pada lapisan sebelumnya. Lapisan ini berperan dalam proses klasifikasi berdasarkan fitur-fitur yang telah diekstraksi.

8. *Output Layer*

Lapisan *output* merupakan lapisan dense terakhir yang menghasilkan prediksi akhir. Jumlah neuron pada lapisan ini

disesuaikan dengan jumlah kelas yang akan diklasifikasikan, dalam hal ini sebanyak tiga neuron untuk tiga jenis ikan cupang.

9. Evaluasi Model

Setelah pelatihan selesai, model dievaluasi menggunakan metrik performa seperti *accuracy* dan *Loss*. Nilai *accuracy* menunjukkan tingkat akurasi prediksi model, sedangkan *Loss* mengukur sejauh mana hasil prediksi berbeda dari nilai sebenarnya.

10. Keputusan (Apakah Model Sudah Optimal?)

Pada tahap akhir dilakukan pengambilan keputusan. Jika model menunjukkan nilai *accuracy* yang tinggi dan *Loss* yang rendah (dianggap telah optimal), maka pelatihan dihentikan dan model siap disimpan serta digunakan. Namun, apabila hasil evaluasi belum memuaskan, maka proses akan kembali ke tahap pelatihan dengan kemungkinan melakukan penyesuaian seperti penyetelan hyperparameter, penambahan data pelatihan, atau perubahan arsitektur jaringan.

3.1.5.2. Perhitungan Algoritma CNN

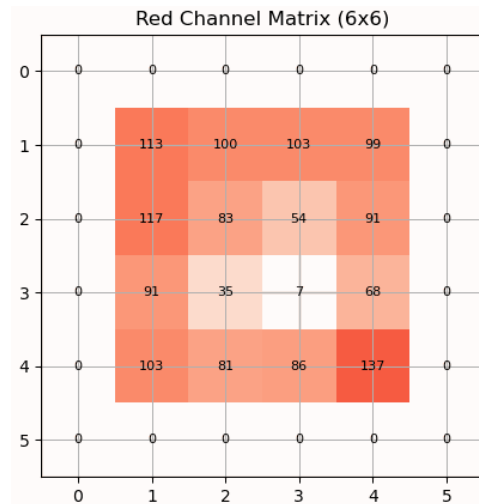
1. *Convolutional Layer*

Berikut adalah citra RGB dari *input* gambar ikan cupang dengan ukuran pixel 320x320

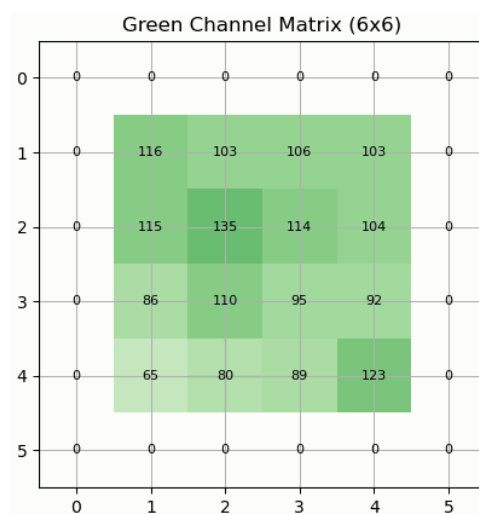


Dalam penelitian ini, peneliti menggunakan sampel berupa deret angka sebagai data *input*. Mengingat adanya keterbatasan dalam penulisan dan visualisasi data pada citra berukuran 320x320 piksel, maka peneliti

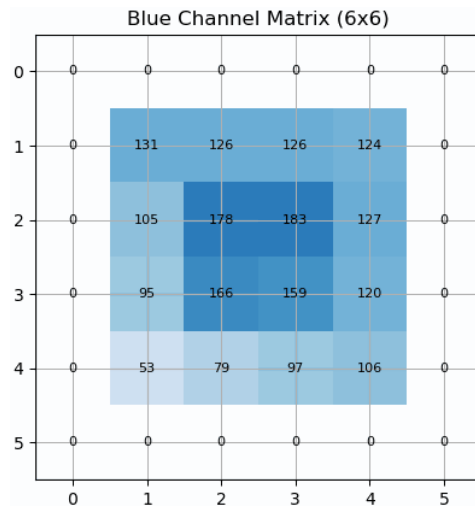
menyederhanakan data dengan menggunakan sampel deret angka berukuran 6x6 piksel. Adapun proses konvolusi dilakukan menggunakan kernel berukuran 3x3 piksel. Berikut adalah *matrix channel* RGB:



Gambar 3. 6 *Channel Red*



Gambar 3. 7 *Channel Green*



Gambar 3. 8 Channel Blue

Dalam penelitian ini filter yang digunakan berukuran 3x3 untuk konvolusi dapat dilihat dibawah ini.

1	0	-1
1	0	-1
1	0	-1

Dengan menggunakan filter 3x3 dilanjutkan dengan proses perhitungan konvolusi tersebut dapat dihitung seperti proses dibawah ini.

Red

Posisi (0,0)

$$(1*0)+(0*0)+(-1*0)+(1*0)+(0*113)+(-1*100)+(1*0)+(0*117)+(-1*83) \\ = -183$$

Posisi (0,1)

$$(1*0)+(0*0)+(-1*0)+(1*113)+(0*100)+(-1*103)+(1*117)+(0*83)+(-1*54) = 73$$

Posisi (0,2)

$$(1*0)+(0*0)+(-1*0)+(1*100)+(0*103)+(-1*99)+(1*83)+(0*54)+(-1*91) \\ = -7$$

Posisi (0,3)

$$(1*0)+(0*0)+(-1*0)+(1*103)+(0*99)+(-1*0)+(1*54)+(0*91)+(-1*0) \\ = 157$$

Posisi (1,0)

$$(1*0)+(0*113)+(-1*100)+(1*0)+(0*117)+(-1*83)+(1*0)+(0*91)+(-1*35) \\ = -218$$

Posisi (1.1)

$$(1*113)+(0*100)+(-1*103)+(1*117)+(0*83)+(-1*54)+(1*91)+(0*35)+(-1*7)= 157$$

Posisi (1.2)

$$(1*100)+(0*103)+(-1*99)+(1*83)+(0*54)+(-1*91)+(1*35)+(0*7)+(-1*68)= -40$$

Posisi (1.3)

$$(1*103)+(0*99)+(-1*0)+(1*54)+(0*91)+(-1*0)+(1*7)+(0*68)+(-1*0) \\ = 164$$

Posisi (2,0)

$$(1*0)+(0*117)+(-1*83)+(1*0)+(0*91)+(-1*35)+(1*0)+(0*103)+(-1*81) \\ = -199$$

Posisi (2,1)

$$(1*117)+(0*83)+(-1*54)+(1*91)+(0*35)+(-1*7)+(1*103)+(0*81)+(-1*86)= 164$$

Posisi (2,2)

$$(1*83)+(0*54)+(-1*91)+(1*35)+(0*7)+(-1*68)+(1*81)+(0*86)+(-1*137) \\ = -97$$

Posisi (2,3)

$$(1*54)+(0*91)+(-1*0)+(1*7)+(0*68)+(-1*0)+(1*86)+(0*137)+(-1*0) \\ = 147$$

Posisi (3,0)

$$(1*0)+(0*91)+(-1*35)+(1*0)+(0*103)+(-1*81)+(1*0)+(0*0)+(-1*0) \\ = -116$$

Posisi (3,1)

$$(1*91)+(0*35)+(-1*7)+(1*103)+(0*81)+(-1*86)+(1*0)+(0*0)+(-1*0) \\ = 101$$

Posisi (3,2)

$$(1*35)+(0*7)+(-1*68)+(1*81)+(0*86)+(-1*137)+(1*0)+(0*0)+(-1*0)$$

$$= -89$$

Posisi (3,3)

$$(1*7)+(0*68)+(-1*0)+(1*86)+(0*137)+(-1*0)+(1*0)+(0*0)+(-1*0)= 93$$

Hasil

-183	73	-7	157
-218	157	-40	164
-199	164	-97	147
-116	101	-89	93

Green

Posisi (0,0)

$$(1*0)+(0*0)+(-1*0)+(1*0)+(0*116)+(-1*103)+(1*0)+(0*115)+(-1*135)$$

$$= -238$$

Posisi (0,1)

$$(1*0)+(0*0)+(-1*0)+(1*116)+(0*103)+(-1*106)+(1*115)+(0*135)+(-1*114)= 11$$

Posisi (0,2)

$$(1*0)+(0*0)+(-1*0)+(1*103)+(0*106)+(-1*103)+(1*135)+(0*114)+(-1*104)= 31$$

Posisi (0,3)

$$(1*0)+(0*0)+(-1*0)+(1*106)+(0*103)+(-1*0)+(1*114)+(0*104)+(-1*0)= 220$$

Posisi (1,0)

$$(1*0)+(0*116)+(-1*103)+(1*0)+(0*115)+(-1*135)+(1*0)+(0*86)+(-1*110)= -348$$

Posisi (1,1)

$$(1*116)+(0*103)+(-1*106)+(1*115)+(0*135)+(-1*114)+(1*86)+(0*110)+(-1*95)= 2$$

Posisi (1,2)

$$(1*103)+(0*106)+(-1*103)+(1*135)+(0*114)+(-1*104)+(1*110)+(0*95)+(-1*92)= 49$$

Posisi (1,3)

$$(1*106)+(0*103)+(-1*0)+(1*114)+(0*104)+(-1*0)+(1*95)+(0*92)+(-1*0)=315$$

Posisi (2,0)

$$(1*0)+(0*115)+(-1*135)+(1*0)+(0*86)+(-1*110)+(1*0)+(0*65)+(-1*80)=-325$$

Posisi (2,1)

$$(1*115)+(0*135)+(-1*114)+(1*86)+(0*110)+(-1*95)+(1*65)+(0*80)+(-1*89)=-32$$

Posisi (2,2)

$$(1*135)+(0*114)+(-1*104)+(1*110)+(0*95)+(-1*92)+(1*80)+(0*89)+(-1*123)=6$$

Posisi (2,3)

$$(1*114)+(0*104)+(-1*0)+(1*95)+(0*92)+(-1*0)+(1*89)+(0*123)+(-1*0)=298$$

Posisi (3,0)

$$(1*0)+(0*86)+(-1*110)+(1*0)+(0*65)+(-1*80)+(1*0)+(0*0)+(-1*0)=-190$$

Posisi (3,1)

$$(1*86)+(0*110)+(-1*95)+(1*65)+(0*80)+(-1*89)+(1*0)+(0*0)+(-1*0)=-33$$

Posisi (3,2)

$$(1*110)+(0*95)+(-1*92)+(1*80)+(0*89)+(-1*123)+(1*0)+(0*0)+(-1*0)=-25$$

Posisi (3,3)

$$(1*95)+(0*92)+(-1*0)+(1*89)+(0*123)+(-1*0)+(1*0)+(0*0)+(-1*0)=184$$

Hasil

-238	11	31	220
-348	2	49	315
-325	-32	6	298

-190	-33	-25	184
------	-----	-----	-----

Blue

Posisi (0,0)

$$(1 \times 0) + (0 \times 0) + (-1 \times 0) + (1 \times 0) + (0 \times 131) + (-1 \times 126) + (1 \times 0) + (0 \times 105) + (-1 \times 178) = -304$$

Posisi (0,1)

$$(1 \times 0) + (0 \times 0) + (-1 \times 0) + (1 \times 131) + (0 \times 126) + (-1 \times 126) + (1 \times 105) + (0 \times 178) + (-1 \times 183) = -73$$

