KERJA PRAKTIK – IF234603

Integrasi dan Pengembangan Fitur pada Aplikasi ZooSite

Lab Manajemen Cerdas Informasi, Teknik Informatika, ITS
Jl. Teknik Kimia, Keputih, Kec. Sukolilo, Surabaya, Jawa Timur 60111, Indonesia
Periode: 1 April 2026 - 30 Juni 2026

Oleh:

Giandra Raihan Nayottama 5025221094
Joanes Don Bosco B 5025221111

Pembimbing Departemen
Ratih Nur Esti Anggraini, S.Kom., M.Sc., Ph.D
Pembimbing Lapangan
Dr. Ir. Adhatus Solichah Ahmadiyah, S.Kom., M.Sc.

DEPARTEMEN TEKNIK INFORMATIKA
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2026

[Halaman ini sengaja dikosongkan]

KERJA PRAKTIK – IF234603

Integrasi dan Pengembangan Fitur pada Aplikasi ZooSite

Lab Manajemen Cerdas Informasi, Teknik Informatika, ITS
Jl. Teknik Kimia, Keputih, Kec. Sukolilo, Surabaya, Jawa Timur 60111, Indonesia
Periode: 1 April 2026 - 30 Juni 2026

Oleh:

Giandra Raihan Nayottama 5025221095
Joanes Don Bosco B 5025221111

Pembimbing Departemen
Ratih Nur Esti Anggraini, S.Kom., M.Sc., Ph.D
Pembimbing Lapangan
Dr. Ir. Adhatus Solichah Ahmadiyah, S.Kom., M.Sc.

DEPARTEMEN TEKNIK INFORMATIKA
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2026

[Halaman ini sengaja dikosongkan]

DAFTAR ISI

DAFTAR ISI **V**
DAFTAR GAMBAR **IX**
DAFTAR TABEL **XI**
DAFTAR KODE SUMBER **XIII**
LEMBAR PENGESAHAN **XV**
ABSTRAK **XVII**
KATA PENGANTAR **XIX**
BAB I PENDAHULUAN **1**
1.1 Latar Belakang 1
1.2 Tujuan 3
1.3 Manfaat 3
1.4 Rumusan Masalah 4
1.5 Lokasi dan Waktu Kerja Praktik 4
1.6 Metodologi Kerja Praktik 4
1.6.1 Perumusan Masalah *5*
1.6.2 Studi Literatur *5*
1.6.3 Analisis dan Perancangan *5*
1.6.4 Implementasi Sistem *5*
1.6.5 Pengujian dan Evaluasi *5*
1.6.6 Kesimpulan dan Saran *6*
1.7 Sistematika Laporan 6
1.7.1 Bab I Pendahuluan *6*
1.7.2 Bab II Profil Perusahaan *6*
1.7.3 Bab III Tinjauan Pustaka *6*
1.7.4 Bab IV Analisis dan Perancangan Infrastruktur Sistem *7*
1.7.5 Bab V Implementasi Sistem *7*
1.7.6 Bab VI Pengujian dan Evaluasi *7*

1.7.7 Bab VII Kesimpulan dan Saran *7*
BAB II PROFIL PERUSAHAAN **9**
2.1 Profil Laboratorium Manajemen Informasi 9
2.2 Lokasi 9
BAB III TINJAUAN PUSTAKA **11**
3.1 Flutter dan Dart 11
3.2 Clean Architecture 11
3.3 Manajemen State: BLoC dan Provider 12
3.4 Convolutional Neural Network (CNN) 12
3.5 Transfer Learning 13
3.6 TensorFlow Lite 14
3.7 FastAPI dan Model YAMNet 14
3.8 Universal Sentence Encoder, Word Embedding, dan TF-IDF 15
3.9 Google Maps Flutter 15
BAB IV ANALISIS DAN PERANCANGAN INFRASTRUKTUR SISTEM **17**
4.1 Analisis Sistem 17
4.1.1 Definisi Umum Aplikasi *17*
4.1.2 Analisis Proyek Sebelumnya *17*
4.2 Perancangan Infrastruktur Sistem 20
4.2.1 Desain Arsitektur Sistem *20*
4.2.2 Proses Penggabungan Modul *21*
BAB V IMPLEMENTASI SISTEM **25**
5.1 Implementasi Infrastruktur Plugin 25
5.1.1 Perbaikan Fitur Klasifikasi Ikan *27*
5.2 Klasifikasi Suara Hewan 28
5.3 Chatbot ZooBot Assistant 32
5.4 Pemilihan Kebun Binatang dan Peta 38

5.5 Pembaruan Antarmuka Halaman Utama 39
BAB VI PENGUJIAN DAN EVALUASI 43
6.1 Tujuan Pengujian 43
6.2 Kriteria Pengujian 43
6.3 Skenario Pengujian 45
6.4 Evaluasi Pengujian 47
BAB VII KESIMPULAN DAN SARAN 53
7.1 Kesimpulan 53
7.2 Saran 53
DAFTAR PUSTAKA 5

[Halaman ini sengaja dikosongkan]

DAFTAR GAMBAR

Gambar 4.1 Ilustrasi Penggabungan 23
Gambar 5.1 Halaman Klasifikasi Hewan (kiri) dan Klasifikasi Ikan (kanan) 26
Gambar 5.2 Halaman Fitur Pengenalan Suara 29
Gambar 5.3 Tampilan Chatbot ZooBot Assistant 35
Gambar 5.4 Alur Keputusan Pencocok Hibrida (Hybrid Text Matcher) pada Chatbot 37
Gambar 5.5 Peta Lokasi Kebun Binatang Surabaya (KBS) 38
Gambar 5.6 Tampilan Galeri Satwa (Wild Gallery) 41
Gambar 6.1 Pengujian Fitur Klasifikasi Hewan 48
Gambar 6.2 Pengujian Fitur Klasifikasi Ikan 49
Gambar 6.3 Pengujian Fitur Pengenalan Suara 50
Gambar 6.4 Peta Lokasi Kebun Binatang Bandung setelah Pergantian Pilihan Kota 51

[Halaman ini sengaja dikosongkan]

DAFTAR TABEL

Tabel 4.1 Analisis Proyek Sebelumnya 19
Tabel 4.2 Fitur Aplikasi Tahap Akhir 20
Tabel 4.3 Pembagian Tugas 21
Tabel 5.1 Perbandingan Fitur Chatbot 34
Tabel 6.1 Kriteria Pengujian 44
Tabel 6.2 Skenario dan Hasil yang Diharapkan 45
Tabel 6.3 Aspek Pengujian Klasifikasi 46
Tabel 6.4 Hasil Evaluasi Pengujian 52

[Halaman ini sengaja dikosongkan]

DAFTAR KODE SUMBER

Kode Sumber 5.1 Pengaktifan model ikan pada home_header_view.dart (perbaikan fitur ikan) 28
Kode Sumber 5.2 Inferensi suara on-device dengan TFLite 30
Kode Sumber 5.3 Konfigurasi endpoint API suara pada constants.dart 31
Kode Sumber 5.4 Pengiriman audio ke REST API 32
Kode Sumber 5.5 Endpoint /classify pada backend FastAPI (main.py) 32
Kode Sumber 5.6 Pencocok hibrida chatbot (hybrid_text_matcher.dart) 36
Kode Sumber 5.7 Penambahan rute chatbot dan pemilihan kebun binatang (router_main.dart) 39
Kode Sumber 5.8 Perubahan tata letak halaman utama (home_view.dart) 40

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN
KERJA PRAKTIK

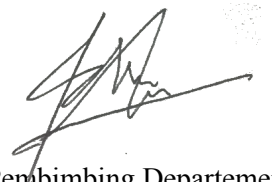
Integrasi dan Pengembangan Fitur pada Aplikasi ZooSite

Oleh:

Giandra Raihan Nayottama	5025221094
Joanes Don Bosco B	5025221111

Disetujui oleh Pembimbing Kerja Praktik:

1. Ratih Nur Esti Anggraini, S.Kom., M.Sc., Ph.D.
   NIP. 198412102014042003



(Pembimbing Departemen)

2. Dr. Ir. Adhatus Solichah Ahmadiyah, S.Kom., M.Sc.
   NIP. 198508262015042002



(Pembimbing Lapangan)

[Halaman ini sengaja dikosongkan]

Integrasi dan Pengembangan Fitur pada Aplikasi ZooSite

Nama Mahasiswa	: Giandra Raihan Nayottama
NRP	: 5025221094
Nama Mahasiswa	: Joanes Don Bosco B
NRP	: 5025221111
Departemen	: Teknik Informatika FTEIC-ITS
Pembimbing Departemen	: Ratih Nur Esti Anggraini, S.Kom., M.Sc., Ph.D
Pembimbing Lapangan	: Dr. Ir. Adhatus Solichah Ahmadiyah, S.Kom., M.Sc.

ABSTRAK

ZooSite merupakan aplikasi mobile berbasis Flutter yang dirancang untuk membantu pengunjung kebun binatang, khususnya Kebun Binatang Surabaya (KBS) dan kebun binatang di Bandung, dalam menjelajahi dan mengenali koleksi satwa secara mandiri. Kerja praktik ini berfokus pada integrasi dan pengembangan sejumlah fitur. Pekerjaan dilakukan selama kurang lebih tiga bulan oleh dua orang mahasiswa dengan pembagian tugas yang jelas.

Fitur yang dikembangkan meliputi: (1) chatbot "ZooBot Assistant" yang menjawab pertanyaan pengunjung menggunakan pendekatan pencocokan teks hibrida (word embedding dan TF-IDF); (2) fitur pemilihan kebun binatang beserta peta lokasi berbasis Google Maps; (3) klasifikasi suara hewan yang berjalan langsung di perangkat (on-device) menggunakan TensorFlow Lite, dilengkapi alternatif REST API berbasis FastAPI dan model YAMNet; (4) klasifikasi gambar hewan dan ikan menggunakan TensorFlow Lite di sisi perangkat, termasuk perbaikan fitur klasifikasi ikan yang sebelumnya tidak berfungsi karena model

ikan tidak pernah diaktifkan; serta (5) pembaruan antarmuka dan tata letak halaman utama. Hasil pengujian menunjukkan seluruh fitur berhasil diintegrasikan dan berjalan sesuai skenario yang dirancang.

Kata Kunci: Flutter, Kecerdasan Buatan, Chatbot, Klasifikasi Suara, Klasifikasi Gambar, Kebun Binatang.

