PENGEMBANGAN APLIKASI *NUTRISCAN* UNTUK DETEKSI BUAH DAN SAYURAN

SKRIPSI

ALFI SYAHRIN AL QODAR
2022230038

PROGRAM STUDI INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS PRABUMULIH
2026

PENGEMBANGAN APLIKASI *NUTRISCAN* UNTUK DETEKSI BUAH DAN SAYURAN

Untuk Memenuhi Salah Satu Syarat Menyusun Proposal Skripsi
Program Studi Informatika Universitas Prabumulih

ALFI SYAHRIN AL QODAR
2022230038

PROGRAM STUDI INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS PRABUMULIH
2026

HALAMAN PENGESAHAN

PENGEMBANGAN APLIKASI *NUTRISCAN* UNTUK DETEKSI BUAH DAN SAYURAN

SKRIPSI

Diajukan Sebagai Salah Satu Syarat Untuk Mendapatkan Gelar Sarjana Program Studi Informatika Universitas Prabumulih

Oleh :

Alfi Syahrin Al Qodar
2022230038

Prabumulih, 27 Juli 2026

Pembimbing I	Pembimbing II
(Nur Aini Hutagalung, S.Kom., M.Si., M.Kom.)	(Hermizahadiwidastra, S.Kom., M.Kom.)
NUPTK. 2434764 664300022	NUPTK. 2863762663130192

Mengetahui,
Ketua Program Studi Informatika

(Iwan Setiawan, S. Kom.,M. Kom)
NUPTK. 1541769670130303

HALAMAN PERSETUJUAN

Skripsi ini dengan judul "Pengembangan Aplikasi *NutriScan* untuk Deteksi Buah dan Sayuran" telah di pertahankan di hadapan Tim Penguji Skripsi Program Studi Informatika Fakultas Ilmu Komputer Universitas Prabumulih pada tanggal 28 Juli 2026.

Prabumulih, 28 Juli 2026
Tim Penguji Tugas Akhir

Ketua:

Nur Aini Hutagalung, S.Kom., M.Si., M.Kom. (.)
NUPTK. 2434764 664300022

Anggota:

Hermizahadiwidastra, S.Kom., M.Kom. ( )
NUPTK. 2863762663130192

Muchlis, S.Kom., M.Si ( )
NUPTK. 5339756658200053

Mengetahui,

Dekan Fakultas Ilmu Komputer	Ketua Program Studi Informatika
(Khana Wijaya, S.Kom., M.Kom.)	(Iwan Setiawan, S.Kom., M.Kom.)
NUPTK. 7146768669130363	NUPTK. 1541769670130303

PERNYATAAN INTEGRITAS

Saya yang bertanda tangan di bawah ini

Nama : Alfi Syahrin Al Qodar

NIM : 2022230038

Program Studi : Informatika

Menyatakan dengan sebenarnya bahwa, skripsi ini benar-benar merupakan hasil karya saya, dan tidak terdapat karya orang lain, apabila di kemudian hari terbukti atau dapat dibuktikan bahwa tugas akhir ini hasil *plagiasi*, maka saya bersedia menerima sanksi sesuai dengan ketentuan yang berlaku atas perbuatan tersebut.



Prabumulih, 28 Juli 2026

Yang membuat pernyataan



Alfi Syahrin Al Qodar

NIM. 2022230038

ABSTRAK

PENGEMBANGAN APLIKASI *NUTRISCAN* UNTUK DETEKSI BUAH DAN SAYURAN

Alfi Syahrin Al Qodar, 2022230038, 2026

Penelitian ini mengembangkan sebuah sistem deteksi kondisi jeruk berbasis *machine learning* menggunakan *google ml kit* yang diintegrasikan dalam aplikasi *flutter*. Sistem dirancang untuk memudahkan pengguna mengidentifikasi kualitas jeruk secara cepat melalui gambar yang diambil dari kamera maupun galeri. Dengan pendekatan *prototype*, pengembangan dilakukan secara iteratif agar hasil akhir sesuai kebutuhan pengguna sekaligus efisien dalam implementasi. Metode yang diterapkan meliputi pengumpulan dataset gambar buah jeruk, pra-pemrosesan data, serta pengujian *model* klasifikasi untuk memastikan akurasi deteksi.

Kata Kunci: Deteksi Jeruk, *Google ML Kit*, *Flutter*, *Prototype*, Klasifikasi Gambar, *Firebase*, *Live Scan*.

ABSTRACT

DEVELOPMENT OF THE NUTRISCAN APP FOR DETECTING FRUITS AND VEGETABLES

Alfi Syahrin Al Qodar, 2022230038, 2026

This research develops a machine learning-based orange condition detection system using google ml kit integrated into a flutter application. The system is designed to allow users to quickly identify orange quality through images taken from the camera or gallery. Using a prototype approach, development is carried out iteratively to ensure the final result meets user needs and is efficient in implementation. The applied methods included collecting a dataset of citrus fruit images, pre-processing the data, and testing the classification model to ensure detection accuracy.