Posisi (0,2)

$$(1 \times 0) + (0 \times 0) + (-1 \times 0) + (1 \times 126) + (0 \times 126) + (-1 \times 124) + (1 \times 178) + (0 \times 183) + (-1 \times 127) = 53$$

Posisi (0,3)

$$(1 \times 0) + (0 \times 0) + (-1 \times 0) + (1 \times 126) + (0 \times 124) + (-1 \times 0) + (1 \times 183) + (0 \times 127) + (-1 \times 0) = 309$$

Posisi (1,0)

$$(1 \times 0) + (0 \times 131) + (-1 \times 126) + (1 \times 0) + (0 \times 105) + (-1 \times 178) + (1 \times 0) + (0 \times 95) + (-1 \times 166) = -470$$

Posisi (1,1)

$$(1 \times 131) + (0 \times 126) + (-1 \times 126) + (1 \times 105) + (0 \times 178) + (-1 \times 183) + (1 \times 95) + (0 \times 166) + (-1 \times 159) = -137$$

Posisi (1,2)

$$(1 \times 126) + (0 \times 126) + (-1 \times 124) + (1 \times 178) + (0 \times 183) + (-1 \times 127) + (1 \times 166) + (0 \times 159) + (-1 \times 120) = 99$$

Posisi (1,3)

$$(1 \times 126) + (0 \times 124) + (-1 \times 0) + (1 \times 183) + (0 \times 127) + (-1 \times 0) + (1 \times 159) + (0 \times 120) + (-1 \times 0) = 468$$

Posisi (2,0)

$$(1 \times 0) + (0 \times 105) + (-1 \times 178) + (1 \times 0) + (0 \times 95) + (-1 \times 166) + (1 \times 0) + (0 \times 53) + (-1 \times 79) = -423$$

Posisi (2,1)

$$(1 \times 105) + (0 \times 178) + (-1 \times 183) + (1 \times 95) + (0 \times 166) + (-1 \times 159) + (1 \times 53) + (0 \times 79) + (-1 \times 97) = -186$$

Posisi (2,2)

$$(1 \times 178) + (0 \times 183) + (-1 \times 127) + (1 \times 166) + (0 \times 159) + (-1 \times 120) + (1 \times 79) + (0 \times 97) + (-1 \times 106) = 70$$

Posisi (2,3)

$$(1 \times 183) + (0 \times 127) + (-1 \times 0) + (1 \times 159) + (0 \times 120) + (-1 \times 0) + (1 \times 97) + (0 \times 106) + (-1 \times 0) = 439$$

Posisi (3,0)

$$(1 \times 0) + (0 \times 95) + (-1 \times 166) + (1 \times 0) + (0 \times 53) + (-1 \times 79) + (1 \times 0) + (0 \times 0) + (-1 \times 0) = -245$$

Posisi (3,1)

$$(1 \times 95) + (0 \times 166) + (-1 \times 159) + (1 \times 53) + (0 \times 79) + (-1 \times 97) + (1 \times 0) + (0 \times 0) + (-1 \times 0) = -108$$

Posisi (3,2)

$$(1 \times 166) + (0 \times 159) + (-1 \times 120) + (1 \times 79) + (0 \times 97) + (-1 \times 106) + (1 \times 0) + (0 \times 0) + (-1 \times 0) = 19$$

Posisi (3,3)

$$(1 \times 159) + (0 \times 120) + (-1 \times 0) + (1 \times 97) + (0 \times 106) + (-1 \times 0) + (1 \times 0) + (0 \times 0) + (-1 \times 0) = 256$$

Hasil

-304	-73	53	309
-470	-137	99	468
-423	-186	70	439
-245	-108	19	256

Setelah proses konvolusi dilakukan secara terpisah pada masing-masing kanal warna merah (*Red*), hijau (*Green*), dan biru (*Blue*) langkah selanjutnya adalah melakukan penjumlahan terhadap hasil konvolusi dari ketiga kanal tersebut untuk memperoleh representasi akhir dari fitur yang telah diekstraksi.

-725	11	77	686
------	----	----	-----

-1036	22	108	947
-947	-54	-21	884
-551	-40	-95	533

2. ReLU

Pada tahap ini, fungsi aktivasi ReLU diterapkan dengan tujuan untuk mengubah nilai piksel pada citra, di mana seluruh nilai yang bernilai negatif atau kurang dari nol akan disubstitusi menjadi nol. Proses ini dilakukan guna mempertahankan hanya informasi positif yang relevan dalam representasi fitur citra.

0	11	77	686
0	22	108	947
0	0	0	884
0	0	0	533

3. Pooling Layer

Setelah penerapan fungsi aktivasi ReLU, tahap selanjutnya adalah proses *pooling* dengan menggunakan metode *max pooling*. Teknik ini berfungsi untuk mengambil nilai maksimum dari setiap wilayah lokal pada citra *input* berdasarkan ukuran kernel 2x2. Dengan demikian, hanya nilai tertinggi dari tiap area kecil yang dipertahankan, sehingga dimensi data dapat direduksi tanpa menghilangkan fitur penting. Ilustrasi dari proses *max pooling* ditampilkan pada bagian berikut.

0	11	77	686
0	22	108	947
0	0	0	884
0	0	0	533

0	11	77	686
0	22	108	947
0	0	0	884
0	0	0	533

0	11	77	686
0	22	108	947
0	0	0	884
0	0	0	533

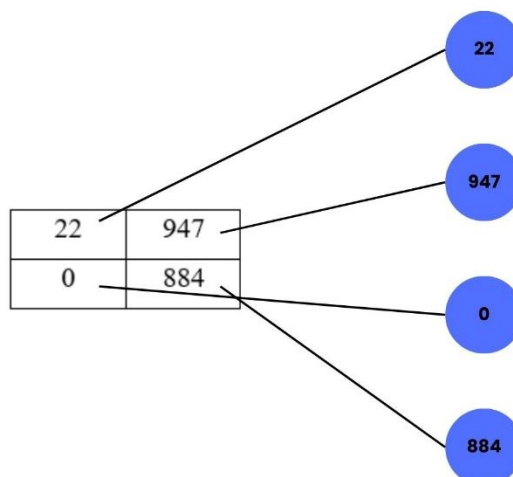
0	11	77	686
0	22	108	947
0	0	0	884
0	0	0	533

Hasil

22	947
0	884

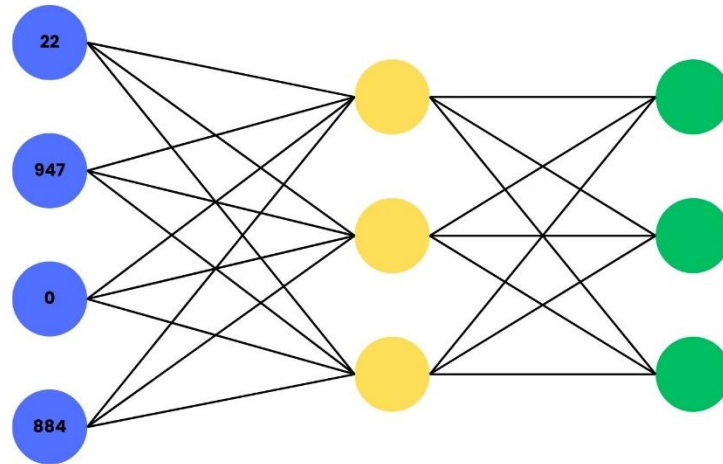
4. *Flattening Layer*

Setelah melalui lapisan *pooling* dengan metode *max pooling*, data kemudian diproses pada lapisan *flatten* untuk mengubahnya menjadi vektor satu dimensi yang selanjutnya digunakan sebagai masukan pada lapisan *fully connected*.



Gambar 3. 9 *Flattening Layer*

5. Fully-Connected Layer



Gambar 3. 10 *Fully-connected Layer*

Gambar 3.10 menunjukkan struktur lapisan *fully connected* yang dihasilkan setelah proses *Flattening*.

$$J1 = (22*0,1) + (947*0,1) + (0*0,1) + (884*0,1) = 2,2 + 94,7 + 0 + 88,4 = 185,3$$

$$J2 = (22*0,2) + (947*0,2) + (0*0,2) + (884*0,2) = 4,4 + 189,4 + 0 + 176,8 = 370,6$$

$$J3 = (22*0,3) + (947*0,3) + (0*0,3) + (884*0,3) = 6,6 + 284,1 + 0 + 265,2 = 555,9$$

Hidden layer pertama direpresentasikan oleh J1, J2, dan J3, di mana setiap neuron dari I1, I2, dan I3 dikalikan dengan bobot (weightweight) yang berbeda-beda untuk menghasilkan nilai J1=185,3, J2=370,6, J3=555,9. Nilai-nilai tersebut kemudian melalui proses serupa, yaitu dikalikan dengan bobot yang berbeda, untuk menghasilkan H1, H2, H3, dan H4 yang menjadi *hidden layer* berikutnya.

$$H1 = (185,3*0,3) + (370,6*0,3) + (555,9*0,3) = 55,59 + 111,18 + 166,77 = 333,54$$

$$H2 = (185,3*0,2) + (370,6*0,2) + (555,9*0,2) = 37,06 + 74,12 + 111,18 = 222,36$$

$$H3 = (185,3*0,1) + (370,6*0,1) + (555,9*0,1) = 18,53 + 37,06 + 55,59 = 111,18$$

Langkah selanjutnya, setiap neuron H1, H2, dan H3 kemudian dikalikan lagi dengan nilai weight yang berbeda-beda untuk menghasilkan nilai O1, O2, dan O3 sebagai *hidden layer* selanjutnya.

$$O1 = (333,54*0,3) + (222,36*0,3) + (111,18*0,3) = 100,062 + 66,708 + 33,354 = 200,124$$

$$O2 = (333,54*0,2) + (222,36*0,2) + (111,18*0,2) = 66,708 + 44,472 + 22,236 = 133,416$$

$$O3 = (333,54*0,1) + (222,36*0,1) + (111,18*0,1) = 33,354 + 22,236 + 11,118 = 66,708$$

Proses perhitungan fungsi aktivasi *Softmax* dilakukan dengan mengubah setiap nilai *output* menjadi nilai probabilitas. Hal ini dilakukan dengan cara membagi nilai eksponensial dari masing-masing *output* (O1, O2, dan O3) dengan jumlah total dari seluruh nilai eksponensial *output* tersebut.

$$Softmax (O) = \frac{e^{O_i}}{\sum (e^{O_j})} =$$

$$S(O1) = \frac{e^{200,124}}{e^{200,124} + e^{133,416} + e^{66,708}} = \frac{543,993}{543,993 + 362,662 + 181,331} = 0,5$$

$$S(O2) = \frac{e^{133,416}}{e^{200,124} + e^{133,416} + e^{66,708}} = \frac{362,662}{543,993 + 362,662 + 181,331} = 0,3333$$

$$S(O3) = \frac{e^{66,708}}{e^{200,124} + e^{133,416} + e^{66,708}} = \frac{181,331}{543,993 + 362,662 + 181,331} = 0,1667$$

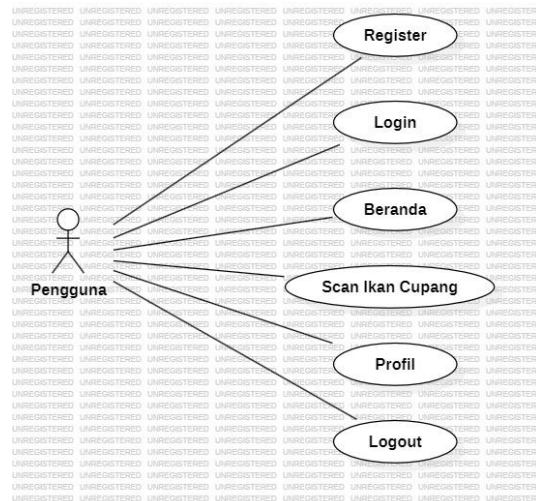
$$S(O1) + S(O2) + S(O3) = 0,5 + 0,3333 + 0,1667 = 1$$

Berdasarkan bobot yang didapatkan, untuk nilai probabilitas yang lebih besar ada di O1 yaitu 0,5 yang berarti *input* citra yang dimasukkan diprediksi adalah ikan cupang jenis *Halfmoon*.