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Allah SWT atas penyertaan dan karunia-Nya sehingga penulis dapat menyelesaikan salah satu kewajiban penulis sebagai mahasiswa Departemen Teknik Informatika ITS yaitu Kerja Praktik yang berjudul: Integrasi dan Pengembangan Fitur pada Aplikasi ZooSite

Penulis menyadari bahwa masih banyak kekurangan baik dalam melaksanakan kerja praktik maupun penyusunan buku laporan kerja praktik ini. Namun penulis berharap buku laporan ini dapat menambah wawasan pembaca dan dapat menjadi sumber referensi.

Melalui buku laporan ini penulis juga ingin menyampaikan rasa terima kasih kepada orang-orang yang telah membantu menyusun laporan kerja praktik baik secara langsung maupun tidak langsung antara lain:

1. Kedua orang tua penulis.
2. Ibu Ratih Nur Esti Anggraini, S.Kom., M.Sc., Ph.D. selaku dosen pembimbing kerja praktik sekaligus koordinator kerja praktik.
3. Ibu Dr. Ir. Adhatus Solichah Ahmadiyah, S.Kom., M.Sc. selaku pembimbing lapangan selama kerja praktik berlangsung.
4. Teman-teman penulis yang senantiasa memberikan semangat ketika penulis melaksanakan KP.

Taiwan, 22 Juli 2026

Giandra Raihan Nayottama dan Joanes Don Bosco B

[Halaman ini sengaja dikosongkan]

BAB I
PENDAHULUAN

1.1. Latar Belakang

Kebun binatang merupakan salah satu sarana rekreasi sekaligus edukasi yang menyimpan beragam koleksi satwa. Informasi mengenai setiap satwa pada umumnya hanya tersedia pada papan nama di depan kandang, sehingga cakupannya terbatas dan seringkali tidak cukup untuk menjawab rasa ingin tahu pengunjung. Akibatnya, pengunjung kesulitan memperoleh informasi yang lebih lengkap dan interaktif mengenai satwa yang mereka lihat. Perkembangan teknologi mobile dan kecerdasan buatan membuka peluang untuk menghadirkan pengalaman menjelajah kebun binatang yang lebih menarik dan informatif melalui perangkat pintar. Peluang inilah yang mendasari pengembangan aplikasi ZooSite sebagai media bantu pengunjung kebun binatang.

Aplikasi ZooSite bukan dibangun dari nol, melainkan merupakan hasil pengembangan berkelanjutan dari beberapa Tugas Akhir (TA) mahasiswa terdahulu, yang masing-masing telah menghasilkan satu modul dengan capaiannya sendiri. Fitur chatbot berawal dari penelitian Cikanindi (2021), yang menghasilkan location-aware chatbot berbasis Universal Sentence Encoder (USE) dan Firebase Realtime Database dalam bentuk aplikasi Android terpisah. Fitur klasifikasi hewan dikembangkan oleh Kusumo (2023) menggunakan Convolutional Neural Network (CNN) dan berhasil ditingkatkan akurasinya oleh Auliya (2024) melalui pendekatan transfer learning. Fitur klasifikasi ikan dikembangkan oleh Fuadah (2025) menggunakan transfer learning dengan cakupan 26 jenis ikan, sedangkan fitur pengenalan suara hewan dikembangkan oleh Saputra (2025) menggunakan CNN yang dilatih untuk mengenali suara sejumlah hewan koleksi kebun binatang. Setiap modul tersebut terbukti dapat menghasilkan keluaran (output)

sesuai rancangan TA masing-masing pada saat diujikan secara terpisah.

Meskipun setiap modul telah berhasil pada tahap TA masing-masing, terdapat sejumlah kekurangan (gap) yang teridentifikasi ketika seluruh modul hendak digunakan sebagai satu aplikasi utuh. Pertama, kelima modul tersebut dibangun secara independen dengan teknologi yang berbeda-beda (Java dan Chaquopy untuk chatbot, Flutter/TFLite untuk klasifikasi gambar dan suara), sehingga belum tergabung ke dalam satu basis kode aplikasi. Kedua, modul chatbot versi awal bergantung pada koneksi internet serta layanan Firebase dan model USE yang diunduh secara daring, dan pada saat kerja praktik ini dimulai layanan tersebut sudah tidak dapat diakses (domain telah kedaluwarsa) sehingga chatbot versi awal tidak dapat dijalankan sama sekali. Ketiga, pada basis kode awal aplikasi ZooSite (ZooSiteApp-original), fitur klasifikasi ikan yang seharusnya sudah tersedia ternyata tidak berfungsi karena model ikan tidak pernah diaktifkan pada alur pemilihan model. Keempat, fitur pemilihan kebun binatang beserta peta lokasi, serta antarmuka halaman utama yang mampu menampung seluruh fitur secara rapi, juga belum tersedia pada basis kode awal tersebut.

Berdasarkan kekurangan-kekurangan (gap) tersebut, kerja praktik ini difokuskan untuk mengembangkan dan mengintegrasikan seluruh modul warisan menjadi satu aplikasi ZooSite yang utuh (ZooSiteApp-main). Chatbot ZooBot Assistant dibangun ulang sebagai modul Flutter yang berjalan sepenuhnya di perangkat (offline) menggunakan pencocokan hibrida word embedding dan TF-IDF, sehingga tidak lagi bergantung pada Firebase maupun koneksi internet. Fitur klasifikasi ikan diperbaiki dengan mengaktifkan pemanggilan model yang sesuai sebelum proses klasifikasi dijalankan. Fitur pemilihan kebun binatang beserta peta lokasi berbasis Google Maps, serta pembaruan antarmuka halaman utama, dikembangkan dan diintegrasikan ke dalam basis kode yang

sama mengikuti struktur clean architecture berbasis fitur, sehingga penambahan fitur baru tidak mengganggu fitur yang sudah berjalan.

Pekerjaan kerja praktik ini dilakukan selama kurang lebih tiga bulan dan dibagi kepada dua mahasiswa untuk mempercepat pengembangan sekaligus memberikan pembagian tanggung jawab yang jelas atas masing-masing fitur, sebagaimana diuraikan pada Bab IV.

1.2. Tujuan

Tujuan kerja praktik ini adalah mengintegrasikan dan mengembangkan fitur berbasis kecerdasan buatan pada aplikasi ZooSite, meliputi chatbot, klasifikasi suara, klasifikasi gambar, serta fitur pemilihan kebun binatang dan peta lokasi.

1.3. Manfaat

Manfaat yang diperoleh dengan adanya pengembangan aplikasi ZooSite antara lain adalah mempermudah pengunjung kebun binatang dalam mengenali satwa secara mandiri tanpa harus bergantung pada papan informasi di depan kandang. Fitur klasifikasi gambar memungkinkan pengguna mengidentifikasi jenis hewan maupun ikan hanya dengan memotret atau memilih gambar dari galeri, lalu memperoleh nama, nama latin, deskripsi, dan fakta uniknya. Fitur pengenalan suara membantu pengguna mengenali hewan melalui rekaman suara yang diproses langsung di perangkat menggunakan TensorFlow Lite, sehingga dapat digunakan tanpa koneksi internet. Chatbot ZooBot Assistant memudahkan pengunjung memperoleh informasi seputar kebun binatang, seperti jam buka, tiket, dan koleksi hewan, secara interaktif. Fitur pemilihan kebun binatang beserta peta lokasi memungkinkan konten aplikasi menyesuaikan kota yang dipilih dan membantu pengguna menemukan lokasi kebun binatang. Selain itu, arsitektur berbasis fitur (clean architecture) membuat

aplikasi lebih mudah dikembangkan dan ditambah fitur baru tanpa mengganggu fitur yang sudah ada.

1.4. Rumusan Masalah

Berdasarkan latar belakang di atas, rumusan masalah dari kerja praktik ini disusun agar setiap poin dapat dikerjakan secara konkret dan jawabannya dapat ditunjukkan pada Bab VII (Kesimpulan), sebagai berikut:

1. Bagaimana mengintegrasikan modul chatbot, klasifikasi suara, serta pemilihan kebun binatang dan peta ke dalam satu basis kode aplikasi ZooSite?
2. Bagaimana memperbaiki fitur klasifikasi ikan yang sebelumnya tidak dapat berfungsi?
3. Bagaimana memperbarui antarmuka halaman utama agar dapat menampung seluruh fitur baru secara rapi?

1.5. Lokasi dan Waktu Kerja Praktik

Lokasi pelaksanaan kerja praktik ini dilakukan secara remote karena kedua penulis sedang menjalani program *Double Degree* di Taiwan. Meskipun kegiatan dilakukan dari jarak jauh, seluruh proses kerja praktik tetap berjalan dengan baik melalui koordinasi daring bersama pembimbing lapangan dan tim pengembang proyek.

Adapun kerja praktik dimulai pada tanggal 1 April 2026 hingga 30 Juni 2026.

1.6. Metodologi Kerja Praktik

Metodologi dalam pembuatan buku kerja praktik meliputi:

1.6.1. Perumusan Masalah

Identifikasi kebutuhan dilakukan dengan menelaah basis kode awal (ZooSiteApp-original) untuk menentukan fitur yang

belum terintegrasi dan bagian yang belum berfungsi, kemudian mendiskusikan pembagian tugas dan target akhir bersama dosen lapangan kami.

1.6.2. Studi Literatur

Kedua penulis mempelakari framework Flutter dan bahasa Dart, pola arsitektur clean architecture, manajemen state menggunakan BLoC dan Provider, TensorFlow Lite untuk inferensi model di perangkat, FastAPI dan model YAMNet untuk klasifikasi suara, Google Maps Flutter, serta teknik pencocokan teks (word embedding dan TF-IDF) untuk chatbot.

1.6.3. Analisis dan Perancangan Sistem

Analisis kebutuhan tiap fitur dilakukan dan dituangkan ke dalam rancangan modul mengikuti struktur clean architecture yang telah ada, sehingga setiap fitur baru menempati folder fitur (feature) tersendiri dengan lapisan data, domain, dan presentation.

1.6.4. Implementasi Sistem

Implementasi dilakukan secara paralel oleh dua mahasiswa sesuai pembagian tugas, kemudian diintegrasikan ke dalam satu basis kode utama (ZooSiteApp-main) melalui penyesuaian dependency injection, routing, dan penyediaan aset.

1.6.5. Pengujian dan Evaluasi

Pengujian dilakukan secara fungsional pada perangkat/emulator Android untuk memastikan setiap fitur berjalan sesuai skenario, meliputi klasifikasi gambar, klasifikasi ikan, klasifikasi suara, chatbot, serta peta dan galeri.