Keywords: Orange Detection, Google ML Kit, Flutter, Prototype, Image Classification, Firebase, Live Scan.

KATA PENGANTAR

Puji dan syukur penulis ucapkan kehadirat Allah SWT atas limpahan rahmat dan karunia-nya, sehingga penulis dapat menyelesaikan Skripsi yang berjudul Pengembangan Aplikasi *NutriScan* untuk Deteksi Buah dan Sayuran, Proposal ini merupakan salah satu syarat untuk mendapatkan gelar Sarjana di Program Studi S1 Informatika Fakultas Ilmu Komputer Universitas Prabumulih. Penulis banyak menerima bantuan, arahan, dan bimbingan dari berbagai pihak. Penulis mengucapkan terima kasih kepada:

1. Ibu Dr. Yuniar Pratiwi, S.Si., M.Si selaku Rektor Universitas Prabumulih.
2. Bapak Andi Christian selaku Wakil Rektor Universitas Prabumulih.
3. Bapak Khana Wijaya, S.Kom., M.Kom selaku Dekan Fakultas Ilmu Komputer Univesitas Prabumulih sekaligus Dosen Pembimbing Akademik.
4. Ibu Suhartini, S.Kom., M.Kom selaku Wakil Dekan Keuangan Fakultas Ilmu Komputer Univesitas Prabumulih.
5. Ibu Nur Aini Hutagalung, S.Kom., M.Si., M.Kom Selaku Wakil Dekan Kemahasiswaan Fakultas Ilmu Komputer sekaligus Dosen Pembimbing I yang telah sabar memberikan bimbingan, arahan, serta membantu peneliti untuk mengembangkan pemikiran dalam menyelesaikan tugas skripsi sehingga proses penulisan dan penelitian ini bisa selesai tepat waktu.
6. Bapak Iwan Setiawan, S.Kom., M.Kom selaku Ketua Program Studi Informatika Universitas Prabumulih.
7. Bapak Hermizahadiwidastra, S.Kom., M.Kom dan selaku Dosen Pembimbing II yang telah sabar dalam memberikan bimbingan, petunjuk, arahan, dan membantu peneliti dalam mengembangkan pemikiran dalam penulisan sehingga peneliti dapat menyelesaikan tugas skripsi ini.
8. Bapak Muchlis, S. Kom., M.Si selaku dosen penguji yang telah memberikan saran, masukan, dan arahan dalam penyempurnaan skripsi ini.
9. Bapak dan Ibu staf administrasi Fakultas Ilmu Komputer Universitas Prabumulih yang telah membantu penulis dalam proses administrasi selama penyusunan skripsi ini.

Penulis menyadari bahwa skripsi ini masih memiliki kekurangan. Oleh karena itu, penulis mengharapkan kritik dan saran yang membangun demi penyempurnaan skripsi ini.

Prabumulih, 28 Juli 2026

Penulis

Alfi Syahrin Al Qodar

NIM. 2022230038

DAFTAR ISI

Halaman

HALAMAN JUDUL ..... **i**
HALAMAN PENGESAHAN ..... **ii**
LEMBAR PERSETUJUAN ..... **iii**
PERNYATAAN INTEGRITAS ..... **iv**
ABSTRAK ..... **v**
ABSTRACT ..... **vi**
KATA PENGANTAR ..... **vii**
DAFTAR ISI ..... **ix**
DAFTAR TABEL ..... **xii**
DAFTAR GAMBAR ..... **xiii**
DAFTAR LAMPIRAN ..... **xv**
DAFTAR SINGKATAN ..... **xvi**
BAB I (PENDAHULUAN) 1
1.1. Latar Belakang 1
1.2. Rumusan Masalah 3
1.3. Batasan Masalah 3
1.4. Tujuan Penelitian 3
1.5. Manfaat Penelitian 4
1.6. Sistematika Penelitian 4
BAB II LANDASAN TEORI **6**
2.1. Pengembangan Sistem 6
2.2. *Flutter* 7
2.3. *Dart* 8
2.4. *Firebase CLI* 9
2.5 *Google ML Kit* 10
2.6. *Machine Learning* 11
2.7. *Prototype* 12
2.8. *Chatbot* 13