3.2 Perancangan Sistem (System Design)

Perancangan sistem bertujuan untuk merepresentasikan seluruh proses kegiatan yang akan diterapkan dalam sebuah sistem. Dalam menggambarkan sebuah proses, dimodelkan dengan menggunakan *Unified Modelling Language* (UML).

3.2.1 Use Case Diagram



Gambar 3. 11 Use Case Diagram

Diagram ini menggambarkan sebuah aplikasi yang berfokus pada identifikasi atau klasifikasi ikan cupang menggunakan gambar dan klasifikasi secara *Realtime*. Aplikasi ini juga menyediakan fitur dasar manajemen akun (*Register*, *Login*, *logout*) dan informasi tambahan (deskripsi ikan, tutorial, info pengembang). Alur umum yang tersirat adalah pengguna mendaftar/*Login*, kemudian dapat menggunakan fitur inti klasifikasi, dan ada juga fitur informasi dan dukungan.

a. Skenario Use Case Login

Tabel 3. 1 Login

SKENARIO 1	
ID Use Case	001
Nama Use Case	Login

Tujuan	Untuk dapat masuk ke halaman beranda
Deskripsi	
Prioritas	<i>Primary</i>
Actor	Pengguna
Skenario Utama	
Pre condition	Sudah Melakukan <i>Register</i>
Aksi Aktor	Reaksi Sistem
1. Mengisi Email dan Kata sandi	2. Sistem melakukan validasi
	3. Sistem Menampilkan Halaman Beranda
Post condition	Jika Berhasil, “Pengguna akan dialihkan kehalaman beranda” Jika Gagal, “Pengguna harus melakukan pengisian ulang pada form Login.”.

b. Skenario *Use Case Register*

Tabel 3. 2 *Register*

SKENARIO 2	
ID <i>Use Case</i>	002
Nama <i>Use Case</i>	<i>Register</i>
Tujuan	Untuk melakukan pendaftaran akun
Deskripsi	
Prioritas	<i>Primary</i>
Actor	Pengguna
Skenario Utama	

Pre condition	Sudah mengakses halaman utama aplikasi
Aksi Aktor	Reaksi Sistem
1. Klik tombol Daftar	2. Menampilkan halaman Daftar
3. Mengisi halaman daftar, Nama, Email, dan Kata sandi 4. Klik centang pada box konfirmasi 5. Klik tombol Daftar	6. Menampilkan proses 7. Menampilkan halaman <i>Login</i>
Post condition	Jika Berhasil, “Pengguna akan dialihkan kehalaman <i>Login</i> ” Jika Gagal, “ Pengguna harus melakukan pengecekan ulang pada form <i>Register</i> ”.

c. Skenario *Use Case* Deskripsi Ikan Cupang

Tabel 3. 3 Deskripsi Ikan Cupang

SKENARIO 3	
ID Use Case	003
Nama Use Case	Deskripsi Ikan Cupang
Tujuan	Untuk mengetahui perbedaan lebih dalam tentang jenis-jenis ikan cupang
Deskripsi	
Prioritas	<i>Primary</i>
Actor	Pengguna
Skenario Utama	
Pre condition	telah melakukan <i>Login</i>
Aksi Aktor	Reaksi Sistem

1. Memilih salahsatu dari jenis ikan cupang	2. Menampilkan halaman deskripsi ikan cupang
Post condition	Sistem menampilkan halaman deskripsi ikan cupang

d. Skenario *Use Case* Tutorial Aplikasi

Tabel 3. 4 Tutorial Aplikasi

SKENARIO 4	
ID <i>Use Case</i>	004
Nama <i>Use Case</i>	Tutorial Aplikasi
Tujuan	Mengetahui cara menggunakan aplikasi
Deskripsi	
Prioritas	<i>Primary</i>
Actor	Pengguna
Skenario Utama	
Pre condition	Sudah melakukan <i>Login</i>
Aksi Aktor	Reaksi Sistem
1. Memilih menu Profil	2. Menampilkan pilihan tutorial aplikasi, tentang aplikasi, dan keluar aplikasi
3. Pengguna memilih “Tutorial Aplikasi”	4. Menampilkan seputar cara menggunakan aplikasi
Post condition	Sistem menampilkan informasi untuk penggunaan aplikasi

e. Skenario *Use Case* Tentang Aplikasi

Tabel 3. 5 Tentang Aplikasi

SKENARIO 5	
ID Use Case	005
Nama Use Case	Tentang Aplikasi
Tujuan	Melihat informasi mengenai pengembang
Deskripsi	
Prioritas	Primary
Actor	Pengguna
Skenario Utama	
Pre condition	Sudah melakukan <i>Login</i> dan berada dihalaman profil
Aksi Aktor	Reaksi Sistem
1. Memilih “Tentang Aplikasi” pembuat	2. Menampilkan informasi pengembang
3. Melihat informasi pengembang	
Post condition	Sistem menampilkan informasi mengenai pengembang aplikasi.

f. Skenario Use Case Klasifikasi Gambar dari Galeri

Tabel 3. 6 Klasifikasi Gambar dari Galeri

SKENARIO 6	
ID Use Case	006
Nama Use Case	Klasifikasi Gambar dari Galeri
Tujuan	Mengklasifikasikan jenis ikan cupang(<i>Halfmoon</i> , <i>Plakat</i> , dan <i>Serit</i>) dengan <i>input</i> gambar dari galeri
Deskripsi	

Prioritas	<i>Primary</i>
Actor	Pengguna
Skenario Utama	
Pre condition	Telah berada di halaman Scan Cupang
Aksi Aktor	Reaksi Sistem
1. Menekan tombol “Galeri”	2. Menampilkan galeri
3. Memilih gambar ikan cupang yang akan di klasifikasi	4. Menampilkan gambar di halaman Scan Cupang
5. Menekan tombol “Pindai Gambar”	6. Memproses gambar 7. Mengklasifikasikan gambar ikan cupang dengan algoritma <i>CNN</i> dan <i>framework TensorFlow</i> 8. Menampilkan hasil klasifikasi gambar
Post condition	Jika berhasil, “Sistem akan menampilkan jenis ikan cupang(<i>Halfmoon</i> , <i>Plakat</i> , dan <i>Serit</i>), nilai probabilitas, deskripsi jenis ikan cupang dan detail probabilitas” Jika Gagal, “ Sistem akan menampilkan Hasil Prediksi Tidak Dikenali ”.

g. Skenario *Use Case* Klasifikasi Gambar dari Kamera

Tabel 3. 7 Klasifikasi Gambar dari Kamera

SKENARIO 7	
ID Use Case	007
Nama Use Case	Klasifikasi Gambar dari Kamera

Tujuan	Mengklasifikasikan jenis ikan cupang(<i>Halfmoon</i> , <i>Plakat</i> , dan <i>Serit</i>) dengan <i>input</i> gambar dari kamera
Deskripsi	
Prioritas	<i>Primary</i>
Actor	Pengguna
Skenario Utama	
<i>Pre condition</i>	Telah berada di halaman Scan Cupang
Aksi Aktor	Reaksi Sistem
1. Menekan tombol “Kamera”	2. Menampilkan kamera
3. Mengarahkan kamera ke objek ikan cupang	4. Menampilkan gambar di halaman Scan Cupang
5. Menekan tombol “Pindai Gambar”	6. Memproses gambar 7. Mengklasifikasikan gambar ikan cupang dengan algoritma <i>CNN</i> dan <i>framework TensorFlow</i> 8. Menampilkan hasil klasifikasi gambar
<i>Post condition</i>	Jika berhasil, “Sistem akan menampilkan jenis ikan cupang(<i>Halfmoon</i> , <i>Plakat</i> , dan <i>Serit</i>), nilai probabilitas, deskripsi jenis ikan cupang dan detail probabilitas” Jika Gagal, “ Sistem akan menampilkan Hasil Prediksi Tidak Dikenali ”.

Tabel 3. 8 Klasifikasi *Realtime*

SKENARIO 8

ID Use Case	008
Nama Use Case	Klasifikasi <i>Realtime</i>
Tujuan	Mengklasifikasikan secara <i>Realtime</i> jenis ikan cupang(<i>Halfmoon</i> , <i>Plakat</i> , dan <i>Serit</i>) dengan kamera
Deskripsi	
Prioritas	<i>Primary</i>
Actor	Pengguna
Skenario Utama	
Pre condition	Telah berada di halaman Scan Cupang
Aksi Aktor	Reaksi Sistem
1. Menekan tombol “Klasifikasi <i>Realtime</i> ”	2. Menampilkan kamera <i>Realtime</i>
3. Mengarahkan kamera ke objek ikan cupang	4. Menampilkan gambar di halaman Scan Cupang
	5. Memproses gambar 6. Melakukan klasifikasi <i>Realtime</i> dengan algoritma <i>CNN</i> dan <i>framework TensorFlow</i>
Post condition	Jika berhasil, Sistem akan menampilkan hasil klasifikasi <i>Realtime</i> jenis ikan cupang(<i>Halfmoon</i> , <i>Plakat</i> , dan <i>Serit</i>) dengan <i>bounding box</i> , nama kelas, dan presentase akurasi.

h. Skenario Use Case Logout

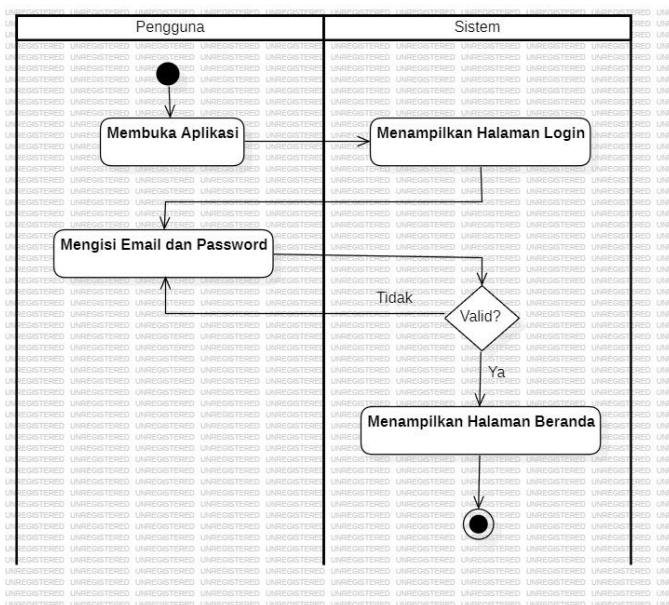
Tabel 3. 9 Logout

SKENARIO 8	
ID Use Case	008

Nama Use Case	Logout
Tujuan	Untuk keluar dari akun
Deskripsi	
Prioritas	<i>Primary</i>
Actor	Pengguna
Skenario Utama	
Pre condition	Telah berada di halaman Profil
Aksi Aktor	Reaksi Sistem
3. Memilih “Keluar” dari akun anda	4. Memproses 5. Menampilkan halaman <i>Login</i>
Post condition	Sistem menampilkan halaman awal <i>Login</i> .