1.6.6. Kesimpulan dan Saran

Setelah seluruh modul berhasil digabungkan menjadi satu aplikasi, dilakukan pengujian fungsional untuk memastikan setiap fitur berjalan dengan benar. Pengujian meliputi fitur klasifikasi gambar hewan dan ikan, pengenalan suara, chatbot, serta pemilihan kebun binatang dan peta. Evaluasi juga dilakukan dengan menjalankan aplikasi secara langsung pada perangkat/emulator Android dan menguji alur kerja mulai dari pengambilan gambar maupun perekaman suara hingga menampilkan hasil klasifikasi, serta interaksi tanya-jawab dengan chatbot. Setelah itu, hasil pekerjaan ditinjau bersama dosen pembimbing lapangan untuk memastikan aplikasi sudah berjalan sesuai dengan yang direncanakan. Dari hasil pengujian dan evaluasi tersebut kemudian ditarik kesimpulan serta disusun saran untuk pengembangan aplikasi selanjutnya.

1.7. Sistematika Laporan

1.7.1. Bab I Pendahuluan

Berisi latar belakang, tujuan, manfaat, rumusan masalah, lokasi dan waktu kerja praktik, metodologi, serta sistematika laporan.

1.7.2. Bab II Profil Perusahaan

Berisi gambaran umum instansi/perusahaan tempat pelaksanaan kerja praktik.

1.7.3. Bab III Tinjauan Pustaka

Berisi dasar teori teknologi yang digunakan dalam menyelesaikan proyek kerja praktik.

1.7.4. Bab IV Analisis dan Perancangan Infrastruktur Sistem

Berisi analisis kebutuhan dan perancangan solusi/sistem yang dikerjakan selama kerja praktik.

1.7.5. Bab V Implementasi Sistem

Berisi uraian tahapan implementasi tiap fitur aplikasi beserta kode sumber terkait.

1.7.6. Bab VI Pengujian dan Evaluasi

Berisi uraian tahapan implementasi tiap fitur aplikasi beserta kode sumber terkait.

1.7.7. Bab VII Kesimpulan dan Saran

Bab ini berisi kesimpulan dan saran yang didapat dari proses pelaksanaan kerja praktik.

[Halaman ini sengaja dikosongkan]

BAB II
PROFIL PERUSAHAAN

2.1. Profil Laboratorium Manajemen Cerdas Informasi

Laboratorium Manajemen Cerdas Informasi (MCI) adalah salah satu laboratorium yang bernaung di bawah Departemen Teknik Informatika, Institut Teknologi Sepuluh Nopember (ITS), Surabaya. Laboratorium ini menawarkan bidang keahlian yang menekankan kemampuan menganalisis, mensintesis, dan mengevaluasi proses bisnis dan sistem informasi pada sistem enterprise, mengimplementasikan rekayasa pengetahuan ke dalam suatu aplikasi, melakukan investigasi, pengujian, dan evaluasi kematangan tata kelola teknologi informasi, serta merancang dan mengimplementasikan solusi basis data terdistribusi dan teknologi big data.

Mata kuliah pada bidang keahlian Manajemen Cerdas Informasi antara lain Sistem Enterprise, Rekayasa Pengetahuan, Sistem Informasi Geografis, Audit Sistem, Tata Kelola Teknologi Informasi, Basis Data Terdistribusi, Big Data, serta Topik Khusus Manajemen Informasi. Di samping aktivitas penelitian, laboratorium ini juga berperan sebagai tempat bagi mahasiswa untuk mengerjakan proyek, tugas akhir, maupun kerja praktik yang berhubungan dengan pengembangan sistem informasi dan aplikasi berbasis data, termasuk kerja praktik ini.

2.2. Lokasi

Jl. Teknik Kimia, Keputih, Kec. Sukolilo, Surabaya, Jawa Timur 60111, Indonesia.

[Halaman ini sengaja dikosongkan]

BAB III
TINJAUAN PUSTAKA

3.1. Flutter dan Dart

Flutter adalah framework pengembangan aplikasi lintas platform buatan Google yang memungkinkan pembuatan aplikasi Android, iOS, web, dan desktop dari satu basis kode. Flutter menggunakan bahasa pemrograman Dart serta konsep widget, yaitu komponen antarmuka yang disusun secara berlapis (composition) untuk membentuk tampilan. Berbeda dengan pendekatan native, Flutter menggambar antarmukanya sendiri melalui mesin rendering Skia sehingga tampilan konsisten di berbagai perangkat.

Dart mendukung kompilasi ahead-of-time (AOT) untuk performa tinggi saat rilis dan just-in-time (JIT) yang memungkinkan hot reload saat pengembangan. Fitur hot reload membuat perubahan kode langsung terlihat tanpa perlu membangun ulang aplikasi dari awal, sehingga proses pengembangan antarmuka menjadi lebih cepat. Karena kebutuhan aplikasi mobile yang ringan, responsif, dan mudah dikembangkan secara berkelanjutan, aplikasi ZooSite dibangun sepenuhnya menggunakan Flutter dan Dart [6].

3.2. Clean Architecture

Clean Architecture adalah pola perancangan yang memisahkan kode ke dalam beberapa lapisan yang saling independen, umumnya data, domain, dan presentation. Lapisan domain memuat entitas dan aturan bisnis inti yang tidak bergantung pada kerangka kerja, lapisan data menangani sumber data, model, dan repositori, sedangkan lapisan presentation mengurus antarmuka serta pengelolaan state. Pemisahan tanggung

jawab ini membuat setiap lapisan dapat dikembangkan dan diuji secara terpisah.

Prinsip utamanya adalah aturan ketergantungan (dependency rule), yaitu ketergantungan hanya boleh mengarah ke dalam menuju domain. Dengan demikian, logika inti tidak terpengaruh perubahan teknologi di lapisan luar seperti basis data maupun antarmuka. Pola ini membuat kode lebih mudah dipelihara dan dikembangkan; pada kerja praktik ini clean architecture menjadi dasar penyatuan modul-modul warisan karena setiap fitur dapat ditempatkan pada direktori tersendiri tanpa mengganggu fitur lain [10].

3.3. Manajemen State: BLoC dan Provider

Manajemen state adalah cara aplikasi mengelola dan menyebarkan perubahan data ke antarmuka. BLoC (Business Logic Component) memisahkan logika bisnis dari tampilan melalui aliran event yang masuk dan state yang keluar. Pola ini membuat alur data menjadi terprediksi dan mudah diuji karena setiap perubahan state berasal dari event yang jelas.

Provider merupakan mekanisme yang lebih ringan untuk menyediakan (dependency injection) dan berbagi objek atau state ke seluruh pohon widget. Pemisahan ini membuat antarmuka hanya bertugas menampilkan state, sedangkan proses seperti klasifikasi atau pemanggilan data ditangani pada lapisan terpisah. Pada ZooSite, BLoC digunakan untuk chatbot dan klasifikasi gambar, sedangkan Provider digunakan untuk klasifikasi suara serta penyimpanan pilihan kebun binatang [6].

3.4. Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) adalah arsitektur jaringan saraf tiruan yang dirancang untuk mengolah data berbentuk grid seperti citra. CNN menerapkan operasi konvolusi

menggunakan sejumlah filter untuk mengekstraksi fitur, mulai dari tepi dan tekstur hingga pola yang lebih kompleks. Informasi tersebut kemudian diringkas melalui lapisan pooling dan diklasifikasikan pada lapisan fully connected [12].

Keunggulan utama CNN adalah kemampuannya mempelajari fitur secara otomatis tanpa perlu ekstraksi fitur manual. Hal ini membuat CNN sangat sesuai untuk tugas pengenalan objek pada gambar maupun suara. Pada aplikasi ZooSite, CNN menjadi dasar model klasifikasi gambar hewan [2] dan juga digunakan pada pengenalan suara hewan setelah sinyal audio diubah menjadi representasi spektral [5].

3.5. Transfer Learning

Transfer learning adalah teknik memanfaatkan pengetahuan dari model yang telah dilatih pada dataset besar untuk diterapkan pada tugas baru yang datanya lebih terbatas. Alih-alih melatih model dari nol, bobot model pra-latih digunakan kembali dan hanya sebagian lapisan yang disesuaikan (fine-tuning) sesuai tugas baru. Cara ini memungkinkan model memanfaatkan fitur umum yang sudah dipelajari sebelumnya.

Pendekatan ini menghemat waktu pelatihan dan umumnya menghasilkan akurasi yang lebih tinggi ketika data yang tersedia terbatas. Teknik ini banyak digunakan pada tugas klasifikasi citra yang datasetnya tidak terlalu besar. Dalam pengembangan ZooSite, transfer learning digunakan untuk meningkatkan akurasi klasifikasi hewan [3] dan klasifikasi ikan [4].

3.6. TensorFlow Lite

TensorFlow Lite (TFLite) adalah pustaka ringan untuk menjalankan model machine learning secara langsung di perangkat (on-device inference). Model yang telah dilatih dikonversi ke format .tflite yang berukuran kecil dan dioptimalkan. Dengan optimasi tersebut, model dapat berjalan pada perangkat dengan sumber daya terbatas seperti ponsel.

Karena inferensi dilakukan di perangkat, aplikasi dapat bekerja tanpa koneksi internet sekaligus menjaga privasi data pengguna. Proses klasifikasi pun menjadi lebih cepat karena data tidak perlu dikirim ke server terlebih dahulu. Pada ZooSite, TFLite digunakan untuk klasifikasi gambar hewan dan ikan serta klasifikasi suara, dengan berkas model disimpan pada aset aplikasi [7].

3.7. FastAPI dan Model YAMNet

FastAPI adalah framework web berbasis Python untuk membangun REST API dengan performa tinggi dan penulisan kode yang ringkas [8]. Framework ini banyak digunakan untuk menyediakan layanan machine learning melalui antarmuka HTTP. YAMNet adalah model jaringan saraf pra-latih yang mampu mengenali berbagai peristiwa suara berdasarkan basis data AudioSet [9].

Pada fitur klasifikasi suara, backend FastAPI menerima berkas audio, melakukan praproses menggunakan pustaka librosa, lalu menjalankan model turunan YAMNet untuk mengenali jenis suara hewan. Hasil klasifikasi kemudian dikembalikan ke aplikasi dalam format JSON. Pendekatan berbasis server ini menjadi alternatif ketika inferensi ingin dipindahkan dari perangkat ke sisi server.