2.9. Buah dan Sayuran 14
2.10. Penelitian Terdahulu 15
2.11. Kerangka Berpikir Penelitian 19
BAB III METODE PENELITIAN 22
3.1. Objek Penelitian dan Jadwal Penelitian 22
3.1.1. Objek Penelitian 22
3.1.2. Subjek Penelitian 23
3.1.3. Jadwal Penelitian 24
3.2. Metode Penelitian 24
3.3. Sumber Data 27
3.4. Teknik Pengumpulan Data 29
3.5. Metode Pengembangan Sistem 31
3.6. Alat Bantu Perancangan 34
3.7. Pengujian Sistem 41
BAB IV ANALISA DAN PERANCANGAN 45
4.1. Analisa Sistem 45
4.1.1. Analisa dan Evaluasi Sistem Berjalan 45
4.1.2. Analisa Kebutuhan 47
4.2. Perancangan Sistem (Desain) yang diusulkan 52
4.2.1. Perancangan Sistem (Logis) 52
4.2.2. Perancangan *Database* (Fisik) 64
4.2.3. Perancangan Antarmuka (UI/UX) 67
BAB V HASIL DAN PEMBAHASAN 78
5.1. Spesifikasi Perangkat 78
5.1.1. Perangkat Keras yang Digunakan 78
5.1.2. Perangkat Lunak yang Digunakan 79
5.2. Implementasi *Database* 80
5.2.1. Implementasi Koleksi users 80
5.2.2. Implementasi Koleksi *detections* 81
5.2.3. Implementasi Koleksi *chat_sessions* 82
5.2.4. Implementasi Koleksi *chat_messages* 82
5.3. Tampilan Antarmuka (UI) 83

5.3.1. Halaman Intro (*SplashScreen*) 83
5.3.2. Halaman Login 84
5.3.3. Halaman Registrasi (*SignUp)* 85
5.3.4. Halaman Utama (*HomePage)* 86
5.3.5. Halaman Deteksi (Kamera & Galeri) 87
5.3.6. Halaman Hasil Deteksi (*ResultPage*) 88
5.3.7. Halaman *Chatbot* (*NutriBot*) 90
5.3.8. Halaman Riwayat Deteksi (*History Page*)..... 91
5.4. Hasil Pengujian Sistem 92
5.4.1. Pengujian Fungsional (*Black Box Testing*) 92
5.4.2. Pengujian Akurasi Deteksi..... 97
5.4.3. Pengujian Kecepatan Respons Sistem 100
5.4.4. Pembahasan Hasil Pengujian 101
BAB VI KESIMPULAN DAN SARAN..... 103
6.1. Kesimpulan 103
6.2. Saran 104
DAFTAR PUSTAKA..... 106
LAMPIRAN..... 112

DAFTAR TABEL

Halaman

Tabel 2.1 Tabel Penelitian Terdahulu 16
Tabel 3.1 Jadwal Penelitian 24
Tabel 3.2 *Flowchart* Proses Aplikasi Deteksi Buah dan Sayuran 25
Tabel 3.3 Tabel *Use Case* Diagram Aplikasi Deteksi Buah dan Sayuran 34
Tabel 4.1 Evaluasi Sistem Berjalan 46
Tabel 4.2 Kebutuhan Pengguna 47
Tabel 4.3 Kebutuhan Data Sistem NutriScan 48
Tabel 4.4 Kebutuhan Fungsional Sistem NutriScan 49
Tabel 4.5 Kebutuhan Non-Fungsional 50
Tabel 4.6 Spesifikasi Perangkat Keras Minimum untuk Pengguna 51
Tabel 4.7 Spesifikasi Perangkat Keras Pengembangan 51
Tabel 4.8 Spesifikasi Perangkat Lunak Pengembangan 51
Tabel 4.9 Koleksi users – Data Pengguna 65
Tabel 4.10 Koleksi detections – Riwayat Hasil Deteksi 65
Tabel 4.11 Koleksi *chat_sessions* – Sesi Percakapan *Chatbot* 66
Tabel 4.12 Koleksi *chat_messages* – Pesan dalam Sesi *Chatbot* 67
Tabel 5.1 Perangkat Keras yang Digunakan 78
Tabel 5.2 Spesifikasi Perangkat Keras Pengujian (Test Device) 78
Tabel 5.3 Spesifikasi Perangkat Lunak yang Digunakan 79
Tabel 5.4 Hasil Pengujian *Black Box Testing* 92
Tabel 5.5 Hasil Pengujian Akurasi Deteksi Buah 98
Tabel 5.6 Hasil Pengujian Akurasi Deteksi Sayuran 98
Tabel 5.7 Rekap Hasil Pengujian Akurasi 99
Tabel 5.8 Hasil Pengujian Kecepatan Respons Sistem 100
Tabel 5.9 Rekap Hasil Pengujian Kecepatan Respons 100
Tabel 5.10 Ringkasan Evaluasi Kinerja Aplikasi *NutriScan* 101

