

Perancangan Antarmuka Aplikasi *Mobile* Menggunakan Kotlin untuk Sistem Pemantauan Pemberian Makanan Kucing Otomatis Berbasis Deep Learning

1st Yosua Gery Tampubolon
Fakultas Teknik Elektro
Universitas Telkom
 Bandung, Indonesia

yosuatampubolon@student.telkomuniversity.ac.id

2nd Meta Kalista
Fakultas Teknik Elektro
Universitas Telkom
 Bandung, Indonesia

metakalista@telkomuniversity.ac.id

3rd Rifqi Muhammad Fikri
Fakultas Teknik Elektro
Universitas Telkom
 Bandung, Indonesia

rifqimff@telkomuniversity.ac.id

Abstrak — Sistem pemantauan pemberian makanan kucing otomatis berbasis deep learning dirancang untuk membantu pemilik kucing dalam memantau dan mengontrol proses pemberian makan secara efektif. Penelitian ini berfokus pada perancangan antarmuka aplikasi *mobile* berbasis Kotlin yang berperan sebagai media interaksi antara pengguna dan perangkat. Perancangan antarmuka dilakukan dengan mempertimbangkan prinsip *user-centered design* untuk memastikan kemudahan penggunaan, kenyamanan, dan efisiensi. Metode pengembangan meliputi analisis kebutuhan, pembuatan *wireframe*, implementasi menggunakan Android Jetpack, serta pengujian fungsionalitas antarmuka. Hasil pengujian menunjukkan bahwa antarmuka aplikasi mampu menampilkan data pemantauan secara real-time, termasuk status pengenalan wajah kucing, persentase sisa makanan dalam tangki penyimpanan makanan, dan opsi pengaturan perangkat. Penelitian ini diharapkan dapat menjadi referensi dalam pengembangan antarmuka aplikasi *mobile* yang terintegrasi dengan sistem Internet of Things (IoT) dan teknologi deep learning untuk pemantauan hewan peliharaan.

Kata kunci— antarmuka pengguna, aplikasi *mobile*, Kotlin, Android Jetpack, pemberian makanan kucing otomatis

I. PENDAHULUAN

Perkembangan teknologi Internet of Things (IoT) dan machine learning telah membuka peluang besar dalam menghadirkan solusi otomatis untuk berbagai aspek kehidupan, termasuk perawatan hewan peliharaan. Salah satu kebutuhan penting yang dihadapi pemilik hewan adalah memastikan proses pemberian makan dilakukan secara tepat waktu dan sesuai kebutuhan. Namun, kesibukan dan keterbatasan mobilitas sering kali menjadi kendala sehingga diperlukan suatu sistem yang mampu memantau sekaligus mengatur proses pemberian makan secara otomatis dan efisien.

Seiring dengan kemajuan teknologi, berbagai penelitian telah dilakukan untuk mengembangkan sistem pemantauan dan pemberian makan hewan berbasis IoT yang dilengkapi kamera, sensor, dan algoritma machine learning [1]. Beberapa penelitian sebelumnya memanfaatkan teknologi pengenalan wajah untuk mengidentifikasi hewan tertentu sehingga proses pemberian makan menjadi lebih terkontrol.

Aplikasi mobile menjadi komponen penting dalam sistem ini karena berfungsi sebagai antarmuka utama antara pengguna dan perangkat, memungkinkan pemantauan secara real-time dan pengaturan perangkat dari jarak jauh. Bahasa pemrograman Kotlin yang diintegrasikan dengan framework Android Jetpack menawarkan solusi pengembangan antarmuka yang modern, responsif, dan efisien, yang dapat meningkatkan pengalaman pengguna atau *user experience*.

Meskipun berbagai sistem telah berhasil dikembangkan, masih terdapat tantangan dalam menciptakan antarmuka yang intuitif, mudah digunakan, dan terintegrasi secara optimal dengan sistem IoT dan deep learning. Banyak antarmuka yang belum sepenuhnya mengakomodasi prinsip *user-centered design* [2], sehingga informasi yang disajikan tidak tersampaikan secara efektif kepada pengguna. Oleh karena itu, diperlukan penelitian yang fokus pada perancangan dan implementasi antarmuka aplikasi mobile yang mampu menampilkan informasi pemantauan secara jelas, akurat, dan real-time[3].