3.8. Universal Sentence Encoder, Word Embedding, dan TF-IDF

Universal Sentence Encoder (USE) adalah model yang memetakan kalimat menjadi vektor berdimensi tetap sehingga kemiripan makna antar-kalimat dapat dihitung secara numerik. Dengan representasi ini, dua kalimat yang berbeda susunan kata namun bermakna serupa tetap dapat dikenali kemiripannya. Model ini menjadi dasar chatbot ZooSite versi awal [1] [13].

Word embedding merepresentasikan kata sebagai vektor sehingga kata bermakna serupa memiliki posisi berdekatan, sedangkan TF-IDF (Term Frequency-Inverse Document Frequency) menilai bobot kata berdasarkan frekuensi kemunculannya pada dokumen dan korpus. Kedua pendekatan tersebut memiliki keunggulan berbeda, yaitu embedding kuat menangkap makna sedangkan TF-IDF andal pada kata kunci spesifik. Pada versi sekarang, chatbot menggabungkan keduanya dalam sebuah pencocok hibrida untuk meningkatkan ketepatan jawaban terhadap pertanyaan pengguna.

3.9. Google Maps Flutter

Google Maps Flutter adalah plugin resmi untuk menampilkan peta Google Maps di dalam aplikasi Flutter. Plugin ini memungkinkan penambahan penanda (marker), pengaturan titik pusat, serta tingkat zoom secara langsung dari kode Dart. Interaksi peta lain seperti menggeser dan memperbesar tampilan juga didukung secara bawaan.

Peta ditampilkan menggunakan Maps SDK sehingga memerlukan kunci API dan izin yang sesuai pada konfigurasi platform. Tanpa

kunci API yang valid, peta tidak akan tampil dengan benar. Pada ZooSite, plugin ini digunakan untuk menampilkan lokasi kebun binatang beserta penandanya sesuai kota yang dipilih pengguna [6].

BAB IV
ANALISIS DAN PERANCANGAN INFRASTRUKTUR SISTEM

4.1. Analisis Sistem

Pada bab ini dijelaskan tahapan analisis dan perancangan sistem yang dikerjakan selama kerja praktik. Analisis diawali dengan membandingkan basis kode awal (ZooSiteApp-original) dengan target akhir (ZooSiteApp-main) untuk menentukan fitur yang perlu ditambahkan maupun diperbaiki.

4.1.1. Definisi Umum Aplikasi

ZooSite adalah aplikasi mobile untuk pengunjung kebun binatang yang menyediakan empat fungsi utama: klasifikasi gambar hewan dan ikan, klasifikasi/pengenalan suara hewan, chatbot informasi (ZooBot Assistant), serta peta lokasi kebun binatang. Pengguna dapat memilih kebun binatang (misalnya KBS Surabaya atau Bandung) melalui pemilih yang berada di bagian atas halaman, dan navigasi utama terbagi menjadi tab Peta, Fitur, dan Gallery.

Setiap fitur pada aplikasi berasal dari modul Tugas Akhir yang berbeda, sehingga tugas utama kerja praktik ini adalah menyatukan modul-modul tersebut (Cikanindi, 2021; Kusumo, 2023; Auliya, 2024; Fuadah, 2025; Saputra, 2025) ke dalam satu basis kode.

4.1.2. Analisis Proyek Sebelumnya

Aplikasi ZooSite bukan dibangun dari nol, melainkan merupakan hasil pengembangan berkelanjutan dari beberapa Tugas Akhir mahasiswa terdahulu. Setiap Tugas Akhir

menghasilkan satu modul atau fitur yang berdiri sendiri, sehingga sebelum kerja praktik ini dilakukan, fitur-fitur tersebut belum tergabung dalam satu aplikasi yang utuh. Analisis terhadap proyek-proyek sebelumnya dilakukan untuk memahami teknologi, kondisi, dan keterkaitan setiap modul, sebagaimana diuraikan berikut.

Fitur chatbot berasal dari Tugas Akhir Cikanindi (2021), yang menerapkan Universal Sentence Encoder (USE) untuk location-aware chatbot pada aplikasi layanan wisata kebun binatang. Modul ini diimplementasikan sebagai aplikasi Android terpisah (BandungZooChatbot) berbasis Java yang menjalankan kode Python melalui Chaquopy, dengan basis pengetahuan disimpan pada Firebase Realtime Database secara daring.

Fitur klasifikasi hewan dikembangkan pertama kali oleh Kusumo (2023) menggunakan algoritma Convolutional Neural Network (CNN), lalu ditingkatkan oleh Auliya (2024) menggunakan transfer learning untuk memperbaiki akurasi klasifikasi. Kedua Tugas Akhir tersebut menghasilkan model klasifikasi hewan (dikenal sebagai model kusumo dan auliya) yang berjalan di perangkat menggunakan TensorFlow Lite.

Fitur klasifikasi ikan dikembangkan oleh Fuadah (2025) menggunakan transfer learning, menghasilkan model klasifikasi ikan (model fuadah) beserta data 26 jenis ikan. Sementara itu, fitur pengenalan suara hewan dikembangkan oleh Saputra (2025) menggunakan Convolutional Neural Network (CNN) untuk mengenali suara hewan pada aplikasi ZooSite.

Dari hasil analisis tersebut, teridentifikasi bahwa masing-masing modul memiliki teknologi, format basis

pengetahuan, dan tingkat kematangan yang berbeda-beda, serta sebagian belum terintegrasi maupun belum berfungsi dengan baik. Ringkasan modul warisan tiap proyek sebelumnya ditunjukkan pada Tabel

Tabel 4.1 Analisis Proyek Sebelumnya

Sumber Tugas Akhir	Tahun	Fitur	Teknologi
Cikanindi	2021	Chatbot (location-aware)	Universal Sentence Encoder, Firebase, Chaquopy (Android/Java)
Kusumo	2023	Klasifikasi hewan	CNN, TensorFlow Lite
Auliya	2024	Perbaikan klasifikasi hewan	Transfer Learning, TensorFlow Lite
Fuadah	2025	Klasifikasi ikan	Transfer Learning, TensorFlow Lite
Saputra	2025	Pengenalan suara hewan	CNN

4.2. Perancangan Infrastruktur Sistem

4.2.1. Desain Arsitektur Sistem

Aplikasi dirancang mengikuti clean architecture berbasis fitur. Setiap fitur baru ditempatkan pada direktori lib/src/features/ dan dibagi menjadi lapisan data (data source, model, repository), domain (entity, repository abstrak, use case), dan presentation (bloc/provider, screen, view, widget). Integrasi antarfitur dilakukan melalui berkas dependency injection (injection_container) dan routing (router) yang terpusat, serta service bersama seperti ZooSelectionService untuk menyimpan pilihan kebun binatang pengguna.

Tabel 4.2 Fitur Aplikasi Tahap Akhir

No	Komponen	Teknologi
1	Klasifikasi Gambar (Hewan & Ikan)	TensorFlow Lite; model auliya, fuadah (ikan), kusumo (.tflite)
2	Klasifikasi Suara	REST API FastAPI + model YAMNet; plugin record, audioplayers, http
3	Chatbot ZooBot	Hybrid matcher (word embedding + TF-IDF); brainfile per kota
4	Pemilihan Kebun Binatang & Peta	google_maps_flutter; shared_preferences; ZooSelectionService
5	Antarmuka Halaman Utama	Flutter widget; layout carousel + bottom navigation

Tabel 4.3 Pembagian Tugas

No	Nama	Tanggung Jawab
1	Giandra	Fitur chatbot (ZooBot Assistant); fitur pemilihan kebun binatang dan peta (Google Maps); penambahan routing untuk layar chatbot dan pemilihan kebun binatang.
2	Bosco	Pembaruan antarmuka dan tata letak halaman utama; fitur klasifikasi suara beserta API suara; perbaikan integrasi fitur klasifikasi ikan yang sebelumnya tidak berfungsi.

4.2.2. Proses Penggabungan Modul

Tugas utama kerja praktik ini adalah menggabungkan dan mengintegrasikan seluruh modul warisan menjadi satu aplikasi ZooSite yang utuh (ZooSiteApp-main). Karena setiap modul semula berdiri sendiri dengan teknologi yang berbeda, proses penggabungan dilakukan secara bertahap dengan tahapan sebagai berikut.

Pertama, seluruh modul disatukan ke dalam satu basis kode Flutter dengan struktur clean architecture berbasis fitur. Setiap fitur ditempatkan pada direktori *lib/src/features/* (chatbot, image_classification, sound_classification, dan home) dan dibagi menjadi lapisan data, domain, serta presentation, sehingga setiap modul dapat berdampingan tanpa saling mengganggu.

Kedua, dilakukan penyesuaian dependency injection pada berkas *injection_container.dart* dan *injection_container_main.dart.* Setiap fitur baru didaftarkan layanannya (service, repository, data source, dan provider/bloc),

termasuk penambahan *ZooSelectionService* sebagai layanan bersama untuk menyimpan pilihan kebun binatang pengguna.

Ketiga, dilakukan penyesuaian routing pada berkas *router.dart* dan *router_main.dart* untuk mendaftarkan layar-layar baru, yaitu *ChatBotScreen* dan *ZooSelectionScreen*, agar dapat diakses dari seluruh bagian aplikasi.

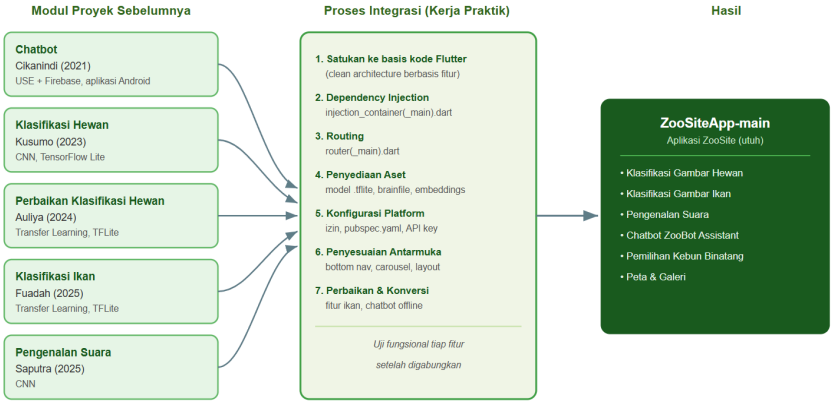
Keempat, dilakukan penyediaan aset yang dibutuhkan tiap modul, meliputi berkas model klasifikasi (*.tflite*) untuk hewan, ikan, dan suara, serta basis pengetahuan chatbot berupa *brainfile.json*, *embeddings.bin*, dan *vocab.bin* untuk setiap kota (Surabaya dan Bandung), yang didaftarkan pada berkas *pubspec.yaml*.