3.2.2 Activity Diagram

a. Activity Diagram Login

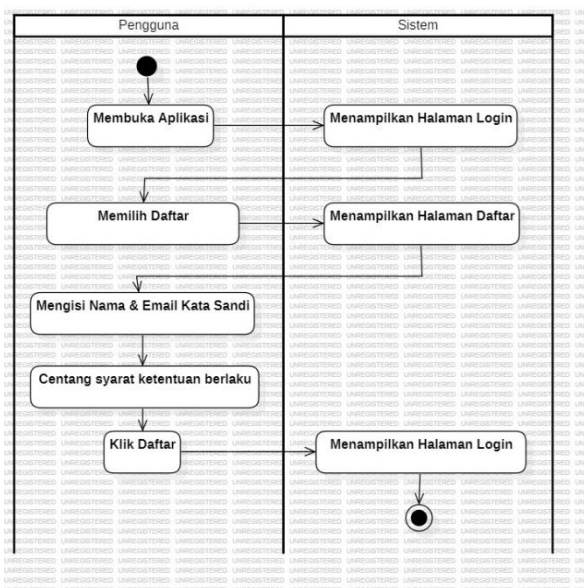


Gambar 3. 12 Activity Diagram Login

Diagram yang disajikan merupakan sebuah *Activity Diagram* yang

mengilustrasikan alur proses *Login* bagi pemilik dan pembeli pada sebuah aplikasi. Dimulai dari inisiasi oleh pemilik atau pembeli, proses diawali ketika mereka "Membuka Aplikasi". Setelah itu, sistem akan merespons dengan "Menampilkan Halaman *Login*", tempat pemilik dan pembeli diharapkan untuk "Mengisi Email dan Password" yang terdaftar. Kemudian, sistem akan melakukan validasi terhadap *kRedensial* yang dimasukkan; jika ditemukan "Tidak Valid", maka akan diarahkan kembali untuk mengulang proses pengisian email dan password. Namun, apabila *kRedensial* dinyatakan "Valid", sistem akan melanjutkan dengan "Menampilkan Halaman Beranda", yang menandai selesainya alur proses *Login* ini.

b. Activity Diagram Register

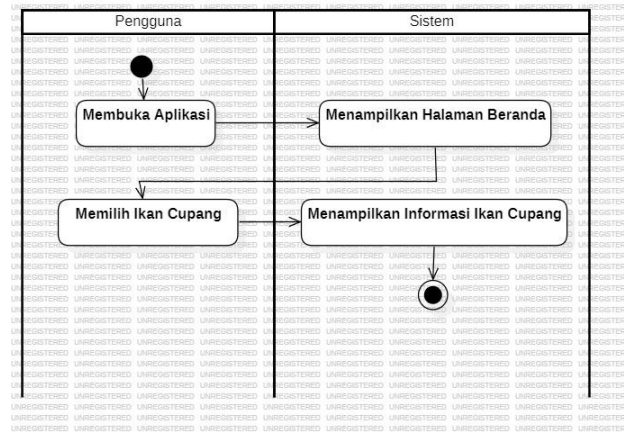


Gambar 3. 13 Activity Diagram Register

Diagram aktivitas ini mengilustrasikan alur proses pendaftaran pemilik dan pembeli baru pada sebuah sistem. Proses dimulai ketika "Membuka Aplikasi", yang kemudian diikuti oleh sistem dengan "Menampilkan Halaman *Login*". Dari halaman *Login* tersebut, memiliki opsi untuk "Memilih Daftar", dan sistem akan segera merespons dengan "Menampilkan Halaman Daftar". Di halaman pendaftaran, pemilik atau pembeli diharuskan untuk "Mengisi Nama, Email, dan Kata Sandi" mereka, serta "Centang syarat ketentuan berlaku" untuk menyetujui kebijakan yang ada. Setelah semua

informasi terisi dan syarat disetujui, pengguna kemudian "Klik Daftar". Sebagai konfirmasi pendaftaran, sistem akan kembali "Menampilkan Halaman Login", yang menandakan bahwa proses registrasi telah berhasil.

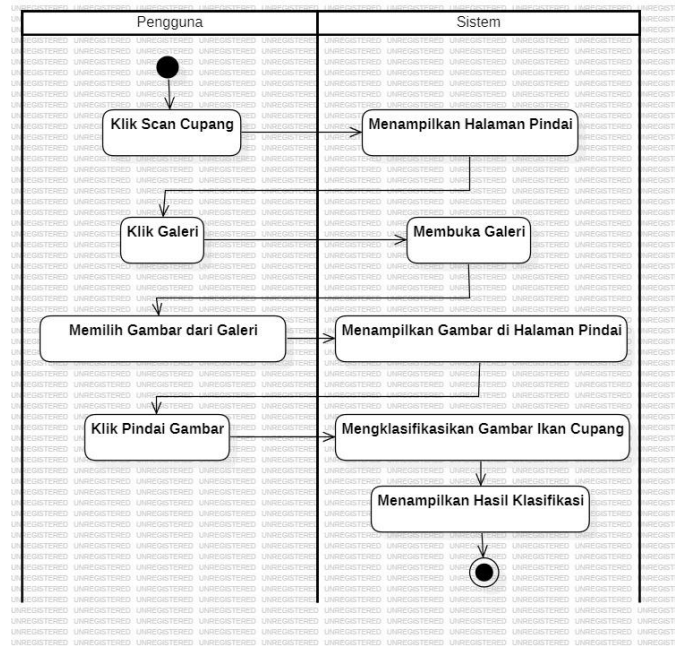
a. *Activity Diagram Deskripsi Ikan Cupang*



Gambar 3. 14 *Activity Diagram Deskripsi Ikan Cupang*

Diagram aktivitas ini menguraikan alur interaksi pemilik atau pembeli dan sistem dalam menampilkan informasi ikan cupang. Proses dimulai ketika "Membuka Aplikasi", yang kemudian direspons oleh sistem dengan "Menampilkan Halaman Beranda". Dari halaman beranda ini, pemilik atau pembeli dapat "Memilih ikan cupang" yang ingin diketahui informasinya. Setelah membuat pilihan, sistem akan melanjutkan dengan "Menampilkan Informasi tentang ikan cupang" yang relevan, kemudian alur aktivitas berakhir.

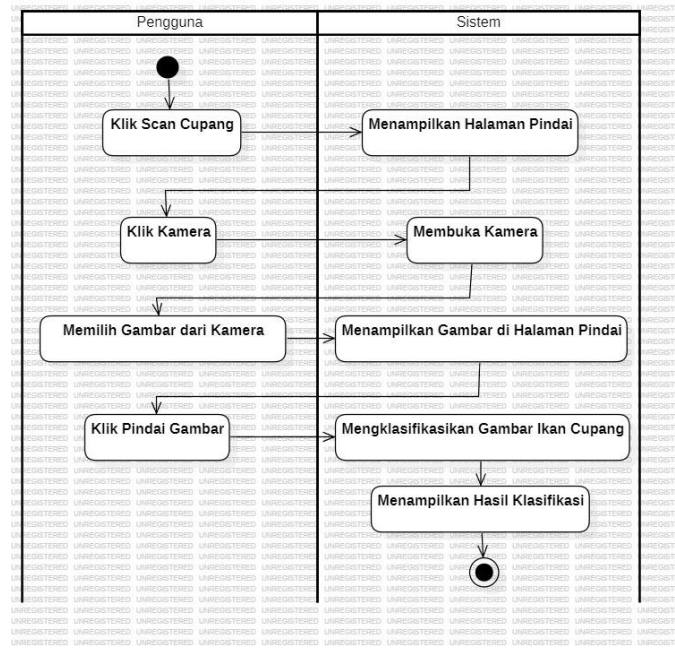
b. *Activity Diagram Klasifikasi Gambar dari Galeri*



Gambar 3. 15 Activity Diagram Klasifikasi Gambar dari Galeri

Diagram aktivitas ini menjelaskan alur proses klasifikasi gambar ikan cupang melalui fitur pemindaian dalam sebuah aplikasi. Dimulai ketika pemilik atau pembeli "Klik Scan Cupang", sistem akan merespons dengan "Menampilkan Halaman Pindai". Dari halaman tersebut, memiliki opsi untuk "Klik Galeri", yang kemudian akan membuat sistem "Membuka Galeri" perangkat. Setelah galeri terbuka, pengguna dapat "Memilih Gambar Dari Galeri" yang ingin diklasifikasikan, dan sistem akan "Menampilkan gambar di Halaman Pindai" tersebut. Langkah selanjutnya "Klik Pindai Gambar", memicu sistem untuk "Mengklasifikasikan gambar ikan cupang dengan algoritma CNN dan *framework TensorFlow*" secara otomatis. Setelah proses klasifikasi selesai, sistem akan "Menampilkan Hasil Klasifikasi" kepada pemilik atau pembeli, menandai berakhirnya alur proses ini.

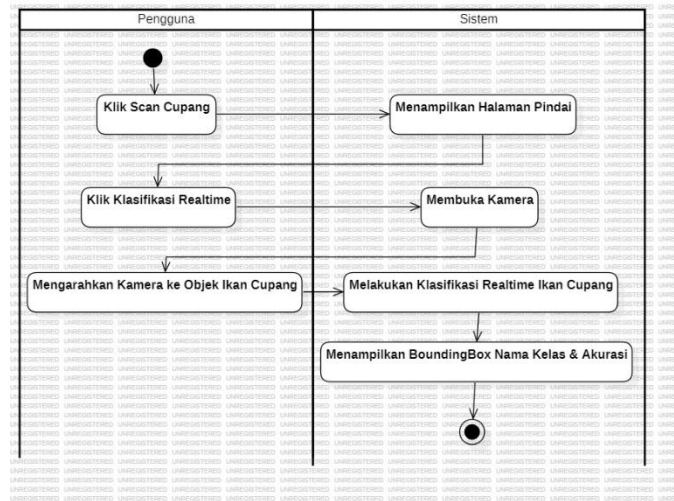
c. Activity Diagram Klasifikasi Gambar dari Kamera



Gambar 3. 16 Activity Diagram Klasifikasi Gambar dari Kamera

Diagram aktivitas ini menggambarkan alur proses klasifikasi gambar ikan cupang melalui fitur pemindaian menggunakan kamera pada sebuah aplikasi. Prosesnya dimulai ketika pemilik atau pembeli "Klik Scan Cupang", dan sistem akan segera "Menampilkan Halaman Pindai". Dari halaman tersebut, kemudian "Klik Kamera" untuk mengaktifkan fungsi kamera, yang direspons oleh sistem dengan "Membuka Kamera". Setelah kamera aktif, pemilik dan pembeli dapat "Mengambil gambar dari kamera", dan sistem akan "Menampilkan gambar di Halaman Pindai" yang baru saja diambil. Selanjutnya "Klik Pindai Gambar", yang akan memicu sistem untuk "Mengklasifikasikan gambar ikan cupang dengan algoritma CNN dan *framework TensorFlow*" secara otomatis. Setelah proses klasifikasi selesai, sistem akan "Menampilkan Hasil Klasifikasi" kepada pemilik atau pembeli, menandai berakhirnya alur proses ini.