DAFTAR GAMBAR

Halaman

Gambar 2.1. Kearngka Berpikir Penelitian 20
Gambar 3.1. Model Konseptual *Prototype* 33
Gambar 3.2. *Flowchart* Sistem 36
Gambar 3.3. Hasil Perancangan Peneliti (2026) 39
Gambar 3.4. Skema Pengujian Fungsional Aplikasi FreshScan 43
Gambar 4.1 *Use Case* Diagram Aplikasi *NutriScan* 53
Gambar 4.2 *Use Case* Diagram pada *User* di Aplikasi *NutriScan* 55
Gambar 4.3 *Use Case* Diagram pada *Firebase* di Aplikasi *NutriScan* 56
Gambar 4.4 *Use Case* Diagram pada *Google* ML Kit di Aplikasi *NutriScan* 57
Gambar 4.5 *Activity* Diagram – Proses Deteksi Buah dan Sayuran 58
Gambar 4.6 *Activity* Diagram – Proses Percakapan *Chatbot NutriBot* 59
Gambar 4.7 *Sequence* Diagram – Proses *Login* 61
Gambar 4.8 *Sequence* Diagram – Deteksi Objek via Kamera 62
Gambar 4.9 *Sequence* Diagram – Proses *Chatbot NutriBot* 63
Gambar 4.10 *Class* Diagram Aplikasi *NutriScan* 64
Gambar 4.11 Tampilan Halaman *Splash Screen* Aplikasi *NutriScan* 68
Gambar 4.12 Tampilan Halaman *Login* Aplikasi *NutriScan* 69
Gambar 4.13 Tampilan Halaman Registrasi Aplikasi *NutriScan* 70
Gambar 4.14 Tampilan Halaman Utama Aplikasi *NutriScan* 71
Gambar 4.15 Tampilan Halaman Deteksi Aplikasi *NutriScan* 72
Gambar 4.16 Tampilan Halaman Hasil Deteksi Aplikasi *NutriScan* 74
Gambar 4.17 Tampilan Halaman *Chatbot NutriBot* Aplikasi *NutriScan* 76
Gambar 4.18 Tampilan Halaman Riwayat Deteksi Aplikasi *NutriScan* 77
Gambar 5.1 Struktur Koleksi *users* pada *Firebase Console* 80
Gambar 5.2. Struktur Koleksi *detections* pada *Firebase Console* 81
Gambar 5.3. Struktur Koleksi *chat_sessions* pada *Firebase Console* 82
Gambar 5.4. Struktur Koleksi *chat_messages* pada *Firebase Console* 83
Gambar 5.5. Tampilan Halaman *Splash Screen* Aplikasi *NutriScan* 84

Gambar 5.6. Tampilan Halaman *Login* Aplikasi *NutriScan*..... 85
Gambar 5.7. Tampilan Halaman Registrasi Aplikasi *NutriScan*..... 86
Gambar 5.8. Tampilan Halaman Utama Aplikasi *NutriScan* 87
Gambar 5.9. Tampilan Halaman Deteksi saat Proses Berlangsung 88
Gambar 5.10. Tampilan Halaman Hasil Deteksi Aplikasi *NutriScan* 89
Gambar 5.11 Tampilan Halaman *Chatbot NutriBot*..... 90
Gambar 5.12 Tampilan Halaman Riwayat Deteksi Aplikasi *NutriScan* 91

DAFTAR LAMPIRAN

Halaman

Lampiran 1 Dokumentasi Penelitian Buah-buahan di Lokasi Pasar Lama Tanjung Enim, Kab. Muara Enim 113
Lampiran 2 Dokumentasi Penelitian Sayur-sayuran di Lokasi Pasar Baru Tanjung Enim, Kab. Muara Enim 116

UNIVERSITAS PRABUMULIH
UNPRA
2000

DAFTAR SINGKATAN

1.	AI	: Artificial Intelligence
2.	API	: Application Programming Interface
3.	App	: Application
4.	CLI	: Command Line Interface
5.	CNN	: Convolutional Neural Network
6.	Confident Score	: Nilai tingkat kepercayaan hasil deteksi objek
7.	Dataset	: Kumpulan data untuk pelatihan atau pengujian sistem
8.	Detection System	: Sistem pendeteksian objek
9.	Firebase	: Platform Backend-as-a-Service dari Google
10.	Firebase CLI	: Alat berbasis command line untuk mengelola layanan Firebase
11.	Firestore	: Database NoSQL berbasis cloud dari Firebase
12.	Flutter	: Framework UI open-source untuk pengembangan aplikasi multiplatform
13.	Framework	: Kerangka kerja pengembangan perangkat lunak
14.	Laggy	: Kondisi aplikasi atau sistem yang mengalami keterlambatan, patah-patah, atau tidak responsif
15.	Image Labeling	: Proses pemberian label pada objek dalam citra
16.	Live Scan	: Proses pemindaian objek secara real-time menggunakan kamera
17.	Machine Learning (ML)	: Metode pembelajaran mesin berbasis data
18.	ML Kit	: Google Machine Learning Kit
19.	Mobile Application	: Aplikasi yang berjalan pada perangkat mobile
20.	Mobile Vision	: Teknologi pengolahan citra pada perangkat mobile