Kelima, dilakukan penyesuaian konfigurasi platform. Pada AndroidManifest.xml ditambahkan izin RECORD_AUDIO, atribut *usesCleartextTraffic*, serta meta-data kunci API Google Maps. Pada *pubspec.yaml* ditambahkan sejumlah dependensi baru seperti *record, audioplayers, http, google_maps_flutter, shared_preferences,* dan lainnya sesuai kebutuhan tiap modul.

Keenam, dilakukan penyesuaian antarmuka halaman utama agar seluruh fitur dapat diakses secara rapi, meliputi perubahan tata letak menjadi dapat digulir, penambahan bottom navigation (Peta, Fitur, Gallery), serta perapian carousel fitur pada halaman utama sehingga tiap indeks carousel terhubung ke fitur yang tepat.

Ketujuh, dilakukan penyelesaian konflik dan perbaikan pada modul yang belum berfungsi setelah digabungkan. Sebagai contoh, fitur klasifikasi ikan tidak dapat berjalan karena model ikan tidak pernah diaktifkan pada alur pemilihan model; hal ini diperbaiki dengan menambahkan pemanggilan pemilihan model yang sesuai (lihat Bab V).

Selain itu, beberapa modul juga dikonversi agar sesuai dengan arsitektur aplikasi. Modul chatbot yang semula berupa aplikasi Android terpisah berbasis USE dan Firebase (Cikanindi, 2021) diimplementasikan ulang menjadi modul Flutter yang berjalan sepenuhnya di perangkat menggunakan pencocokan hibrida (word embedding dan TF-IDF) dengan basis pengetahuan lokal, sehingga tidak lagi bergantung pada koneksi internet.



Gambar 4.1 Illustrasi Penggabungan

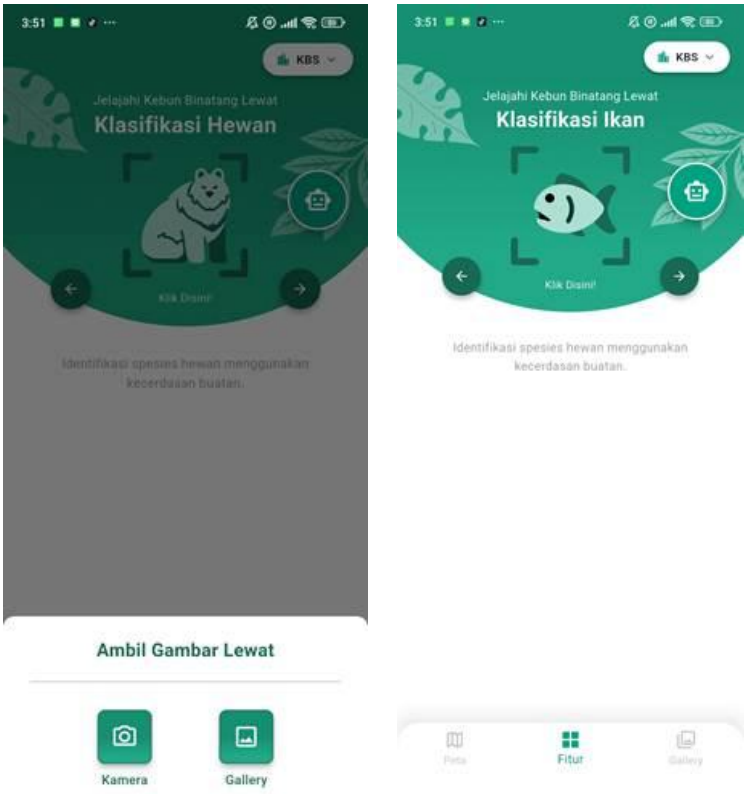
[Halaman ini sengaja dikosongkan]

BAB V
IMPLEMENTASI SISTEM

Bab ini membahas implementasi dari setiap fitur yang dikerjakan selama kerja praktik, disertai kode sumber dan tangkapan layar yang relevan.

5.1. Implementasi Infrastruktur Plugin

Fitur klasifikasi gambar memungkinkan pengguna mengidentifikasi spesies hewan maupun ikan dengan mengambil gambar melalui kamera atau memilih dari galeri. Inferensi dilakukan di perangkat menggunakan TensorFlow Lite. Ketika pengguna menekan area pemindaian, aplikasi menampilkan pilihan sumber gambar (Kamera atau Gallery) seperti terlihat pada Gambar 5.1, kemudian gambar diproses oleh model .tflite yang sesuai.



Gambar 5.1 Halaman Klasifikasi Hewan (kiri) dan Klasifikasi Ikan (kanan)

5.1.1 Perbaikan Fitur Klasifikasi Ikan

Pada versi awal aplikasi, fitur klasifikasi ikan tidak dapat berfungsi meskipun berkas model (fuadah_model.tflite) beserta data ikan (nama, nama latin, deskripsi, dan fakta unik) telah tersedia. Setelah ditelusuri, penyebabnya berada pada pemilih fitur (carousel) di halaman utama, yaitu berkas home_header_view.dart. Ketika pengguna menekan header ikan, aplikasi langsung membuka pemilih sumber gambar tanpa terlebih dahulu mengaktifkan model ikan. Akibatnya, proses klasifikasi berjalan menggunakan model yang tidak sesuai, sehingga hasilnya tidak valid. Pada seluruh basis kode versi awal, model ikan (ClassificationModelType.fuadah) memang tidak pernah dipanggil melalui updateModel(), sehingga model tersebut tidak pernah benar-benar dimuat saat fitur digunakan.

Perbaikan dilakukan dengan menambahkan pemanggilan pemilihan model ikan sebelum pemilih gambar dibuka, seperti pada Kode Sumber 5.1. Selain itu, urutan carousel pada berkas home_provider.dart dirapikan menjadi indeks 0 (Klasifikasi Hewan), 1 (Klasifikasi Ikan), dan 2 (Pengenalan Suara). Setelah perbaikan, model ikan berhasil dimuat dengan benar dan fitur dapat mengenali 26 jenis ikan yang terdaftar (mulai dari Ikan Arwana, Ikan Koi, Ikan Nemo, hingga Ikan Koki).

```
} else if (provider.currentHeader == 1) {
  // Klasifikasi Ikan
  context.classificationProvider.updateModel(
    ClassificationModelType.fuadah,   // <-- baris perbaikan
  );
  context.homeBloc.add(const
```

```
OpenImageSourceBottomSheetEvent());
}
```

Kode Sumber 5.1 Pengaktifan model ikan pada home_header_view.dart (perbaikan fitur ikan)

5.2. Klasifikasi Suara Hewan

Fitur klasifikasi suara memungkinkan pengguna merekam suara hewan (atau memilih berkas audio dari perangkat), memutar ulang rekaman, lalu mengklasifikasikannya. Pengelolaan perekaman dilakukan oleh SoundClassificationProvider menggunakan plugin record (format WAV, 16 kHz, mono, sesuai kebutuhan model), audioplayers untuk pemutaran, serta file_picker dan path_provider untuk pemilihan dan penyimpanan berkas. Halaman fitur pengenalan suara ditampilkan pada Gambar 5.2.



Gambar 5.2 Halaman Fitur Pengenalan Suara

Pada versi final aplikasi, inferensi dijalankan langsung di perangkat (on-device) menggunakan TensorFlow Lite melalui berkas model animal_sound_model.tflite, sehingga fitur tidak bergantung pada koneksi ke server. Berkas WAV diuraikan menjadi sampel audio ternormalisasi, kemudian dijalankan

melalui interpreter TFLite dan hasilnya ditentukan melalui operasi argmax untuk memperoleh indeks kelas hewan, sebagaimana pada Kode Sumber 5.2.

```
class SoundClassificationLocalDataSourceImpl {
  static const _modelAsset =
      'assets/models/animal_sound_model.tflite';

  Future<SoundAnimalModel> classifySound(File audioFile)
async {
    final bytes = await compute(_readFile, audioFile.path);
    final samples = await compute(_parseWav, bytes); // WAV ->
float32
    final interpreter = await _getInterpreter();
    interpreter.resizeInputTensor(0, [samples.length]);
    interpreter.allocateTensors();
    final input = Float32List.fromList(samples);
    final outShape = interpreter.getOutputTensor(0).shape;
    final output = Float32List(outShape[1]);
    interpreter.run(input, output.reshape([1, outShape[1]]));
    final idx = _argmax(output);          // indeks kelas
    return SoundAnimalData.getByIndex(idx); // -> data hewan
  }
}
```

Kode Sumber 5.2 Inferensi suara on-device dengan TFLite

Selain jalur on-device, dikembangkan pula alternatif berbasis REST API yang memindahkan inferensi ke sisi server menggunakan FastAPI dan model YAMNet. Untuk mode ini, alamat backend dikonfigurasi secara terpusat pada berkas

constants.dart agar mudah disesuaikan dengan lingkungan pengembangan (emulator, perangkat fisik, atau simulator), seperti pada Kode Sumber 5.3.

```
class Constants {
  // Sound Classification API (mode REST/alternatif)
  // Android Emulator : '10.0.2.2'
  // Physical device  : IP mesin, mis. '192.168.1.10'
  static const String soundApiHost = '10.0.2.2';
  static const int soundApiPort = 8000;
  static const String soundApiBaseUrl =
      'http://$soundApiHost:$soundApiPort';
}
```

Kode Sumber 5.3 Konfigurasi endpoint API suara pada constants.dart

Pada mode REST, berkas audio hasil rekaman dikirim ke endpoint /classify menggunakan permintaan HTTP multipart, kemudian respons JSON diuraikan menjadi objek SoundAnimalModel, sebagaimana ditunjukkan pada Kode Sumber 5.4.

```
class Constants {
  // Sound Classification API (mode REST/alternatif)
  // Android Emulator : '10.0.2.2'
  // Physical device  : IP mesin, mis. '192.168.1.10'
  static const String soundApiHost = '10.0.2.2';
  static const int soundApiPort = 8000;
  static const String soundApiBaseUrl =
```

```
'http://$soundApiHost:$soundApiPort';
}
```