Penelitian ini bertujuan untuk merancang dan mengimplementasikan antarmuka aplikasi mobile berbasis Kotlin yang terintegrasi dengan sistem pemantauan pemberian makanan kucing otomatis berbasis deep learning. Aplikasi ini dirancang untuk menampilkan status pengenalan wajah kucing, persentase sisa makanan, serta menyediakan fitur pengaturan perangkat melalui desain yang sederhana namun informatif. Dengan pendekatan ini, diharapkan pengguna dapat mengakses informasi dengan mudah, mengontrol perangkat secara efisien, serta mendapatkan pengalaman penggunaan yang optimal.

II. KAJIAN TEORI

A. Android Studio

Android Studio adalah Integrated Development Environment (IDE) resmi untuk pengembangan aplikasi Android, yang didukung oleh Google. IDE ini menyediakan berbagai fitur seperti code editor, emulator, integrasi dengan sistem version control, dan dukungan penuh untuk bahasa Kotlin serta Java (Android Developers, 2023). Android Studio juga dilengkapi dengan Android Jetpack, kumpulan library, alat, dan panduan arsitektur yang membantu pengembangan aplikasi lebih cepat dan konsisten.

B. Bahasa Pemrograman Kotlin

Kotlin merupakan bahasa pemrograman modern yang dirancang untuk bekerja secara penuh dengan Java, namun menawarkan sintaks yang lebih ringkas dan aman dari error null pointer. Kotlin menjadi bahasa resmi untuk pengembangan Android sejak 2017 (JetBrains, 2023). Dibandingkan Java, Kotlin mendukung fitur seperti lambda expression, extension function, dan null safety yang memudahkan pengembangan aplikasi.

C. Android Jetpack

Android Jetpack adalah sekumpulan library, tool, dan panduan arsitektur yang disediakan Google untuk mempermudah pengembangan aplikasi Android yang tangguh, konsisten, dan kompatibel dengan berbagai versi Android (Android Developers, 2023). Komponen-komponen utama Jetpack mencakup Architecture Components (ViewModel, LiveData, Room), UI Components (Navigation, RecyclerView), dan Foundation Components (AppCompat, Android KTX). Penggunaan Jetpack membantu memisahkan logika bisnis dari logika UI, sehingga kode lebih bersih dan terstruktur.

D. Front-End pada Aplikasi Android

Front-end dalam pengembangan aplikasi Android merujuk pada bagian antarmuka pengguna (user interface) yang berinteraksi langsung dengan pengguna. Desain front-end harus mempertimbangkan aspek estetika, kemudahan navigasi, dan responsivitas terhadap berbagai ukuran layar (Nielsen, 2020). Dalam Android Studio, front-end biasanya didefinisikan menggunakan XML untuk layout dan dikombinasikan dengan logika Kotlin untuk mengatur interaksi pengguna.

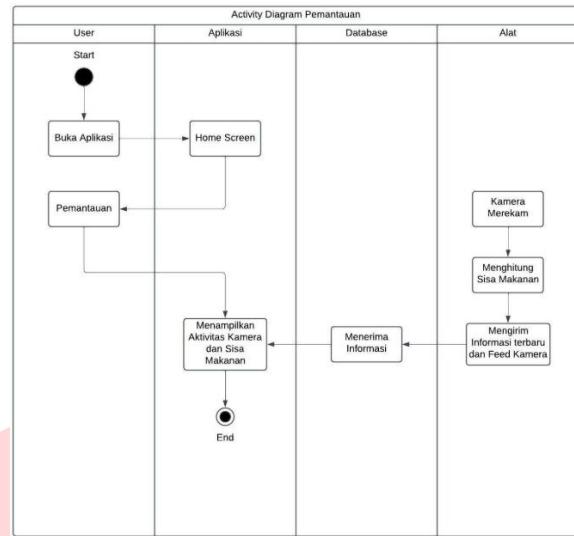
III. METODE

Metode penelitian ini menggunakan pendekatan *Research and Development* (R&D) untuk merancang dan mengimplementasikan sistem pemantauan pemberian makanan kucing secara otomatis berbasis Android. Sistem dirancang agar pemilik hewan peliharaan dapat memantau dan mengontrol pemberian makan kucing secara otomatis melalui aplikasi mobile yang terintegrasi dengan perangkat feeder. Proses perancangan dilakukan melalui beberapa tahap berikut:

A. Analisis Kebutuhan Berdasarkan Fitur-Fitur

1. Fitur Pemantauan

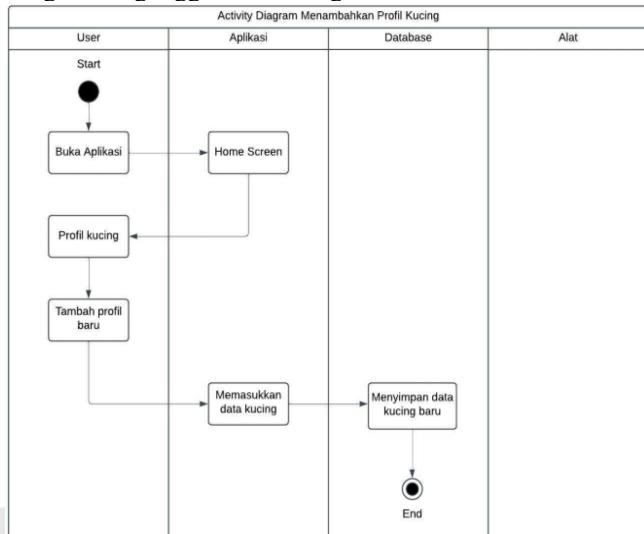
Memungkinkan pengguna memantau hewan peliharaan melalui kamera secara real time.



GAMBAR 1.

2. Fitur Penambahan Data Kucing

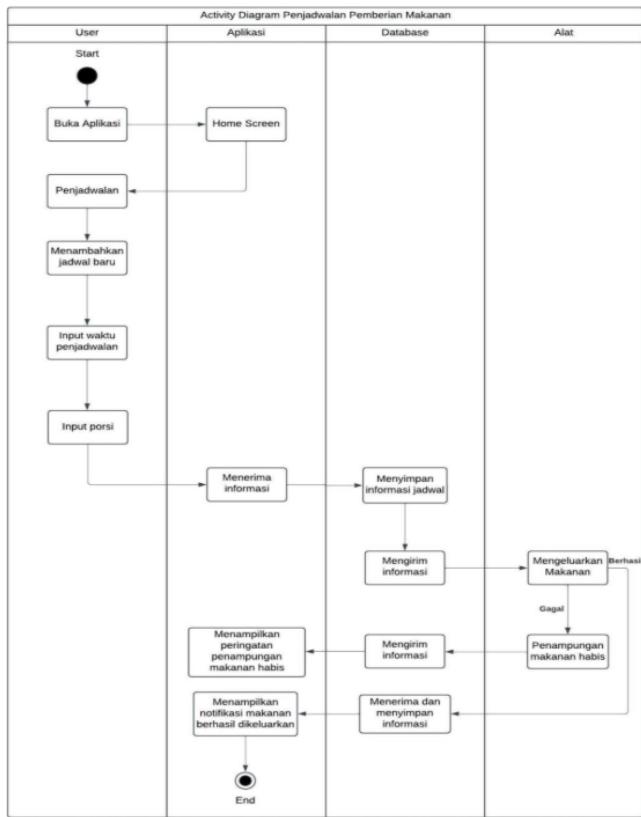
Memudahkan pengguna menambahkan profil kucing dengan menggunakan foto sebagai identifikasi.



GAMBAR 2.

3. Fitur Penjadwalan

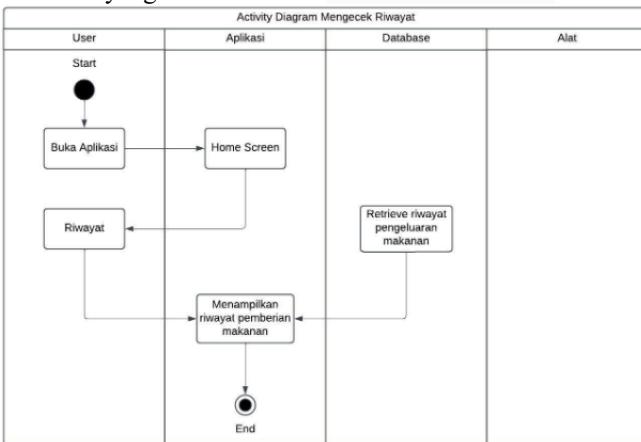
Fitur ini bertujuan untuk menambahkan jadwal makanan melalui aplikasi yang telah dibuat.