21.	Object Detection	: Teknik pendeteksian dan pengenalan objek
22.	On-device Processing	: Pemrosesan data langsung di perangkat pengguna
23.	Prototype	: Metode pengembangan sistem berbasis purwarupa
24.	R&D	: Research and Development
25.	Real-time	: Pemrosesan atau respon secara langsung
26.	Smartphone	: Perangkat komunikasi pintar berbasis sistem operasi
27.	Test Image	: Gambar uji untuk pengujian sistem
28.	UI	: User Interface
29.	UX	: User Experience
30.	Black Box Testing	: Metode pengujian sistem berdasarkan fungsi tanpa melihat kode internal
31.	SBC	: Single Board Computer
32.	YOLO	: You Only Look Once
33.	Deep Learning	: Metode pembelajaran mesin berbasis jaringan saraf dalam untuk analisis data kompleks.
34.	SSD	: Single Shot Detector.
35.	R-CNN	: Region-based Convolutional Neural Network.

BAB I
PENDAHULUAN

1.1. Latar Belakang

Perkembangan teknologi informasi dan komunikasi yang semakin pesat telah memberikan dampak signifikan terhadap beragam aspek kehidupan manusia. Penggunaan perangkat *mobile* tidak lagi sebatas sarana penghubung, tetapi juga menjadi *platform* utama untuk aplikasi inovatif yang mendukung aktivitas sehari-hari, termasuk edukasi, kesehatan, dan pengenalan objek secara otomatis. Fenomena ini menunjukkan bahwa kebutuhan terhadap aplikasi yang mampu memanfaatkan kecerdasan buatan (*artificial intelligence, AI*) semakin meningkat.

Buah dan sayuran merupakan komponen penting dalam pemenuhan kebutuhan gizi masyarakat, namun pada praktiknya masih banyak pengguna yang mengalami kesulitan dalam mengenali jenis buah dan sayuran secara cepat dan akurat. Kesulitan ini umumnya dialami oleh pelajar, anak-anak, maupun masyarakat umum yang belum memiliki pengetahuan cukup mengenai karakteristik *visual* masing-masing objek. Proses identifikasi manual cenderung lambat, Perkembangan teknologi *computer vision* telah memungkinkan pengenalan objek dilakukan secara otomatis menggunakan perangkat *mobile*.

Permasalahan utama dalam pengembangan sistem deteksi objek pada perangkat *mobile* terletak pada keterbatasan performa perangkat, kebutuhan respons *real-time*, serta kemudahan interaksi pengguna dalam memahami hasil identifikasi. Selain itu, hasil deteksi yang hanya berupa teks *visual* terkadang belum cukup membantu seluruh pengguna, khususnya dalam aspek aksesibilitas dan interaktivitas. Oleh karena itu, diperlukan aplikasi yang tidak hanya mampu melakukan identifikasi objek secara cepat, tetapi juga menyediakan penyampaian informasi yang lebih komunikatif melalui dukungan fitur tambahan seperti *voice output*.

Berdasarkan kajian penelitian terdahulu, sebagian besar penelitian terkait deteksi objek masih berfokus pada pengembangan *model* klasifikasi citra, optimasi algoritma *deep learning*, atau implementasi deteksi berbasis *platform* tertentu secara terpisah. Selain itu, masih terbatas penelitian yang mengombinasikan

framework flutter dengan *google ml kit* untuk kebutuhan deteksi buah dan sayuran berbasis *on-device processing*. Kondisi ini menunjukkan adanya *gap riset*, yaitu belum optimalnya integrasi antara deteksi objek *real-time*, antarmuka *mobile* modern, serta fitur interaktif berbasis *audio* dan *chatbot* dalam satu aplikasi terpadu.

Kebaruan (*novelty*) dalam penelitian ini terletak pada pengembangan aplikasi *mobile* bernama *nutriscan* (sebelumnya *fruittag)* yang mengintegrasikan beberapa teknologi dalam satu sistem terpadu. Penelitian ini tidak hanya memanfaatkan *flutter* sebagai *framework* lintas *platform* dan *google ml kit* sebagai teknologi *image labeling*, tetapi juga menambahkan fitur *voice output* untuk menyampaikan hasil identifikasi secara *audio* serta *chatbot* sebagai media interaksi lanjutan bagi pengguna. Kombinasi fitur tersebut menjadikan aplikasi tidak hanya berfungsi sebagai alat deteksi objek, tetapi juga sebagai media edukatif yang lebih komunikatif dan mudah digunakan.