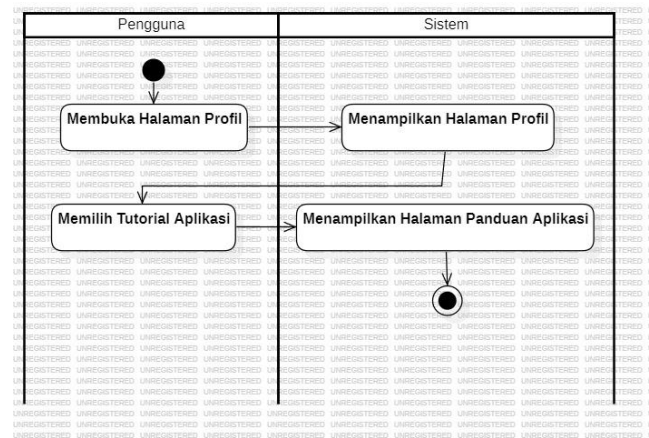
d. Activity Diagram Klasifikasi Realtime



Gambar 3. 17 Activity Diagram Klasifikasi *Realtime*

Diagram aktivitas ini menggambarkan alur proses klasifikasi ikan cupang secara *Realtime* menggunakan kamera. Prosesnya dimulai ketika pemilik atau pembeli "Klik Scan Cupang", yang membuat sistem "Menampilkan Halaman Pindai". Dari halaman pindai, kemudian "Klik Klasifikasi *Realtime*", dan sistem akan merespons dengan "Membuka Kamera". Setelah kamera aktif, pemilik atau pembeli "Mengarahkan kamera ke objek ikan cupang" yang ingin diklasifikasikan. Pada saat itu, sistem akan secara otomatis "Melakukan klasifikasi *Realtime* ikan cupang dengan algoritma *CNN* dan *framework TensorFlow*". Setelah klasifikasi selesai, sistem akan "Menampilkan *Bounding box*, nama kelas, dan persentase akurasi" dari hasil klasifikasi tersebut di layar, menandai berakhirnya alur proses ini.

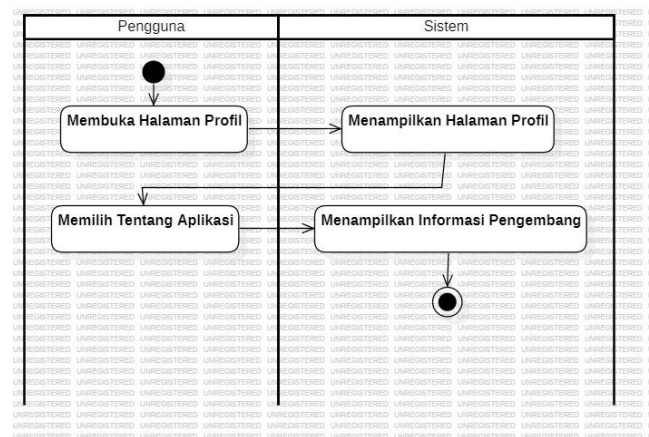
e. Activity Diagram Tutorial Aplikasi



Gambar 3. 18 Activity Diagram Tutorial Aplikasi

Diagram aktivitas ini mengilustrasikan alur sederhana bagi pemilik dan pembeli untuk mengakses tutorial aplikasi. Proses dimulai ketika pemilik atau pembeli "Membuka Halaman Profil", yang kemudian direspons oleh sistem dengan "Menampilkan Halaman Profil" tersebut. Dari halaman profil, pemilik dan pembeli memiliki opsi untuk "Memilih Tutorial Aplikasi". Setelah pilihan tersebut dibuat, sistem akan segera "Menampilkan Halaman Panduan Aplikasi", yang menandai berakhirnya alur proses ini.

f. Activity Diagram Tentang Aplikasi

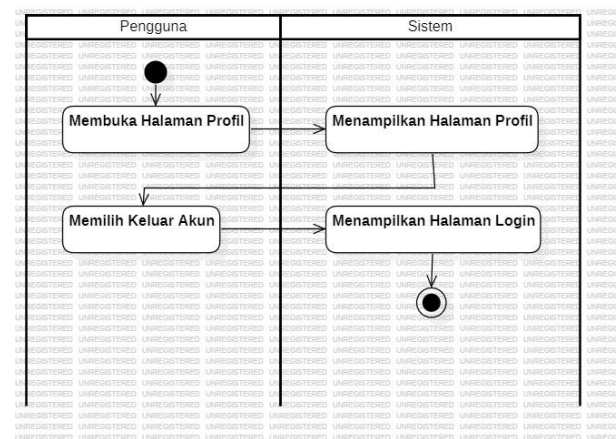


Gambar 3. 19 Activity Diagram Tentang Aplikasi

Diagram aktivitas ini menggambarkan alur singkat bagi pemilik dan pembeli untuk melihat informasi mengenai pengembang aplikasi. Prosesnya

diawali ketika pengguna "Membuka Halaman Profil", yang kemudian akan membuat sistem "Menampilkan Halaman Profil" tersebut. Dari halaman profil, pengguna memiliki opsi untuk "Memilih Tentang pembuat". Setelah pilihan ini dilakukan, sistem akan segera "Menampilkan informasi mengenai pengembang", yang menandai berakhirnya alur proses ini.

g. Activity Diagram Logout



Gambar 3. 20 Activity Diagram Logout

Diagram aktivitas ini mengilustrasikan alur singkat proses keluar akun (logout) dari aplikasi. Dimulai ketika pemilik atau pembeli "Membuka Halaman Profil", yang kemudian direspons oleh sistem dengan "Menampilkan Halaman Profil" tersebut. Dari halaman profil, pemilik atau pembeli dapat "Memilih Keluar akun". Setelah pilihan ini dilakukan, sistem akan segera "Menampilkan Halaman Login" kembali, menandakan bahwa pengguna telah berhasil keluar dari akunnya.

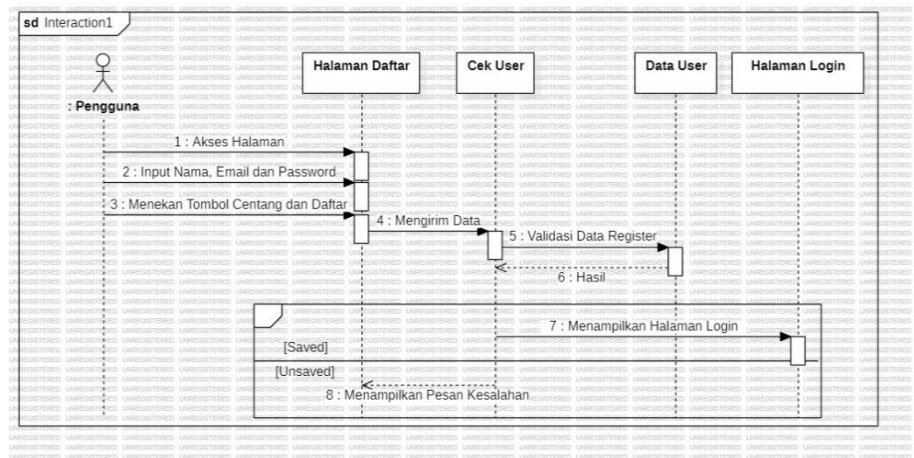
3.2.3 Sequence Diagram

a. Sequence Diagram Login

Gambar 3. 21 *Sequence Diagram Login*

Diagram sekuens ini mengilustrasikan alur interaksi proses *Login* pemilik atau pembeli dengan berbagai komponen sistem. Proses dimulai dengan "Halaman *Login*" yang diakses, kemudian melakukan "*Input* Email dan Password", dan "Menekan Tombol *Login*". Setelah tombol *Login* ditekan, "Halaman *Login*" akan "Mengirim Data" ke komponen "Cek *User*". Komponen "Cek *User*" ini kemudian berinteraksi dengan "Data *User*" untuk melakukan "Validasi Data *Login*". Hasil dari validasi kemudian dikembalikan ke "Cek *User*". Jika validasi berhasil, "Cek *User*" akan mengarahkan ke "Halaman Beranda" untuk "Menampilkan Halaman Beranda", menandai berakhirnya proses *Login*.

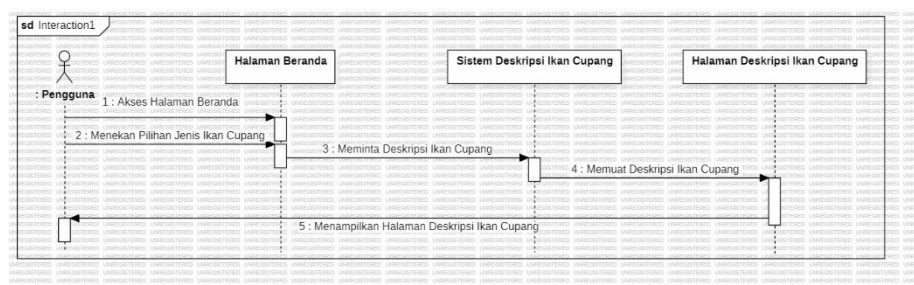
b. Sequence Diagram Register



Gambar 3. 22 Sequence Diagram Register

Diagram sekuens ini menguraikan alur interaksi proses pendaftaran pemilik atau pembeli baru dengan berbagai komponen sistem. Proses dimulai ketika "Akses Halaman Daftar". Selanjutnya, pemilik atau pembeli melakukan "Input Nama, Email dan Password" dan kemudian "Menekan Tombol Centang dan Daftar". Setelah tombol ditekan, "Halaman Daftar" akan "Mengirim Data" ke komponen "Cek User". Komponen "Cek User" ini kemudian berinteraksi dengan "Data User" untuk melakukan "Validasi Data Register". Hasil dari validasi kemudian dikembalikan ke "Cek User". Jika pendaftaran berhasil dan data valid, "Cek User" akan mengarahkan ke "Halaman Login" untuk "Menampilkan Halaman Login", menandai berakhirnya proses registrasi.

c. Sequence Diagram Deskripsi Ikan Cupang

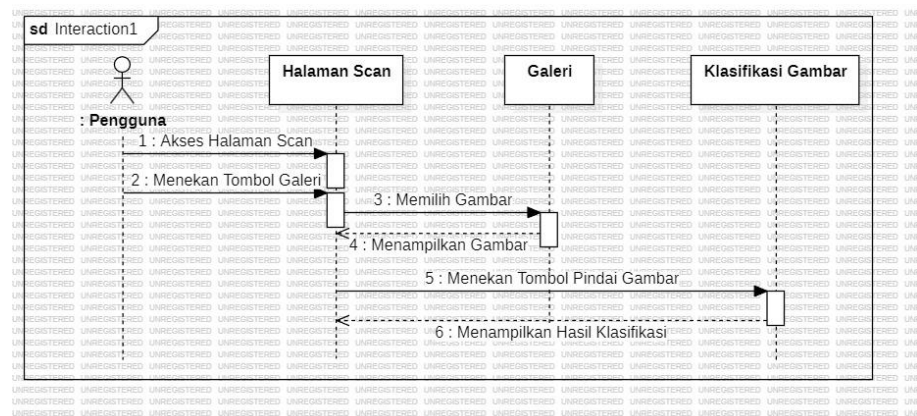


Gambar 3. 23 Sequence Diagram Deskripsi Ikan Cupang

Diagram sekuens ini menggambarkan alur interaksi untuk menampilkan

deskripsi ikan cupang. Proses dimulai ketika pemilik atau pembeli "Akses Halaman Beranda". Dari halaman beranda, kemudian "Menekan Pilihan jenis ikan cupang". Setelah itu, "Halaman Beranda" akan "Meminta Deskripsi Ikan Cupang" dari "Sistem Deskripsi Ikan Cupang". Sistem ini kemudian akan "Memuat Deskripsi Ikan Cupang" dan menampilkannya pada "Halaman Deskripsi Ikan Cupang", menandai selesainya alur proses ini.