GAMBAR 3.

4. Fitur Riwayat

Fitur ini berfungsi untuk menampilkan riwayat pemberian makanan kucing yang telah dilaksanakan sesuai jadwal yang telah ditambahkan dan akan menampilkan tanggal pemberian, waktu pemberian serta berapa banyak makanan yang telah dikeluarkan.



GAMBAR 4.

B. Perancangan Sistem

Proses perancangan dimulai dari perancangan antarmuka sampai perancangan front end. Kemudian dari hasil tersebut dilakukan survei dan pengujian aplikasi yang telah dirancang. Adapun jumlah responden yang melakukan survei dan pengujian aplikasi sebanyak 10 orang.

Perancangan meliputi:

1. Perancangan Arsitektur Aplikasi

Menggunakan bahasa pemrograman Kotlin dengan framework Android Jetpack.

2. Perancangan Basis Data

Untuk penyimpanan data pengguna dan pengaturan perangkat menggunakan *cloud storage* (Google Cloud).

3. Perancangan Antarmuka (*ui/ux*)

Disesuaikan dengan *use case* dan *user flow*.

C. Implementasi Sistem

Implementasi dilakukan dengan:

1. Frontend menggunakan Kotlin dan Android Jetpack (Fragment, RecyclerView, SharedPreferences).
2. Backend untuk pengelolaan data dan integrasi perangkat feeder.
3. Integrasi modul kamera untuk real-time monitoring kucing.
4. Penyimpanan data di cloud database dan pengaturan autentifikasi pengguna.

D. Pengujian Sistem

Pengujian dilakukan menggunakan metode Black Box Testing pada setiap fungsionalitas aplikasi berdasarkan skenario pada use case. Aspek-aspek yang diuji meliputi:

1. Proses login.
2. Fungsi penjadwalan makan.
3. Monitoring live camera.
4. Pengaturan aplikasi.

E. Evaluasi

Evaluasi dilakukan untuk mengumpulkan data-data mengenai kepuasan pengguna dalam desain antarmuka dan fungsionalitas fitur-fitur yang tersedia pada aplikasi untuk mengukur kinerja aplikasi dan tingkat kepuasan pengguna.

Berikut data-data berdasarkan hasil survey pengguna terhadap aplikasi yang telah dibuat:

TABEL 1.
User Acceptance Test

Jawaban	Bobot
Sangat Setuju (SS)	5
Setuju (S)	4
Netral(N)	3
Tidak Setuju(TS)	2
Sangat Tidak Setuju(STS)	1

TABEL 2.
User Acceptance Test Smart Pet Feeder

User Acceptance Test Smart Pet Feeder								
No	Pertanyaan	STS	TS	N	S	SS	Total	Rata-rata
1	Apakah tata letak dan desain aplikasi menarik secara visual ?	0	0	2	4	4	42	4.2
2	Apakah informasi yang ada pada aplikasi mudah dipahami?	0	0	2	2	6	44	4.4
3	Menurut anda, aplikasi ini memiliki fitur yang sudah lengkap ?	0	0	1	4	5	44	4.4
4	Apakah aplikasi berfungsi dengan baik ?	0	0	1	3	6	44	4.4
5	Apakah menambahkan penjadwalan baru mudah untuk dilakukan ?	0	0	0	4	6	46	4.6
6	Apakah informasi pada menu pemantauan sudah jelas ?	0	0	1	3	6	45	4.5
7	Apakah menambahkan profil kucing baru mudah untuk dilakukan	0	0	1	3	6	45	4.5
8	Apakah waktu respons aplikasi (misalnya saat berpindah menu atau memuat data) sudah memuaskan?	0	0	1	3	6	45	4.5
9	Apakah notifikasi (jadwal pemberian, level makanan rendah) muncul tepat waktu dan informatif?	0	0	3	2	5	42	4.2
10	Secara keseluruhan, seberapa puas Anda menggunakan aplikasi Smart Pet Feeder ini?	0	0	1	3	6	45	4.5
	?							