Urgensi penelitian ini didasarkan pada kebutuhan akan solusi teknologi yang praktis, efisien, dan mudah diakses untuk membantu masyarakat dalam mengenali buah dan sayuran secara cepat. Di tengah meningkatnya penggunaan *smartphone*, aplikasi berbasis kecerdasan buatan yang mampu berjalan secara *real-time* dan *on-device* menjadi solusi yang relevan untuk kondisi penggunaan sehari-hari. Selain mendukung efisiensi identifikasi objek, aplikasi ini juga berpotensi meningkatkan literasi teknologi dan pemahaman masyarakat terhadap pemanfaatan *computer vision*.

Berdasarkan uraian di atas, penelitian ini berfokus pada pengembangan aplikasi *mobile* bernama *nutriscan* berbasis *flutter* yang memanfaatkan *google ml kit* untuk mendeteksi dan mengklasifikasikan buah serta sayuran secara otomatis melalui perangkat *smartphone*. Pengembangan aplikasi ini tidak hanya pada kemampuan deteksi objek secara *real-time* dengan tingkat kepercayaan (*confidence score*) tertentu, tetapi juga menghadirkan fitur pendukung berupa *voice output* dan *chatbot* untuk meningkatkan interaktivitas serta kemudahan pengguna dalam memahami hasil identifikasi yang praktis, edukatif, efisien, dan relevan dengan kebutuhan masyarakat di era digital.

1.2. Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan sebelumnya, maka rumusan masalah dalam penelitian ini dapat dirumuskan sebagai berikut: Bagaimana mengembangkan aplikasi *mobile* berbasis *flutter* yang mampu mendeteksi dan mengenali jenis buah dan sayuran menggunakan *google ml kit*, lalu proses integrasi *image labeling api* dari *google ml kit* ke dalam *project flutter* agar sistem dapat melakukan deteksi objek secara *real-time*, hingga sejauh mana tingkat akurasi dan kecepatan deteksi aplikasi dalam mengenali buah dan sayuran pada berbagai kondisi gambar atau pencahayaan?

1.3. Batasan Masalah

Agar penelitian ini terarah dan tidak meluas dari fokus utama, maka ditetapkan beberapa batasan masalah sebagai berikut:

1. Aplikasi hanya dikembangkan untuk *platform android* (minimal versi 8 sampai versi terbaru) menggunakan bahasa pemrograman *dart* & menggunakan *framework flutter*, tanpa pengujian pada sistem operasi *ios*.
2. Fitur deteksi objek hanya difokuskan pada buah dan sayuran, tanpa mencakup objek lain seperti hewan, makanan olahan, atau benda *non-organik*.
3. Sistem deteksi menggunakan *image labeling api* dari *google ml kit*, tanpa melakukan pelatihan ulang (*custom training model*) menggunakan *dataset* baru.

1.4. Tujuan Penelitian

Tujuan dari penelitian ini adalah sebagai berikut:

1. Mengembangkan aplikasi *mobile* berbasis *flutter* yang mampu melakukan deteksi dan klasifikasi buah serta sayuran menggunakan *google ml kit*.
2. Menerapkan dan mengintegrasikan *image labeling api* dari *google ml kit* ke dalam proyek *flutter* agar dapat melakukan proses identifikasi objek secara *real-time* dan *on-device*.
3. Menguji kinerja aplikasi berdasarkan tingkat akurasi deteksi dan kecepatan respons sistem.

4. Aplikasi menyediakan fitur *output* suara (*voice output*) untuk menyampaikan hasil identifikasi jenis buah dan sayuran, tanpa analisis lanjutan berbasis bahasa alami.

1.5. Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan manfaat baik secara teoritis maupun praktis sebagai berikut:

1. Manfaat Teoritis:
   a. memberikan kontribusi terhadap pengembangan ilmu pengetahuan di bidang kecerdasan buatan (*artificial intelligence*), khususnya dalam penerapan *machine learning* dan *computer vision* pada perangkat *mobile*.
   b. menjadi referensi ilmiah bagi penelitian selanjutnya yang berfokus pada integrasi *google ml kit* dan *flutter* untuk pengembangan aplikasi deteksi objek berbasis citra digital.
2. Manfaat Praktis:
   a. Membantu masyarakat, khususnya pelajar, anak-anak, maupun konsumen, dalam mengenali berbagai jenis buah dan sayuran secara cepat, akurat, dan praktis melalui perangkat *mobile*.
   b. Menjadi solusi alternatif dalam pembelajaran interaktif berbasis teknologi yang dapat meningkatkan literasi digital di bidang pengenalan bahan pangan.
   c. Memberikan pengalaman langsung bagi peneliti dalam menerapkan teknologi *mobile development* dan *artificial intelligence*, yang dapat menjadi bekal profesional di bidang teknologi informasi.