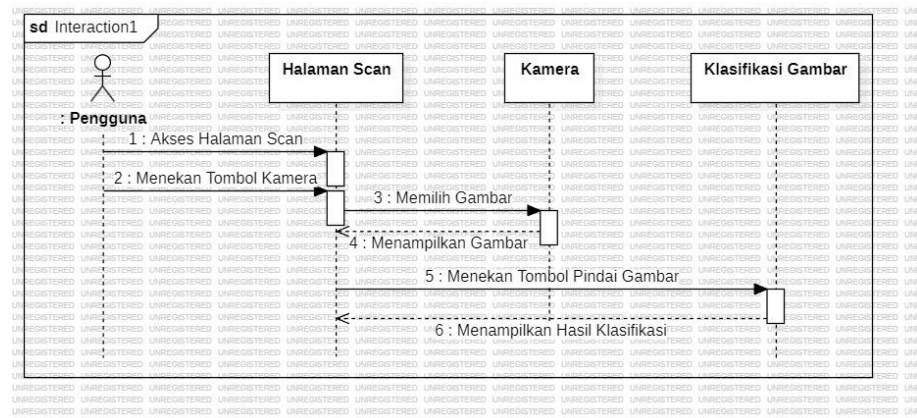
d. *Sequence Diagram Klasifikasi Gambar dari Galeri*



Gambar 3. 24 *Sequence Diagram* Klasifikasi Gambar dari Galeri

Diagram sekuens ini mengilustrasikan alur interaksi untuk melakukan klasifikasi gambar dari galeri. Proses dimulai ketika pemilik atau pembeli "Akses Halaman Scan". Dari halaman scan, kemudian "Menekan Tombol Galeri", yang akan mengarahkan mereka untuk berinteraksi dengan "Galeri". Di dalam Galeri, pemilik atau pembeli "Memilih Gambar" yang ingin diklasifikasikan. Setelah gambar dipilih, pengguna kembali ke "Halaman Scan" dan "Menekan Tombol Pindai Gambar". Aksi ini kemudian memicu proses "Klasifikasi Gambar", yang akan memproses gambar yang dipilih, menandai selesainya alur ini.

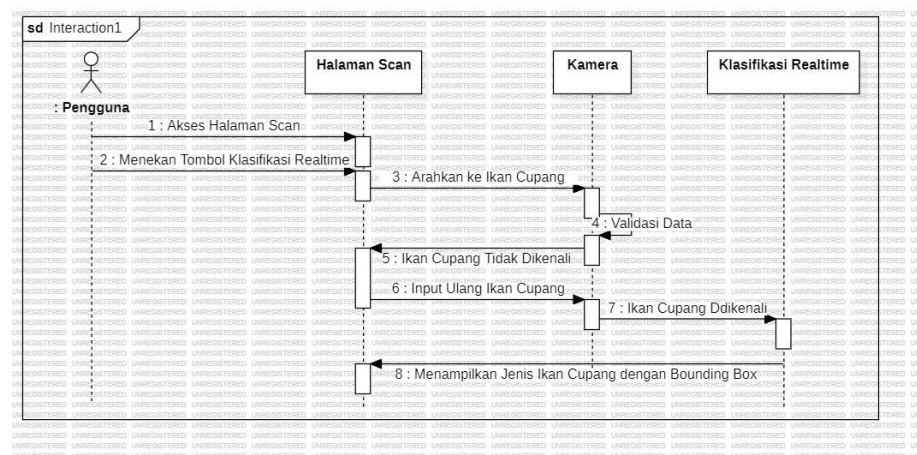
e. *Sequence Diagram* Klasifikasi Gambar dari Kamera



Gambar 3. 25 *Sequence Diagram* Klasifikasi Gambar dari Kamera

Diagram sekuens ini mengilustrasikan alur interaksi untuk melakukan klasifikasi gambar menggunakan kamera. Proses dimulai ketika pemilik atau pembeli "Akses Halaman Scan". Dari halaman scan, kemudian "Menekan Tombol Kamera", yang akan mengarahkan mereka untuk berinteraksi dengan "Kamera". Melalui kamera, pemilik atau pembeli "Mengambil Gambar" yang ingin diklasifikasikan. Setelah gambar diambil, kembali ke "Halaman Scan" dan "Menekan Tombol Pindai Gambar". Aksi ini kemudian memicu proses "Klasifikasi Gambar", yang akan memproses gambar yang diambil, menandai selesainya alur ini.

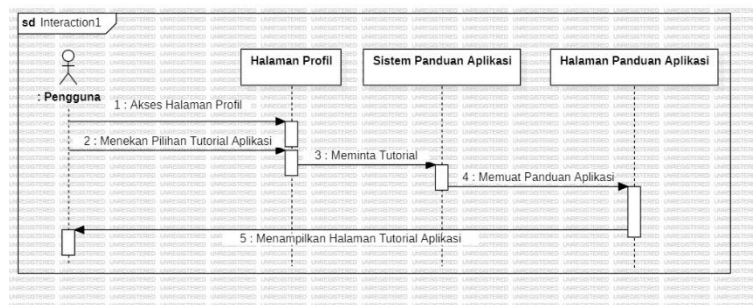
f. *Sequence Diagram* Klasifikasi Realtime



Gambar 3. 26 *Sequence Diagram* Klasifikasi Realtime

Diagram sekuens ini mengilustrasikan alur interaksi untuk melakukan klasifikasi ikan cupang secara *Realtime* menggunakan kamera. Prosesnya dimulai ketika pemilik dan pembeli "Akses Halaman Scan". Dari halaman scan, kemudian "Menekan Tombol Klasifikasi *Realtime*". Setelah itu, interaksi beralih ke "Kamera", di mana sistem akan "Input gambar ikan cupang dengan kamera *Realtime*". Jika pada awalnya ikan cupang belum teridentifikasi, proses mungkin memerlukan "Input ulang ikan cupang" melalui kamera. Setelah ikan cupang berhasil "dikenali" oleh komponen "Klasifikasi *Realtime*", sistem akan "Menampilkan jenis ikan cupang dengan *Bounding box*" di layar, menandai selesainya alur identifikasi *Realtime*.

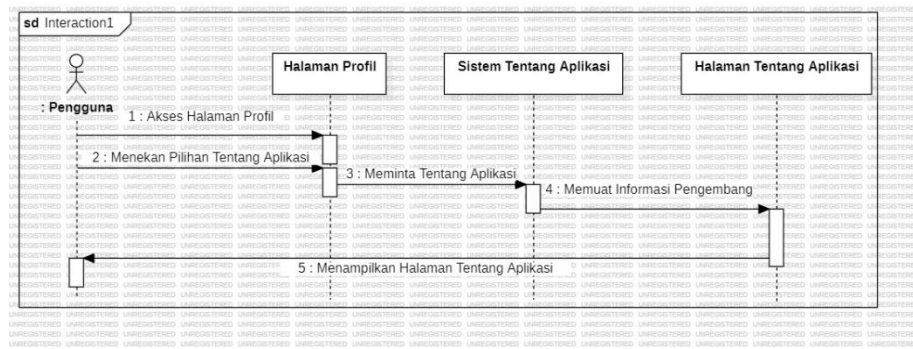
g. Sequence Diagram Tutorial Aplikasi



Gambar 3. 27 Sequence Diagram Tutorial Aplikasi

Diagram sekuens ini mengilustrasikan alur interaksi untuk mengakses tutorial aplikasi. Proses dimulai ketika pemilik dan pembeli "Akses Halaman Profil". Dari halaman profil, kemudian "Menekan Pilihan Tutorial Aplikasi". Setelah itu, "Halaman Profil" akan "Meminta Tutorial" dari "Sistem Panduan Aplikasi". Sistem ini kemudian akan "Memuat Panduan Aplikasi" dan menampilkannya pada "Halaman Panduan Aplikasi", menandai selesainya alur proses ini.

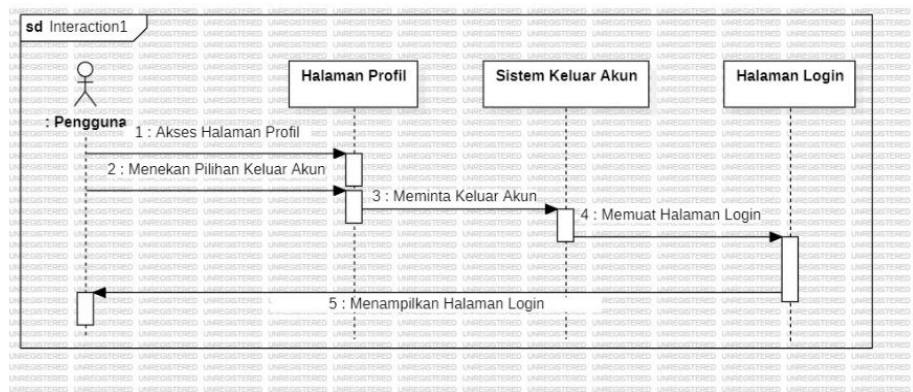
h. Sequence Diagram Tentang Aplikasi



Gambar 3. 28 *Sequence Diagram* Tentang Aplikasi

Diagram sekuens ini menggambarkan alur interaksi untuk mengakses informasi mengenai pengembang aplikasi. Prosesnya dimulai ketika pemilik dan pembeli "Akses Halaman Profil". Dari halaman profil tersebut, selanjutnya "Menekan Pilihan Tentang Aplikasi". Setelah aksi ini, "Halaman Profil" akan "Meminta Tentang Aplikasi" dari "Sistem Tentang Aplikasi". Sistem ini kemudian akan "Memuat Informasi Pengembang" dan menampilkannya pada "Halaman Tentang Aplikasi", menandai selesainya alur interaksi ini.

i. *Sequence Diagram* Logout



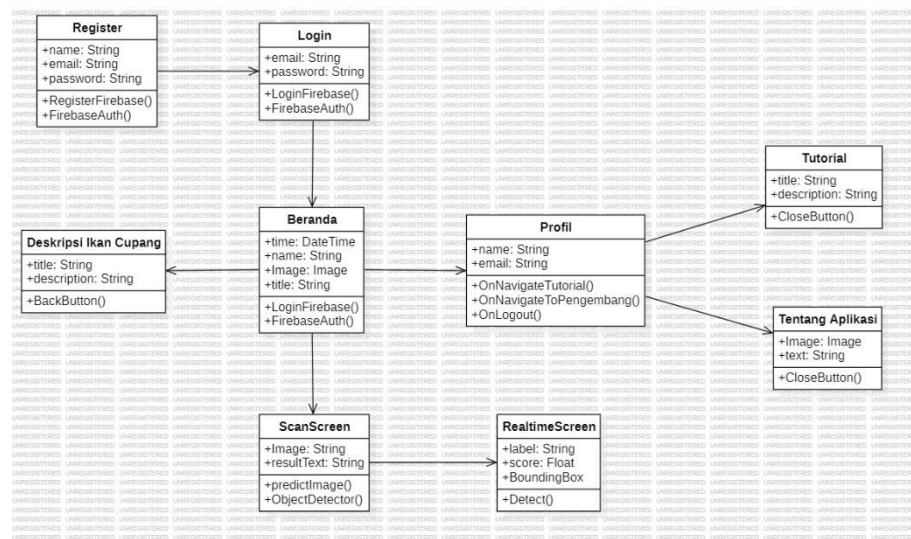
Gambar 3. 29 *Sequence Diagram* Logout

Diagram sekuens ini mengilustrasikan alur interaksi untuk proses keluar akun dari aplikasi. Proses dimulai ketika pemilik atau pembeli "Akses Halaman Profil". Dari halaman profil, pengguna kemudian "Menekan Pilihan Keluar Akun". Setelah itu, "Halaman Profil" akan memicu "Sistem Keluar Akun

Akun" untuk "Memuat Keluar Akun", yang berarti proses logout sedang berlangsung. Setelah berhasil keluar, "Sistem Keluar Akun" akan mengarahkan dan "Memuat Halaman *Login*", menandai selesainya alur proses logout dan pengguna kembali ke layar *Login*.