Menggunakan persamaan untuk menghitung nilai rata-rata berdasarkan hasil survei untuk kepuasan penggunaan antarmuka aplikasi;

$$\text{Nilai Rata-rata} = \frac{\text{Total Rata-Rata}}{\text{Jumlah Data}}$$

Dari hasil UAT yang disajikan diatas, ditemukan bahwa responden sudah setuju bahwa aplikasi yang dibuat sudah mendapatkan nilai rata-rata sebesar 4.42, sehingga dapat dianggap bahwa aplikasi yang dibuat sudah layak untuk digunakan. Nilai terendah yang didapatkan adalah nilai untuk desain visual aplikasi dan juga ketepatan responsifitas

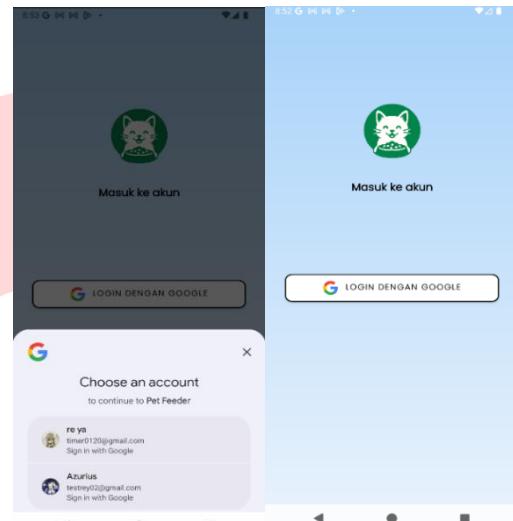
notifikasi yang masih dapat diperbaiki kembali agar lebih optimal.

IV. HASIL DAN PEMBAHASAN

A. Hasil Desain Antarmuka Front End

1. Fitur Login

Fitur login menggunakan akun Google pada aplikasi memudahkan pengguna untuk masuk tanpa membuat akun baru. Gambar 1(A) menunjukkan halaman awal login dengan tombol "Login dengan Google", sedangkan Gambar 1(B) menampilkan daftar akun Google yang tersedia pada perangkat.

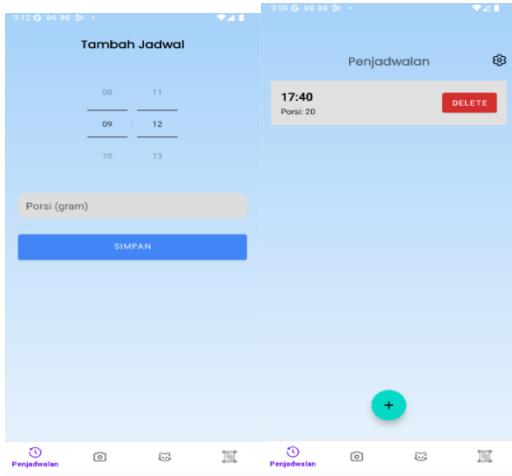


GAMBAR 5(A)

Fitur ini memberikan kemudahan akses, keamanan tinggi melalui sistem autentikasi Google, serta efisiensi dalam pengembangan aplikasi. Pengujian menunjukkan proses login berjalan cepat (± 3 detik) dan stabil di berbagai perangkat Android, sehingga mendukung pengalaman pengguna yang lebih baik.

2. Fitur Penjadwalan

Fitur penjadwalan memungkinkan pengguna mengatur waktu dan porsi makan kucing secara otomatis. Pengguna dapat menambah jadwal melalui form pengaturan jam dan porsi, lalu menyimpannya. Jadwal yang tersimpan ditampilkan dalam daftar dengan opsi hapus, memudahkan pengelolaan pemberian makan secara teratur.