1.6. Sistematika Penelitian

Sistematika penelitian ini disusun untuk memberikan gambaran yang jelas mengenai alur dan isi dari penelitian yang dilakukan. Adapun sistematika dalam penyusunan penelitian ini terdiri atas beberapa bab sebagai berikut:

Bab I – Pendahuluan

Bab ini berisi latar belakang masalah, rumusan masalah, batasan masalah, tujuan

penelitian, manfaat penelitian, serta sistematika penelitian. Bagian ini memberikan gambaran umum mengenai alasan dilaksanakannya penelitian dan arah penelitian yang akan dilakukan.

Bab II – Tinjauan Pustaka

Bab ini memuat teori-teori yang relevan dengan penelitian, seperti konsep dasar kecerdasan buatan (*artificial intelligence*), pengenalan citra digital, *google ml kit*, *flutter*, serta hasil-hasil penelitian terdahulu yang mendukung penyusunan kerangka berpikir penelitian ini.

Bab III – Metodologi Penelitian

Bab ini menjelaskan metode yang digunakan dalam penelitian, meliputi desain penelitian, alat dan bahan, tahapan pengembangan sistem, alur kerja (*workflow*), serta teknik pengumpulan dan analisis data.

Bab IV – Analisis dan Perancangan

Pada bab ini, penulis akan menguraikan tentang analisis dan perancangan terhadap masalah yang diangkat, tahap analisis diperlukan sebagai dasar dalam perancangan sistem yang akan di bangun.

Bab V – Hasil dan Pembahasan

Bab ini terdiri dari gambaran hasil penelitian dan analisa. Baik dari secara kualitatif, kuantitatif dan statistik, serta pembahasan hasil penelitian.

Bab VI – Kesimpulan dan Saran

Bab ini berisi kesimpulan dan saran dari seluruh penelitian yang telah dilakukan. Kesimpulan dapat di kemukakan masalah yang ada pada penelitian serta hasil dari penyelesaian penelitian yang bersifat analisis objetif. Sedangkan saran mencantumkan jalan keluar untuk mengatasi masalah dan kelemahan yang ada. Saran ini tidak lepas ditunjukkan untuk ruang lingkup penelitian.

BAB II
TINJAUAN PUSTAKA

2.1. Pengembangan Sistem

Pengembangan sistem merupakan proses merancang, membangun, dan menyempurnakan perangkat lunak agar sesuai dengan kebutuhan pengguna. Proses ini mencakup analisis kebutuhan, perancangan, implementasi, pengujian, hingga pemeliharaan untuk memastikan sistem berjalan efektif dan efisien. Pendekatan pengembangan modern menekankan pentingnya pemahaman kebutuhan pengguna sejak awal agar solusi yang dibangun tepat sasaran dan mampu berfungsi optimal dalam lingkungan operasional (Lorenzoni et al., 2021).

Pengembangan sistem merupakan rangkaian aktivitas sistematis yang difokuskan pada pembentukan perangkat lunak baru ataupun peningkatan sistem yang sudah ada. Tahapan tersebut meliputi perancangan komponen sistem, implementasi teknologi, pengujian fungsi utama, dan evaluasi performa agar perangkat lunak yang dihasilkan sesuai spesifikasi. *Model* pengembangan semacam ini memastikan bahwa sistem tidak hanya berjalan tetapi juga mudah diperbarui dan skalabel sesuai kebutuhan perkembangan teknologi (Rachma & Muhlas, 2022).

Pendekatan lain yang relevan dalam pengembangan sistem adalah pengembangan berbasis *prototyping*. *Model* ini memungkinkan proses membangun sistem dilakukan secara bertahap melalui evaluasi berulang hingga tercapai bentuk yang ideal. Dalam konteks penelitian ini, pendekatan tersebut digunakan untuk merancang dan mengembangkan aplikasi pendeteksi buah serta sayuran berbasis *flutter* dengan dukungan *google ml kit* yang bertugas sebagai *engine* klasifikasi visual secara otomatis melalui *image labeling* (Ayu et al., 2021).

Berdasarkan uraian para ahli, dapat disimpulkan bahwa pengembangan sistem pada penelitian ini merupakan proses bertahap yang mencakup perancangan antarmuka, integrasi modul pendeteksi gambar, pengujian performa aplikasi di perangkat *android*, serta evaluasi hasil untuk memastikan akurasi klasifikasi. Seluruh prosedur ini dilakukan agar aplikasi deteksi buah dan sayuran dapat berfungsi secara responsif, akurat, dan layak diimplementasikan pada kebutuhan pengguna secara *real-time*.