3.2.4 Class Diagram

Berikut merupakan *Class Diagram* pada sistem yang akan dibangun dalam penelitian:



Gambar 3. 30 Class Diagram

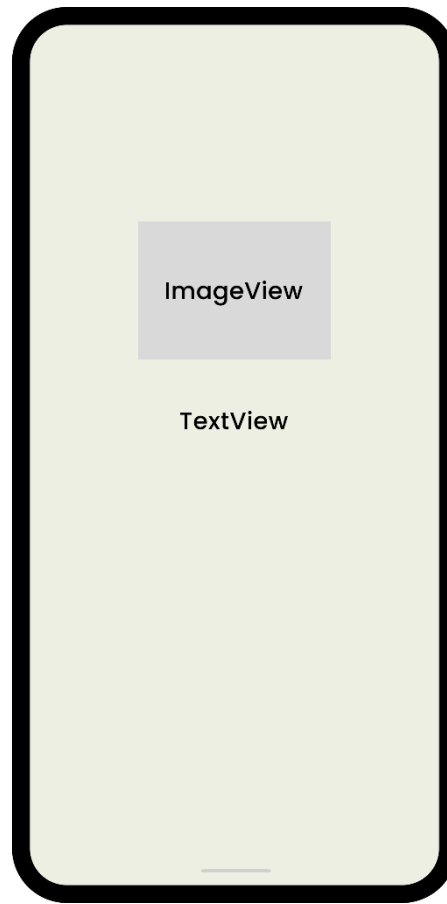
Diagram ini terdapat *Register* dan *Login* yang merupakan layar untuk pendaftaran dan masuk ke aplikasi. Setelah berhasil *Login*, pemilik atau pembeli akan diarahkan ke *Beranda*, yang berfungsi sebagai pusat navigasi utama. Dari *beranda*, dapat mengakses *Deskripsi Ikan Cupang* untuk melihat detail informasi ikan cupang, serta *ScanScreen* dan *RealtimeScreen* yang mengindikasikan fitur pemindaian klasifikasi dan klasifikasi *Realtime* ikan cupang. Selain itu, terdapat akses ke *Profil*, yang mana dari *Profil* ini dapat menuju ke *Tutorial* untuk melihat panduan penggunaan aplikasi, dan *Tentang Aplikasi* untuk melihat informasi mengenai pengembang.

3.3 Perancangan Antarmuka (*Interface Design*)

Perancangan antarmuka mencakup pembuatan rancangan awal berupa sketsa kasar yang menggambarkan struktur tampilan serta fungsionalitas sistem yang diharapkan. Metode ini diterapkan untuk membantu pengembang dalam merancang antarmuka yang bersifat intuitif, efektif, serta efisien dalam mendukung interaksi pengguna dengan sistem (Fatah & Ayu Mufarroha, 2022).

3.3.1 Rancangan Halaman Loading Awal

Halaman Loading merupakan tampilan awal yang muncul secara otomatis ketika aplikasi dijalankan. Tampilan ini berfungsi sebagai pengenalan awal aplikasi, yang menampilkan elemen visual seperti logo aplikasi dan nama aplikasi ikan cupang. Halaman ini digunakan untuk memberikan waktu kepada sistem dalam melakukan proses inisialisasi data atau pemuatan sumber daya sebelum pengguna diarahkan secara otomatis menuju halaman *Login* jika kondisi belum melakukan *Login* dan beranda ketika kondisi sudah melakukan *Login*. Dalam implementasinya, halaman ini ditampilkan dalam durasi singkat, yakni sekitar 2 hingga 3 detik, agar tidak mengganggu pengalaman pengguna secara keseluruhan.



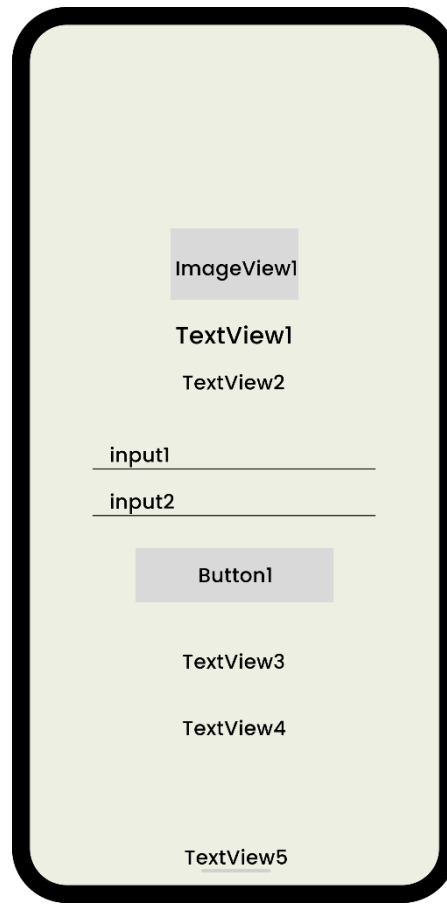
Gambar 3. 31 Rancangan Halaman Loading Awal

Pada halaman loading awal, beberapa komponen yang digunakan adalah sebagai berikut.

- a. *ImageView* : Logo aplikasi
- b. *TextView* : Teks nama aplikasi

3.3.2 Rancangan Halaman *Login*

Halaman *Login* merupakan komponen antarmuka pengguna yang dirancang untuk memberikan akses kepada pengguna ke dalam sistem aplikasi dengan cara memasukkan data autentikasi, seperti alamat email dan kata sandi. Tujuan utama dari perancangan halaman ini adalah untuk memastikan bahwa hanya pengguna yang telah terdaftar dan memiliki kredensial yang valid yang dapat mengakses fitur-fitur yang tersedia dalam aplikasi.



Gambar 3. 32 Rancangan Halaman *Login*

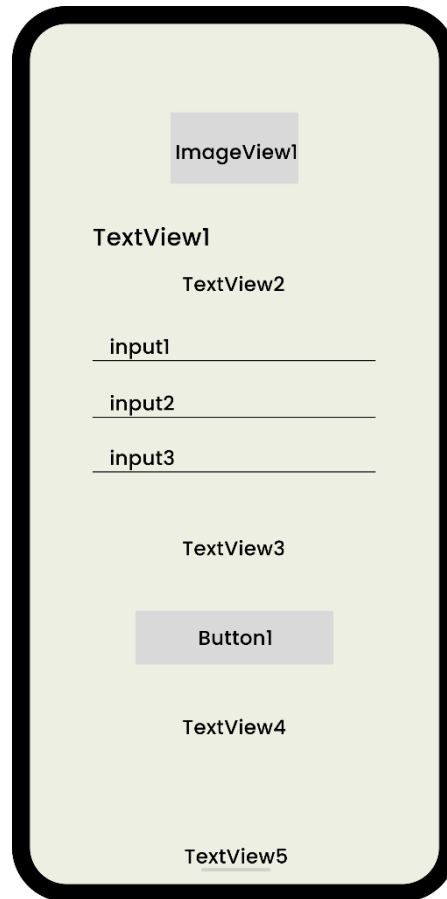
Pada halaman *Login*, beberapa komponen yang digunakan adalah sebagai berikut.

- a. *ImageView1* : Logo aplikasi
- b. *TextView1* : Teks judul halaman *Login*
- c. *TextView2* : Teks “Selamat datang Kembali!”
- d. *Input1* : *Input* email
- e. *Input2* : *Input* password
- f. *Button1* : Tombol untuk masuk ke halaman beranda
- g. *TextView3* : Teks “atau”
- h. *TextView4* : Teks “Belum punya akun? Daftar”
- i. *TextView5* : Teks versi aplikasi

3.3.3 Rancangan Halaman *Register*

Halaman registrasi merupakan bagian dari rancangan antarmuka yang disediakan untuk memungkinkan pengguna baru melakukan proses

pendaftaran atau pembuatan akun pada aplikasi. Halaman ini berperan sebagai tahapan awal yang harus dilalui oleh pengguna sebelum dapat mengakses berbagai layanan dan fitur yang tersedia di dalam sistem aplikasi.



Gambar 3. 33 Rancangan Halaman *Register*

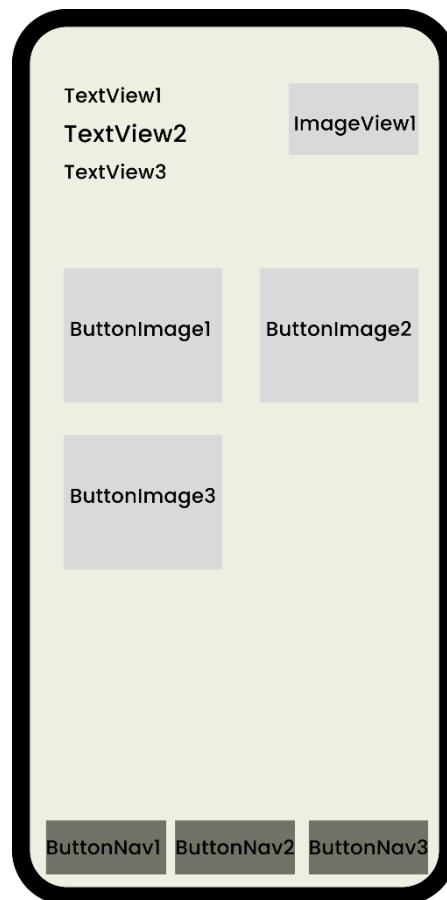
Pada halaman registrasi, beberapa komponen yang digunakan adalah sebagai berikut.

- a. *ImageView1* : Logo aplikasi
- b. *TextView1* : Teks daftar akun
- c. *TextView2* : Teks “Buat akun baru untuk memulai”
- d. *Input1* : *Input* nama
- e. *Input2* : *Input* email
- f. *Input3* : *Input* password
- g. *TextView3* : Teks “Saya setuju dengan ketentuan berlaku”
- h. *Button1* : Tombol untuk mendaftar
- i. *TextView4* : Teks “Sudah punya akun? Masuk”

j. `TextView5` : Teks versi aplikasi

3.3.4 Rancangan Halaman Beranda

Halaman beranda merupakan tampilan awal yang disajikan kepada pengguna setelah berhasil masuk ke dalam aplikasi. Pada halaman ini, ditampilkan informasi umum mengenai berbagai jenis ikan cupang yang bertujuan untuk memberikan edukasi dan pengetahuan dasar kepada pengguna.