GAMBAR 6
(B)

3. Fitur Pemantauan

Fitur pemantauan memungkinkan pengguna melihat kondisi kucing secara real-time melalui kamera yang terintegrasi. Sistem dilengkapi dengan deteksi objek untuk mengenali kucing secara otomatis, seperti yang ditunjukkan dengan label dan akurasi pada layar. Selain itu, terdapat indikator persentase sisa makanan di tangki penyimpanan, sehingga pengguna dapat memantau ketersediaan pakan tanpa harus memeriksa secara langsung.

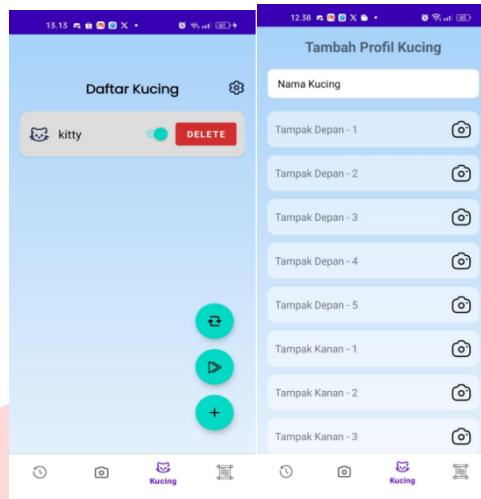
GAMBAR 7
(A)

4. Fitur Penambahan Daftar Kucing

Fitur penambahan data kucing dirancang untuk memungkinkan pengguna mendaftarkan profil kucing secara detail. Pengguna dapat memasukkan nama kucing serta mengunggah foto dari berbagai sudut pandang (tampak depan, tampak kanan dan kiri) untuk meningkatkan akurasi identifikasi sistem. Pada halaman daftar kucing, setiap profil yang telah tersimpan dapat dikelola dengan opsi menghapus atau mengaktifkan profil. Selain itu, tersedia tombol untuk menambah profil baru, memperbarui data, atau memulai interaksi dengan kucing yang terdaftar pada Gambar 4(B).

Fitur ini memberikan kemudahan dalam mengelola data hewan peliharaan dan menjadi bagian penting dari sistem

Smart Pet Feeder untuk penyesuaian pemberian pakan yang lebih personal.

GAMBAR 8
(A)

5. Fitur Riwayat

Fitur riwayat menampilkan catatan waktu dan porsi pakan yang telah terlaksana oleh sistem. Informasi disajikan secara kronologis, memudahkan pemilik untuk memantau jadwal makan dan jumlah pakan yang dikeluarkan. Fitur ini membantu evaluasi pola makan kucing serta memastikan konsistensi pemberian pakan.

Riwayat	
8/7/2025	17:40
20/6/2025	16:02
20/6/2025	16:31
20/6/2025	16:36
20/6/2025	15:58
20/6/2025	16:19
19/6/2025	15:17
19/6/2025	14:56
19/6/2025	15:34
19/6/2025	15:37
18/6/2025	17:26
18/6/2025	16:58
17/6/2025	10:02
17/6/2025	18:03

GAMBAR 9
(A)

B. Pengujian Black-box

Pengujian dilakukan menggunakan metode black-box, di mana pengujinya tidak mempertimbangkan bagaimana sistem bekerja secara internal, melainkan hanya berfokus pada input dan output untuk mengevaluasi fungsionalitas sistem. Pendekatan ini dirancang untuk mengidentifikasi apakah sistem memberikan respons yang sesuai dalam berbagai kondisi penggunaan yang mungkin terjadi.

Ruang lingkup pengujian mencakup fitur login, pemantauan, pengisian penjadwalan, hasil penjadwalan, riwayat, dan pengaturan. Tujuan dari pengujian ini adalah untuk memastikan bahwa sistem berjalan sesuai dengan spesifikasi dan menghasilkan keluaran yang benar dalam

berbagai skenario. Untuk menjaga validitas hasil, setiap keluaran akan dibandingkan dengan hasil yang diharapkan berdasarkan skenario uji yang telah ditetapkan[4].