Kode Sumber 5.4 Pengiriman audio ke REST API

Di sisi backend, endpoint /classify pada FastAPI menerima berkas audio, menyimpannya sementara, melakukan praproses, menjalankan prediksi, lalu mengembalikan data hewan yang cocok. Potongan intinya ditunjukkan pada Kode Sumber 5.5.

```
@app.post("/classify")
async def classify_animal_sound(file: UploadFile = File(...)):
    with tempfile.NamedTemporaryFile(
            delete=False, suffix=".wav") as tmp:
        tmp.write(await file.read())
        temp_path = tmp.name
    audio_array = preprocess_audio(temp_path)
    prediction = predict_animal_sound(audio_array)
    animal_key = prediction["predicted_class"]
    animal_data = ANIMAL_DATA.get(
        animal_key, ANIMAL_DATA["unknown"])
    return {"prediction": prediction,
            "animal_data": animal_data}
```

Kode Sumber 5.5 Endpoint /classify pada backend FastAPI (main.py)

5.3. Chatbot ZooBot Assistant

Chatbot pada aplikasi ini sebelumnya telah memiliki versi awal, yaitu aplikasi Android terpisah bernama

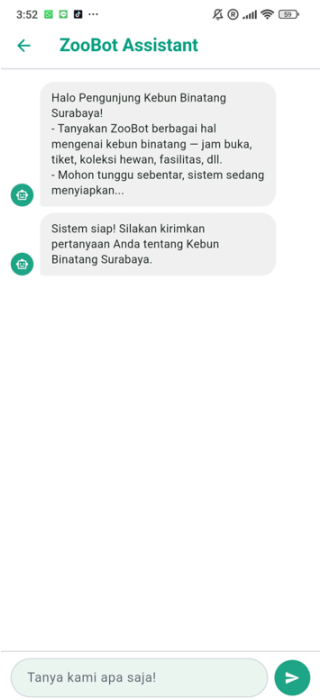
BandungZooChatbot yang ditulis dengan bahasa Java dan menjalankan kode Python melalui pustaka Chaquopy. Pada versi awal tersebut, basis pengetahuan (brainfile berupa pasangan pertanyaan-jawaban) disimpan pada Firebase Realtime Database secara daring, dan proses pencocokan pertanyaan menggunakan model Universal Sentence Encoder dari TensorFlow Hub yang dibandingkan melalui cosine similarity dengan ambang 0,5; bila kemiripan tertinggi berada di bawah ambang tersebut, chatbot menjawab bahwa pertanyaan tidak dapat dipahami. Versi awal ini juga bergantung pada layanan penerjemahan daring (googletrans) serta koneksi internet untuk mengakses Firebase dan mengunduh model. Tetapi saat dicoba untuk di run, aplikasi tidak bisa mendapatkan model karena domainnya sudah expired. Aplikasi ini juga hanya bisa dijalankan secara online.

Pada versi sekarang, chatbot diintegrasikan langsung ke dalam aplikasi ZooSite berbasis Flutter sebagai modul fitur tersendiri (ZooBot Assistant) yang berjalan sepenuhnya di perangkat. Basis pengetahuan dipindahkan menjadi berkas brainfile.json yang dibundel pada aset aplikasi untuk setiap kota, disertai berkas embeddings dan vocab lokal, sehingga tidak lagi bergantung pada Firebase maupun koneksi internet. Metode pencocokan pun ditingkatkan menjadi pendekatan hibrida. Perbandingan versi awal dan versi sekarang dirangkum pada Tabel 5.1.

Tabel 5.1 Perbandingan Fitur Chatbot

Aspek	Versi Awal (BandungZooChatbot)	Versi Sekarang (ZooSite)
Platform	Aplikasi Android native (Java) terpisah, menjalankan Python melalui Chaquopy.	Modul fitur pada aplikasi Flutter ZooSite (Dart), terintegrasi penuh.
Basis pengetahuan	Brainfile pada Firebase Realtime Database (daring).	Berkas brainfile.json pada aset aplikasi, per kota (Surabaya dan Bandung).
Metode pencocokan	Universal Sentence Encoder (TF Hub) + cosine similarity, ambang 0,5.	Pencocokan hibrida: word embedding + TF-IDF dengan domain guard dan konsensus.
Ketergantungan internet	Membutuhkan internet (Firebase, unduh model, googletrans).	Berjalan sepenuhnya offline di perangkat.
Konteks kota	Satu basis pengetahuan (Bandung).	Menyesuaikan kota terpilih melalui ZooSelectionService.

Chatbot ZooBot Assistant menjawab pertanyaan pengunjung seputar kebun binatang, seperti jam buka, tiket, koleksi hewan, dan fasilitas. Basis pengetahuan disimpan dalam bentuk brainfile per kota (Surabaya dan Bandung) beserta berkas embeddings dan vocab. Tampilan percakapan chatbot ditunjukkan pada Gambar 5.3.



Gambar 5.3 Tampilan Chatbot ZooBot Assistant

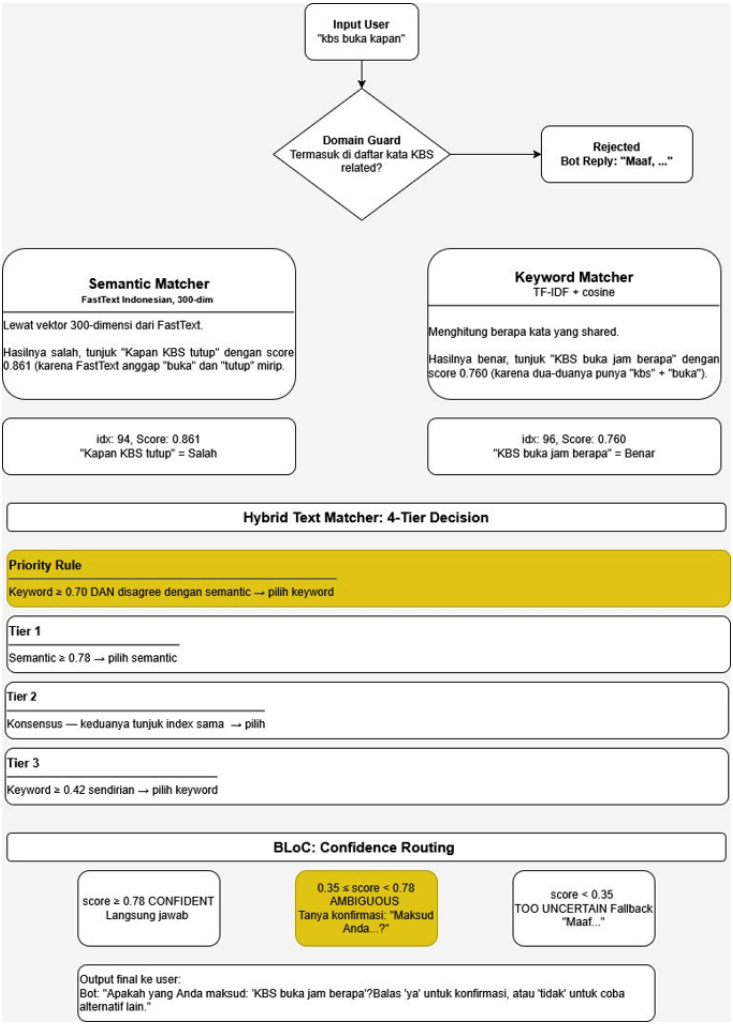
Untuk meningkatkan akurasi jawaban, chatbot menggunakan pencocok hibrida yang menggabungkan pencocokan semantik (word embedding) dan pencocokan kata kunci (TF-IDF) dengan mekanisme konsensus, sebagaimana pada Kode Sumber 5.6. Pendekatan ini mengutamakan presisi sehingga sistem lebih memilih menyatakan "tidak dipahami" daripada memberikan jawaban yang salah.

```
class HybridTextMatcher implements TextMatcher {
  HybridTextMatcher({
```

```
    required this.primary,   // WordEmbeddingMatcher
    required this.fallback,  // TfIdfTextMatcher
    this.semanticVeryHigh = 0.78,
    this.consensusSemanticMin = 0.50,
    this.consensusKeywordMin = 0.25,
    this.keywordStrong = 0.42,
  });
  // 1. semantik sangat tinggi  -> terima semantik
  // 2. konsensus dua matcher   -> terima
  // 3. kata kunci kuat         -> terima keyword
  // 4. selain itu              -> tidak cocok
}
```

Kode Sumber 5.6 Pencocok hibrida chatbot (hybrid_text_matcher.dart)

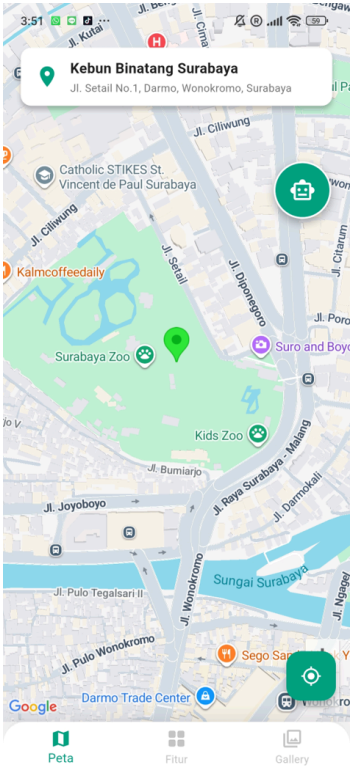
Secara keseluruhan, alur pemrosesan pertanyaan pengguna pada chatbot digambarkan pada Gambar 5.4. Pertanyaan mula-mula disaring oleh Domain Guard untuk memastikan pertanyaan masih relevan dengan kebun binatang. Selanjutnya pertanyaan diproses secara paralel oleh Semantic Matcher (FastText 300 dimensi) dan Keyword Matcher (TF-IDF + cosine). Kedua skor tersebut dipadukan pada Hybrid Text Matcher melalui keputusan empat tingkat (priority rule, tier 1-3), lalu lapisan Confidence Routing pada BLoC menentukan apakah jawaban langsung diberikan, meminta konfirmasi ketika ambigu, atau memberikan pesan fallback ketika terlalu tidak yakin.



Gambar 5.4 Alur keputusan pencocok hibrida (Hybrid Text Matcher) pada chatbot

5.4. Pemilihan Kebun Binatang dan Peta

Fitur ini memungkinkan pengguna memilih kebun binatang yang ingin dijelajahi. Pilihan tersebut disimpan menggunakan shared_preferences melalui ZooSelectionService sehingga tetap tersimpan saat aplikasi dibuka kembali, dan memengaruhi konten fitur lain (misalnya brainfile chatbot dan penanda peta). Tab Peta menampilkan lokasi kebun binatang yang dipilih pada Google Maps, seperti pada Gambar 5.5.