Gambar 3. 34 Rancangan Halaman Beranda

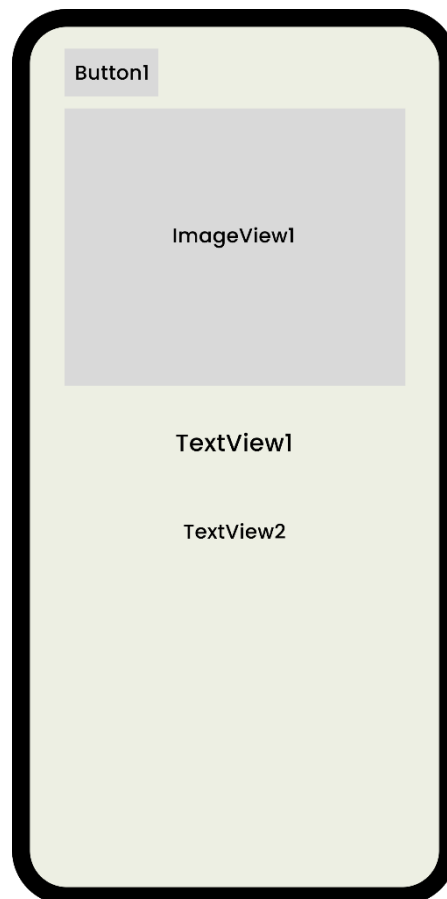
Pada halaman beranda, beberapa komponen yang digunakan adalah sebagai berikut.

- a. `TextView1` : Teks salam
- b. `TextView2` : Nama pemilik akun
- c. `TextView3` : Teks “Temukan ikan cupang favorit”
- d. `ImageView1` : Logo Profile Default

- e. *ButtonImage1* : Beralih ke halaman deskripsi *Halfmoon*
- f. *ButtonImage2* : Beralih ke halaman deskripsi *Plakat*
- g. *ButtonImage3* : Beralih ke halaman deskripsi *Serit*
- h. *ButtonNav1* : Beralih ke menu beranda
- i. *ButtonNav2* : Beralih ke menu pindai
- j. *ButtonNav3* : Beralih ke menu profile

3.3.5 Rancangan Halaman Deskripsi Ikan Cupang

Halaman deskripsi ikan cupang merupakan halaman ketika pengguna menekan salahsatu jenis ikan yang berada di beranda. Pada halaman ini, ditampilkan deskripsi mengenai jenis dari ikan tersebut.



Gambar 3. 35 Rancangan Halaman Deskripsi Ikan Cupang

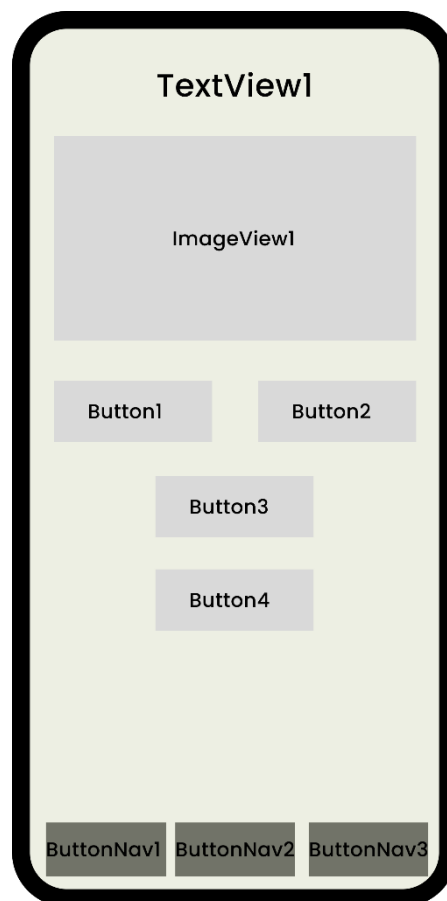
Pada halaman deskripsi, beberapa komponen yang digunakan adalah sebagai berikut.

- a. *Button1* : Tombol untuk kembali ke halaman beranda

- b. *ImageView1* : Gambar jenis ikan cupang
- c. *TextView1* : Jenis ikan cupang
- d. *TextView2* : Deskripsi jenis ikan cupang

3.3.6 Rancangan Halaman Scan

Halaman scan merupakan fitur utama yang dirancang dalam aplikasi, yang difokuskan untuk melakukan proses identifikasi dan klasifikasi terhadap jenis ikan cupang. Tujuan dari perancangan halaman ini adalah untuk menyajikan informasi secara cepat dan akurat berdasarkan gambar ikan cupang yang diunggah atau diambil langsung oleh pengguna melalui perangkat.



Gambar 3. 36 Rancangan Halaman Scan

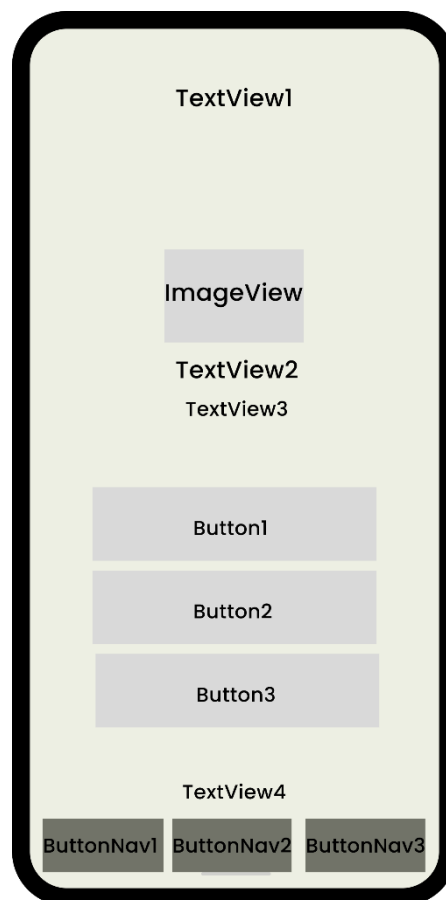
Pada halaman scan, beberapa komponen yang digunakan adalah sebagai berikut.

- a. *TextView1* : Teks “Pindai Ikan Cupang”
- b. *ImageView1* : Menampilkan gambar yang akan dipindai

- c. Button1 : Tombol untuk *input* gambar dari galeri
- d. Button2 : Tombol untuk *input* foto dari kamera
- e. Button3 : Tombol untuk memindai gambar
- f. Button4 : Tombol untuk klasifikasi *Realtime*
- g. ButtonNav1 : Beralih ke menu beranda
- h. ButtonNav2 : Beralih ke menu pindai
- i. ButtonNav3 : Beralih ke menu profile

3.3.7 Rancangan Halaman Profil

Halaman profil merupakan bagian dari antarmuka aplikasi yang dirancang untuk menampilkan informasi pengguna, seperti nama dan alamat email. Selain itu, melalui halaman ini pengguna dapat melihat panduan penggunaan aplikasi (tutorial), informasi mengenai pengembang aplikasi, serta melakukan proses keluar (logout) dari akun yang sedang digunakan.



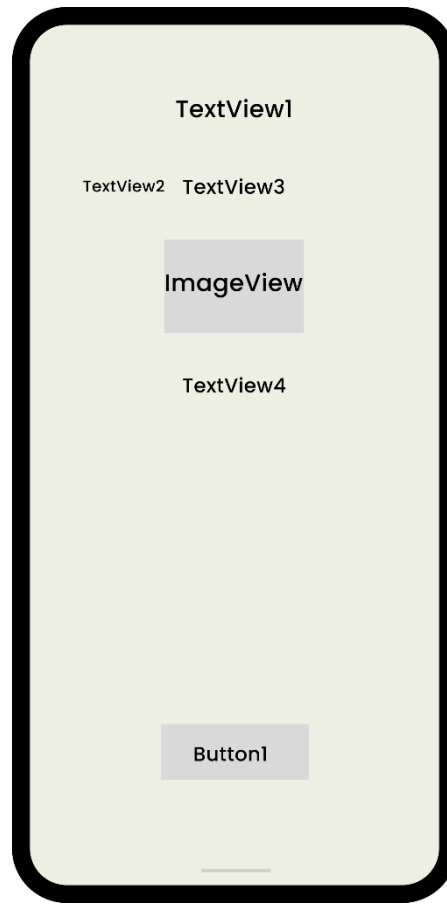
Gambar 3. 37 Rancangan Halaman Profil

Pada halaman profil, beberapa komponen yang digunakan adalah sebagai berikut.

- a. `TextView1` : Teks judul halaman profile
- b. `ImageView1` : Logo profile default
- c. `TextView2` : Nama pemilik akun
- d. `TextView3` : Email pemilik akun
- e. `Button1` : Tombol ke halaman tutorial
- f. `Button2` : Tombol ke halaman developer
- g. `Button3` : Tombol keluar akun
- h. `TextView4` : Teks versi aplikasi
- i. `ButtonNav1` : Beralih ke menu beranda
- j. `ButtonNav2` : Beralih ke menu pindai
- k. `ButtonNav3` : Beralih ke menu profile

3.3.8 Rancangan Halaman Tutorial

Halaman tutorial merupakan bagian dari antarmuka aplikasi yang dirancang untuk menampilkan cara menggunakan scan ikan cupang, membaca hasil scan, eksplorasi katalog ikan, pengaturan profil, dan tips pemeliharaan cupang.



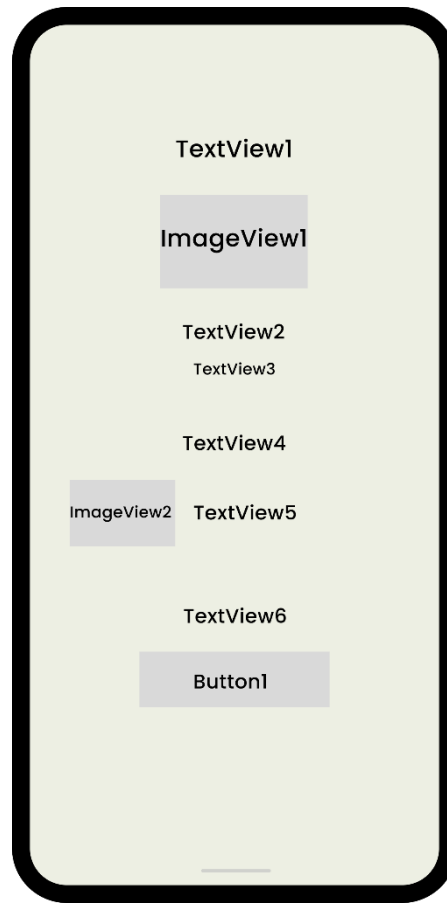
Gambar 3. 38 Rancangan Halaman Tutorial

Pada halaman tutorial, beberapa komponen yang digunakan adalah sebagai berikut.

- a. `TextView1` : Teks “Tutorial Aplikasi”
- b. `TextView2` : Teks nomor tutorial
- c. `TextView3` : Teks nama tutorial
- d. `ImageView` : Logo aplikasi
- e. `TeksView4` : Teks tutorial
- f. `Button1` : Tombol untuk menutup halaman tutorial

3.3.9 Rancangan Halaman Tentang Aplikasi

Halaman developer merupakan bagian dari antarmuka aplikasi yang dirancang untuk menampilkan informasi pengembang aplikasi.



Gambar 3. 39 Rancangan Halaman Tentang Aplikasi

Pada halaman developer, beberapa komponen yang digunakan adalah sebagai berikut.

- a. TextView1 : Teks “Tentang Aplikasi”
- b. ImageView1 : Logo Aplikasi
- c. TextView2 : Teks nama aplikasi
- d. TextView3 : Teks versi aplikasi
- e. TextView4 : Teks “Dikembangkan oleh:”
- f. ImageView2 : Foto peneliti
- g. TextView5 : Nama peneliti
- h. TextView6 : Teks “2025 FKOM. All rights reserved”
- i. Button1 : Tombol untuk menutup halaman developer