1. Pengujian Login

TABEL 2
(a)

No	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Mengklik tombol login dengan Google	Sistem akan menampilkan popup akun Google	Sesuai harapan	Valid
2	Memilih akun Google yang aktif	Sistem akan memverifikasi akun dan mengalihkan pengguna ke halaman pemantauan	Sesuai harapan	Valid
3	Menutup popup akun Google	Sistem akan tetap berada di halaman login	Sesuai harapan	Valid
4	Koneksi internet terputus saat verifikasi akun Google	Sistem akan membatalkan verifikasi dan tetap berada di halaman login	Sesuai harapan	Valid

2. Pengujian Pemantauan

TABEL 3
(a)

No	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Mengklik tombol "Mulai Pemantauan"	Sistem akan mengirim instruksi untuk mengaktifkan alat dan menampilkan livestream dari kamera Smart Pet Feeder	Sesuai harapan	Valid
2	Mengklik tombol "Mulai Pemantauan" lebih dari sekali secara beruntun	Sistem tetap mengirim instruksi namun alat hanya memproses instruksi yang pertama saja dan tidak error	Sesuai harapan	Valid
3	Tetap berada di halaman pemantauan setelah menekan tombol "Mulai Pemantauan"	Sistem akan menampilkan tayangan langsung dari kamera smart pet feeder secara real-time	Sesuai harapan	Valid
4	Berpindah ke halaman lain kemudian kembali lagi ke halaman pemantauan	Sistem akan tetap menampilkan tayangan langsung dari kamera secara real-time.	Sesuai harapan	Valid

3. Pengujian Pengisian Penjadwalan

TABEL 4
(a)

No	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Memilih waktu dan mengisi porsi makanan lalu mengklik tombol "Simpan"	Sistem akan menyimpan jadwal dan menampilkannya di halaman hasil penjadwalan	Sesuai harapan	Valid
2	Memilih waktu namun tidak mengisi porsi makanan lalu mengklik tombol "Simpan"	Sistem akan menolak dan menampilkan pesan "Porsi tidak boleh kosong"	Sesuai harapan	Valid
3	Memilih waktu dan mengisi porsi makanan lalu mengklik tombol "Simpan" lebih dari sekali secara beruntun	Sistem hanya akan menyimpan 1 jadwal dan menampilkannya di halaman hasil penjadwalan	Sesuai harapan	Valid

4. Pengujian Hasil Penjadwalan

TABEL 5
(a)

No	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Mengklik tombol "Delete" di kotak jadwal yang tersimpan	Sistem akan menampilkan pesan konfirmasi untuk menghapus jadwal	Sesuai harapan	Valid
2	Mengklik tombol "Ya" pada pesan konfirmasi penghapusan jadwal	Sistem akan menghapus jadwal	Sesuai harapan	Valid
3	Mengklik tombol "Batal" pada pesan konfirmasi penghapusan jadwal	Sistem tidak menghapus jadwal	Sesuai harapan	Valid

5. Pengujian Riwayat

TABEL 6
(a)

No	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Waktu pemberian makanan sudah tiba sesuai jadwal yang terdaftar pada saat aplikasi dibuka	Sistem akan menampilkan notifikasi tentang pemberian makanan, kemudian menyimpannya ke dalam riwayat	Sesuai harapan	Valid
2	Waktu pemberian makanan sudah tiba sesuai jadwal yang terdaftar pada saat aplikasi ditutup	Sistem akan tetap menampilkan notifikasi tentang pemberian makanan, kemudian menyimpannya ke dalam riwayat	Sesuai harapan	Valid

6. Pengujian Pengaturan

TABEL 7
(a)

No	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Menekan tombol "Shutdown"	Sistem akan menampilkan pesan konfirmasi untuk melakukan shutdown pada Smart Pet Feeder	Sesuai harapan	Valid
2	Menekan tombol "Ya" pada pesan konfirmasi shutdown	Sistem akan mengirim instruksi shutdown ke Smart Pet Feeder	Sesuai harapan	Valid
3	Menekan tombol "Batal" pada pesan konfirmasi shutdown	Sistem akan menutup pesan konfirmasi dan tidak mengirim instruksi apapun	Sesuai harapan	Valid
4	Menekan tombol "Restart"	Sistem akan menampilkan pesan konfirmasi untuk melakukan restart pada Smart Pet Feeder	Sesuai harapan	Valid
5	Menekan tombol "Ya" pada pesan konfirmasi restart	Sistem akan mengirim instruksi restart ke Smart Pet Feeder	Sesuai harapan	Valid
6	Menekan tombol "Batal" pada pesan konfirmasi restart	Sistem akan menutup pesan konfirmasi dan tidak mengirim instruksi apapun	Sesuai harapan	Valid
7	Menekan tombol "Logout"	Sistem akan menampilkan pesan konfirmasi untuk logout dari aplikasi	Sesuai harapan	Valid
8	Menekan tombol "Ya" pada pesan konfirmasi logout	Sistem akan menghapus sesi login dan mengalihkan pengguna ke halaman login	Sesuai harapan	Valid
9	Menekan tombol "Batal" pada pesan konfirmasi logout	Sistem akan menutup pesan konfirmasi dan sesi login tetap aktif	Sesuai harapan	Valid