Gambar 5.5 Peta Lokasi Kebun Binatang Surabaya (KBS)

Layar chatbot dan pemilihan kebun binatang didaftarkan pada routing terpusat agar dapat diakses dari seluruh bagian aplikasi, seperti pada Kode Sumber 5.7.

```
case ChatBotScreen.routeName:
  return _pageBuilder(
      (_) => const ChatBotScreen(),
      settings: settings);
case ZooSelectionScreen.routeName:
  final isFirstLaunch =
      (settings.arguments as bool?) ?? false;
  return _pageBuilder(
      (_) => ZooSelectionScreen(isFirstLaunch: isFirstLaunch),
      settings: settings);
```

Kode Sumber 5.7 Penambahan rute chatbot dan pemilihan kebun binatang (router_main.dart)

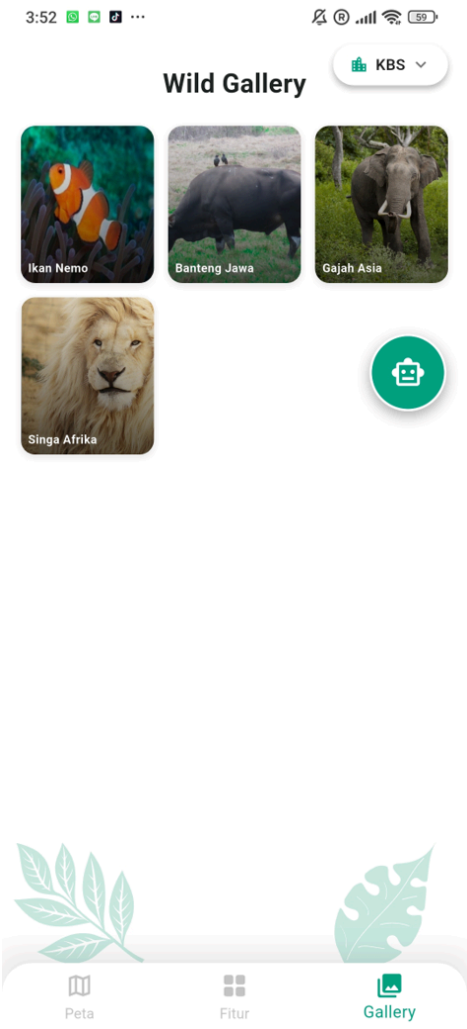
5.5 Pembaruan Antarmuka Halaman Utama

Agar halaman utama dapat menampung fitur-fitur baru secara rapi dan tetap nyaman pada berbagai ukuran layar, tata letak diperbarui dari Column dengan tinggi tetap menjadi SingleChildScrollView yang dapat digulir. Selain itu, judul dan ikon header (misalnya "Klasifikasi Hewan") dikembalikan pada carousel fitur. Perubahan tata letak ditunjukkan pada Kode Sumber 5.8, sedangkan galeri satwa (Wild Gallery) ditampilkan pada Gambar 5.6.

```
return SingleChildScrollView(
  child: Column(
    mainAxisSize: MainAxisSize.min,
    children: [
```

```
      const HomeHeaderView(),
      SizedBox(height: context.heightScale * 32),
      Padding(
        padding: const EdgeInsets.symmetric(horizontal: 20),
        child: HomeChatBotView(chatBotController:
chatBotController),
      ),
      HomeHistoryView( ... ),
      const SizedBox(height: 50),
    ],
  ),
);
```

Kode Sumber 5.8 Perubahan tata letak halaman utama (home_view.dart)



Gambar 5.6 Tampilan Galeri Satwa (Wild Gallery)

Selain perubahan pada kode Dart, integrasi fitur baru juga memerlukan penyesuaian konfigurasi platform. Pada AndroidManifest.xml ditambahkan izin RECORD_AUDIO untuk perekaman suara, atribut usesCleartextTraffic agar aplikasi dapat mengakses API suara melalui HTTP saat pengembangan, serta meta-data kunci API Google Maps. Berkas pubspec.yaml juga diperbarui dengan dependensi baru: record, audioplayers, http, http_parser, google_maps_flutter, shared_preferences, file_picker, dan path_provider.

BAB VI
PENGUJIAN DAN EVALUASI

Bab ini menjelaskan tahap uji coba terhadap aplikasi ZooSite. Pengujian dilakukan untuk memastikan fungsionalitas dan kesesuaian hasil implementasi dengan analisis dan perancangan sistem yang telah dibuat.

6.1. Tujuan Pengujian

Pengujian dilakukan terhadap aplikasi ZooSite yang dikembangkan selama kerja praktik ini. Tujuan utama pengujian adalah memastikan bahwa aplikasi berjalan sesuai dengan rancangan, baik dari sisi fungsi maupun kesesuaian keluaran. Secara khusus, tujuan pengujian ini meliputi:

- Memastikan setiap fitur utama, yaitu klasifikasi gambar hewan dan ikan, pengenalan suara, chatbot, serta pemilihan kebun binatang dan peta, berfungsi sebagaimana mestinya.
- Memverifikasi bahwa hasil klasifikasi gambar dan suara sesuai dengan masukan yang diberikan pengguna.
- Memastikan konten aplikasi menyesuaikan kebun binatang yang dipilih pengguna.
- Menilai kestabilan aplikasi agar dapat digunakan tanpa terjadi galat (crash) pada perangkat Android.

6.2. Kriteria Pengujian

Kriteria pengujian disusun berdasarkan aspek fungsional aplikasi, yaitu kemampuan setiap fitur untuk memberikan keluaran yang benar sesuai masukan pengguna. Penyusunan kriteria ini mengacu pada tujuan pengujian pada subbab sebelumnya serta rumusan masalah kerja praktik, sehingga setiap kriteria dapat ditelusuri kembali ke fitur yang dikembangkan maupun diperbaiki selama integrasi.

Setiap kriteria diberi kode (K1, K2, dan seterusnya) agar mudah dirujuk pada tahap evaluasi. Kriteria mencakup kestabilan aplikasi secara umum (dapat berjalan tanpa galat), ketepatan hasil klasifikasi gambar hewan dan ikan, keberhasilan proses perekaman dan klasifikasi suara, ketepatan jawaban chatbot, serta kesesuaian konten peta terhadap kebun binatang yang dipilih. Dengan demikian, kriteria tidak hanya menilai apakah fitur berjalan, tetapi juga apakah keluarannya sesuai dengan yang diharapkan pengguna. Kriteria selengkapnya dapat dilihat pada Tabel 6.1.

Tabel 6.1 Kriteria Pengujian

Kriteria Pengujian	**Deskripsi**
K1	Aplikasi dapat menampilkan seluruh halaman tanpa terjadi galat (crash).
K2	Fitur klasifikasi gambar hewan menampilkan nama, nama latin, deskripsi, dan fakta unik yang sesuai.
K3	Fitur klasifikasi gambar ikan berhasil memuat model ikan dan menampilkan hasil yang sesuai.
K4	Fitur pengenalan suara dapat merekam, memproses, dan mengembalikan hasil klasifikasi.
K5	Chatbot ZooBot dapat menjawab pertanyaan pengguna

	seputar kebun binatang.
K6	Fitur pemilihan kebun binatang dan peta menampilkan lokasi sesuai kota yang dipilih.

6.3. Skenario Pengujian

A. Pengujian Fungsionalitas Aplikasi

Skenario ini dilakukan untuk memastikan setiap fitur bekerja sesuai dengan fungsinya.

Tabel 6.2 Skenario dan Hasil yang Diharapkan

Skenario	**Hasil yang Diharapkan**
Pengguna mengambil gambar hewan melalui kamera atau galeri.	Aplikasi menampilkan nama, nama latin, deskripsi, dan fakta unik hewan.
Pengguna mengambil gambar ikan melalui kamera atau galeri.	Aplikasi memuat model ikan dan menampilkan hasil klasifikasi ikan yang sesuai.
Pengguna merekam suara hewan lalu menekan tombol Klasifikasi.	Sistem menghasilkan tabel baru dengan struktur gabungan (*union*) dan menampilkannya secara berurutan berdasar waktu.
Pengguna menjalankan fitur *Entity Recognition* pada merged timeline.	Chatbot memberikan jawaban yang relevan dengan pertanyaan.

Pengguna mengganti pilihan kebun binatang (misalnya ke Bandung).	Peta dan konten aplikasi menyesuaikan lokasi kota yang dipilih.

B. Pengujian Aspek Klasifikasi

Pengujian ini difokuskan pada aspek kecerdasan buatan yang menjadi inti aplikasi, yaitu ketepatan proses klasifikasi.

Tabel 6.3 Aspek Pengujian Klasifikasi

Aspek Pengujian	**Deskripsi**
Klasifikasi Gambar Hewan	Klasifikasi Gambar Hewan.
Klasifikasi Gambar Ikan	Menguji model ikan (fuadah) setelah perbaikan integrasi model.
Pengenalan Suara	Menguji inferensi suara on-device (TFLite) dari rekaman audio.
Chatbot ZooBot	Menguji ketepatan pencocokan pertanyaan oleh hybrid matcher.

Pengujian dilakukan pada seluruh halaman utama aplikasi, yang meliputi:

1. Halaman Utama (carousel fitur)
2. Halaman Klasifikasi Hewan
3. Halaman Klasifikasi Ikan
4. Halaman Pengenalan Suara
5. Halaman Chatbot ZooBot
6. Halaman Peta (Surabaya dan Bandung)
7. Halaman Galeri (Wild Gallery)

6.4. Evaluasi Pengujian

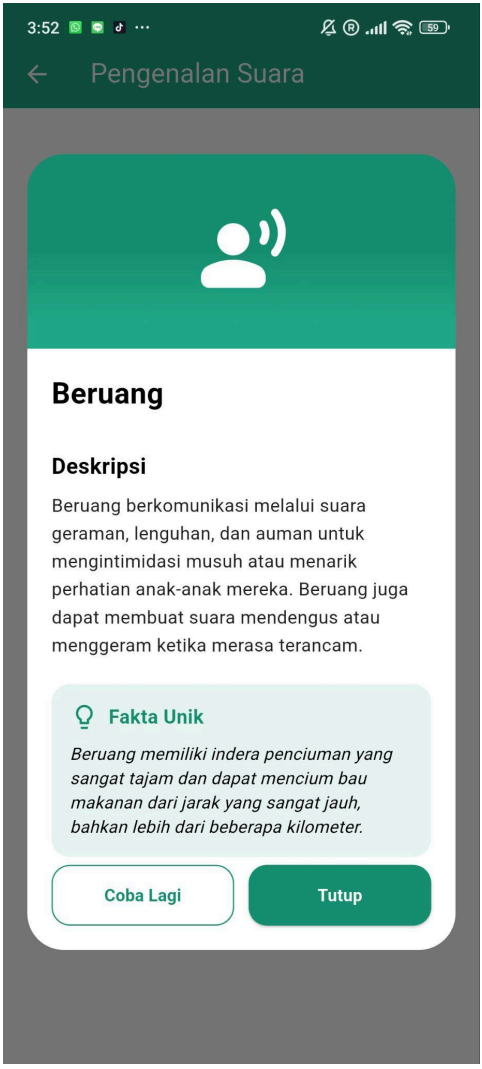
Hasil pengujian menunjukkan bahwa aplikasi ZooSite telah memenuhi seluruh aspek fungsional yang diujikan. Bukti pengujian klasifikasi hewan, ikan, dan pengenalan suara ditampilkan pada Gambar 6.1, Gambar 6.2, dan Gambar 6.3 sedangkan perbandingan peta antara Surabaya dan Bandung ditampilkan pada Gambar 6.4. Ringkasan hasil evaluasi dirangkum pada Tabel 6.4.



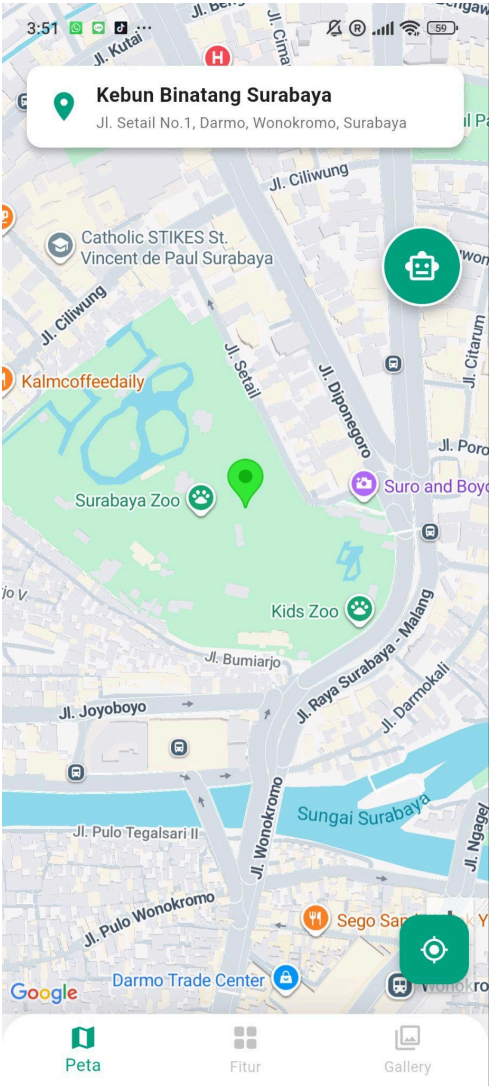
Gambar 6.1 Pengujian fitur Klasifikasi Hewan



Gambar 6.2 Pengujian fitur Klasifikasi Ikan



Gambar 6.3 Pengujian fitur Pengenalan Suara



Gambar 6.4 Peta Lokasi Kebun Binatang Bandung setelah pergantian pilihan kota

Tabel 6.4. Hasil Evaluasi Pengujian

Kriteria Pengujian	**Hasil Pengujian**
K1. Aplikasi berjalan tanpa galat (crash)	Terpenuhi
K2. Klasifikasi gambar hewan	Terpenuhi
K3. Klasifikasi gambar ikan	Terpenuhi
K4. Pengenalan suara	Terpenuhi
K5. Chatbot ZooBot	Terpenuhi
K6. Pemilihan kebun binatang dan peta	Terpenuhi

Berdasarkan hasil evaluasi, aplikasi ZooSite telah berfungsi dengan baik pada seluruh aspek fungsional, meliputi klasifikasi gambar hewan dan ikan, pengenalan suara, chatbot, serta pemilihan kebun binatang dan peta. Perbaikan pada fitur klasifikasi ikan juga telah berhasil sehingga model ikan dapat dimuat dan digunakan dengan benar.

Meskipun demikian, terdapat beberapa catatan untuk pengembangan selanjutnya. Ketepatan hasil klasifikasi gambar dan suara masih dipengaruhi oleh kualitas gambar dan rekaman yang diberikan pengguna, sehingga akurasinya masih dapat ditingkatkan melalui penambahan data latih. Selain itu, alternatif klasifikasi suara berbasis REST API sebaiknya diakses melalui HTTPS pada lingkungan produksi. Dengan demikian, aplikasi ZooSite dinilai layak digunakan, dengan rekomendasi utama berupa peningkatan akurasi model untuk pengalaman pengguna yang lebih optimal.

BAB VII
KESIMPULAN DAN SARAN

7.1. Kesimpulan

Kesimpulan yang diperoleh setelah melaksanakan kerja praktik adalah sebagai berikut:

1. Modul chatbot, klasifikasi suara, serta pemilihan kebun binatang dan peta berhasil diintegrasikan ke dalam satu basis kode ZooSite melalui penyesuaian dependency injection, routing, dan penyediaan aset, mengikuti struktur clean architecture berbasis fitur. (menjawab rumusan 1)
2. Fitur klasifikasi ikan yang sebelumnya tidak berfungsi berhasil diperbaiki dengan mengaktifkan pemilihan model ikan pada alur pemilihan fitur, sehingga model dapat dimuat dan dijalankan dengan benar. (menjawab rumusan 2)
3. Antarmuka halaman utama berhasil diperbarui menjadi tata letak yang dapat digulir dengan navigasi bawah (Peta, Fitur, Gallery), sehingga mampu menampung seluruh fitur baru secara rapi. (menjawab rumusan 3)

7.2. Saran

Saran untuk pengembangan lebih lanjut adalah sebagai berikut:

1. Kunci API Google Maps sebaiknya tidak ditulis langsung pada AndroidManifest.xml, melainkan disimpan melalui variabel lingkungan atau mekanisme secret dan dibatasi restriksinya, guna menjaga keamanan.
2. Backend klasifikasi suara sebaiknya diakses melalui HTTPS pada lingkungan produksi, menggantikan usesCleartextTraffic yang hanya cocok untuk pengembangan.

3. Basis pengetahuan (brainfile) chatbot dapat terus diperkaya dengan variasi pertanyaan untuk meningkatkan cakupan jawaban.
4. Model klasifikasi gambar dan suara dapat ditingkatkan akurasinya dengan penambahan data latih, serta diperluas untuk mencakup lebih banyak spesies.

DAFTAR PUSTAKA

[1] Cikanindi, V. (2021). Penerapan Universal Sentence Encoder (USE) untuk Location-Aware Chatbot pada Aplikasi Layanan Wisata Kebun Binatang. Tugas Akhir, Departemen Teknik Informatika, Fakultas Teknologi Elektro dan Informatika Cerdas, Institut Teknologi Sepuluh Nopember, Surabaya.

[2] Kusumo, N. Y. (2023). Pengembangan Fitur Klasifikasi Hewan pada Aplikasi Wisata Kebun Binatang dengan Algoritma CNN. Tugas Akhir, Departemen Teknik Informatika, Fakultas Teknologi Elektro dan Informatika Cerdas, Institut Teknologi Sepuluh Nopember, Surabaya.

[3] Auliya, M. (2024). Perbaikan Klasifikasi Hewan pada Aplikasi Mobile Kebun Binatang dengan Transfer Learning. Tugas Akhir, Departemen Teknik Informatika, Fakultas Teknologi Elektro dan Informatika Cerdas, Institut Teknologi Sepuluh Nopember, Surabaya.

[4] Fuadah, L. (2025). Pengembangan Fitur Klasifikasi Ikan pada Aplikasi Mobile Kebun Binatang Menggunakan Transfer Learning. Tugas Akhir, Departemen Teknik Informatika, Fakultas Teknologi Elektro dan Informatika Cerdas, Institut Teknologi Sepuluh Nopember, Surabaya.

[5] Saputra, A. (2025). Pengembangan Fitur Pengenalan Suara Hewan pada Aplikasi ZooSite Menggunakan Convolutional Neural Network (CNN). Tugas Akhir, Departemen Teknik Informatika, Fakultas Teknologi Elektro dan Informatika Cerdas, Institut Teknologi Sepuluh Nopember, Surabaya.

[6] Google. (2024). Flutter Documentation. [ONLINE] Tersedia di: https://docs.flutter.dev.

[7] TensorFlow. (2024). TensorFlow Lite Guide. [ONLINE] Tersedia di: https://www.tensorflow.org/lite.

[8] FastAPI. (2024). FastAPI Documentation. [ONLINE] Tersedia di: https://fastapi.tiangolo.com.

[9] Google Research. (2020). YAMNet: Sound Event Classification Model. [ONLINE] Tersedia di: https://tfhub.dev/google/yamnet.
[10] Martin, R. C. (2017). Clean Architecture: A Craftsman's Guide to Software Structure and Design. Prentice Hall.
[11] Pan, S. J., & Yang, Q. (2010). A Survey on Transfer Learning. IEEE Transactions on Knowledge and Data Engineering, 22(10).
[12] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep Learning. Nature, 521, 436-444.
[13] Cer, D., dkk. (2018). Universal Sentence Encoder. arXiv:1803.11175.

BIODATA PENULIS 1

Nama	: Giandra Raihan Nayottama
Tempat, Tanggal Lahir	: Jakarta , 19 Juni 2004
Jenis Kelamin	: Laki-laki
Telepon	: +6287782353445
Email	: giandraraihan1@gmail.com

AKADEMIS

Kuliah	: Departemen Teknik Informatika – FTEIC , ITS
Angkatan	: 2022
Semester	: 8 (Delapan)

BIODATA PENULIS 2

Nama	: Joanes Don Bosco Bhramantyo
Tempat, Tanggal Lahir	: Depok, 8 Maret 2004
Jenis Kelamin	: Laki-laki
Telepon	: +6281299086744
Email	: jbosco0308@gmail.com

AKADEMIS

Kuliah	: Departemen Teknik Informatika – FTEIC , ITS
Angkatan	: 2022
Semester	: 8 (Delapan)