Pengujian black-box berhasil mencapai tingkat keberhasilan 100% sesuai spesifikasi yang ditetapkan, tanpa terdeteksi anomali, kegagalan fungsional, maupun deviasi perilaku sistem. Hasil ini menegaskan bahwa aplikasi telah beroperasi secara stabil dan memenuhi seluruh kebutuhan

fungsional, sehingga siap untuk diimplementasikan dan digunakan oleh pengguna.

V. KESIMPULAN

Berdasarkan hasil pembahasan, seluruh fitur pada aplikasi, meliputi penjadwalan pemberian pakan, pemantauan, pengelolaan profil hewan peliharaan, dan pencatatan riwayat, telah berfungsi dengan baik sesuai rancangan. Aplikasi mampu memberikan kemudahan bagi pengguna dalam mengatur jadwal pemberian pakan, memantau kondisi hewan secara langsung, serta menyimpan data historis secara terstruktur. Dengan tampilan antarmuka yang sederhana dan responsif, sistem ini dinilai efektif dalam membantu pemilik hewan peliharaan mengelola kebutuhan pakan secara teratur dan akurat.

Pengujian dilakukan menggunakan metode black-box, yang berfokus pada evaluasi input dan output tanpa mempertimbangkan proses internal sistem. Lingkup pengujian meliputi fitur login, pemantauan, pengisian penjadwalan, hasil penjadwalan, riwayat, dan pengaturan. Tujuan pengujian adalah memastikan seluruh fungsi berjalan sesuai spesifikasi dan menghasilkan keluaran yang benar pada berbagai skenario penggunaan. Berdasarkan pengujian yang telah dilakukan, sistem mencapai tingkat keberhasilan 100% tanpa adanya anomali, kegagalan fungsional, maupun deviasi perilaku. Hasil ini membuktikan bahwa aplikasi beroperasi secara stabil, memenuhi seluruh kebutuhan fungsional, dan siap diimplementasikan untuk penggunaan nyata

REFERENSI

- [1] O. E. Castillo-Arceo, R. U. Renteria-Flores, and P. C. Santana-Mancilla, “Design and development of a smart pet feeder with IoT and Deep learning,” in Proceedings of the 11th International Electronic Conference on Sensors and Applications, Nov. 26–28, 2024. [Online]. Available: <https://sciforum.net/paper/view/20487>. [Accessed: Aug. 12, 2025].
- [2] A. S. Mutia, “UI UX design with User-Centered Design method for mobile-based E-learning application to enhance learning experience (Case study: LKP ASVINA),” Jurnal Informatika dan Teknologi Komputer (J-ICOM), vol. 5, no. 2, pp. 140–145, Aug. 2024. [Online]. Available: <https://ejurnalunsam.id/index.php/jicom/article/view/11148>. [Accessed: Aug. 12, 2025].
- [3] L. Punchoojit and N. Hongwarittorrn, “Usability studies on mobile user interface design patterns: A systematic literature review,” Advances in Human-Computer Interaction, vol. 2017, p. 6787504, Jun. 2017. [Online]. Available: <https://doi.org/10.1155/2017/6787504>. [Accessed: Aug. 12, 2025].
- [4] S. Nidhra and J. S. Dondeti, “Black Box and White Box Testing Techniques – A Literature Review,” International Journal of Embedded Systems and Applications, vol. 2, no. 2, pp. 29–50, Jun. 2012