2.2. *Flutter*

Flutter merupakan *framework* pengembangan aplikasi *mobile multiplatform* yang dikembangkan oleh *google* menggunakan bahasa pemrograman *dart. Framework* ini mengusung konsep *widget-based architecture* sehingga setiap elemen antarmuka dibangun menggunakan komponen yang fleksibel dan dapat dikustomisasi sesuai kebutuhan pengembangan aplikasi. *Model* tersebut memungkinkan proses desain *ui* menjadi lebih seragam, konsisten, serta mudah dikembangkan secara progresif dalam skala kecil maupun besar (Kusnetsoff, 2021).

Selain menawarkan struktur *ui* yang stabil, *flutter* juga dikenal efisien dalam pengembangan aplikasi *modern* berkat adanya dukungan pustaka bawaan, *hot reload*, dan kemampuan untuk dikompilasi secara *native* ke *android, ios,* maupun *web. Flutter* dirancang untuk mempercepat pembuatan antarmuka responsif dengan kinerja tinggi, sehingga pengembang dapat membangun aplikasi yang kaya interaksi tanpa memerlukan rendering manual pada setiap komponen visual. Hal ini menjadikan *flutter* salah satu *framework* yang paling banyak digunakan untuk pengembangan aplikasi *ai* dan *computer-vision* pada perangkat *mobile* (Azahari et al., 2022).

Dalam lingkup rekayasa perangkat lunak, *flutter* dapat diintegrasikan dengan berbagai *api* dan layanan eksternal untuk memperluas fungsi sistem. Penggunaan *framework modern* seperti *flutter* berperan penting dalam efisiensi perancangan perangkat lunak karena mendukung arsitektur modular yang memudahkan proses implementasi, debugging, dan pengujian. *Flutter* sangat relevan digunakan dalam penelitian berbasis deteksi *visual* karena dukungannya terhadap integrasi *machine learning on-device*, kamera, serta pengolahan gambar secara *real-time* dalam satu ekosistem pengembangan (Fajri & Fitria, 2023).

Dalam lingkup rekayasa perangkat lunak, *flutter* dapat diintegrasikan dengan berbagai *api* dan layanan eksternal untuk memperluas fungsi sistem. Penggunaan *framework modern* seperti *flutter* berperan penting dalam efisiensi perancangan perangkat lunak karena mendukung arsitektur modular yang memudahkan proses implementasi, *debugging*, dan pengujian. Hal ini selaras bahwa *flutter* sangat relevan digunakan dalam penelitian berbasis deteksi *visual*

karena dukungannya terhadap integrasi *machine learning on-device*, kamera, serta pengolahan gambar secara *real-time* dalam satu ekosistem pengembangan.

2.3. *Dart*

Dart merupakan bahasa pemrograman utama yang digunakan dalam pengembangan aplikasi *flutter*. Bahasa ini dirancang dengan sintaks yang ringkas, mudah dipahami, serta mendukung *object-oriented programming*, sehingga cocok untuk pengembangan aplikasi *mobile* modern dengan struktur kode yang terorganisasi. *Dart* dikembangkan oleh *google* untuk menciptakan bahasa yang efisien dan cepat dalam *runtime execution*, sekaligus tetap fleksibel pada tahap pengembangan aplikasi. Hal ini membuat *dart* menjadi fondasi penting dalam proses kompilasi dan *rendering ui* secara mulus pada berbagai perangkat (Haider, 2021).

Kelebihan utama *dart* terletak pada kemampuannya melakukan kompilasi ke *native machine code* dan *javascript*, memungkinkan aplikasi berjalan optimal di *mobile* maupun *web* tanpa perlu perubahan signifikan pada kode program. Dukungan *dart* terhadap *ahead-of-time (AOT)* dan *just-in-time (JIT) compilation* menjadikannya unggul dalam kecepatan *startup* aplikasi dan pengembangan *real-time update* menggunakan *hot reload*. Kombinasi dua mode kompilasi ini memberikan pengalaman *coding* yang lebih dinamis, terutama dalam proyek yang membutuhkan proses iterasi *ui* cepat seperti *image processing* (Gulzar, 2023).

Dart juga menyediakan *asynchronous programming model* berbasis *future* dan *async–await* yang sangat membantu dalam pemrosesan data *visual*, termasuk pengambilan gambar, pemanggilan *api*, hingga proses klasifikasi menggunakan mesin pembelajaran. Dukungan *asynchronous* merupakan elemen teknis yang krusial dalam pengembangan aplikasi *ai* dan *mobile vision* karena mencegah *ui* menjadi *laggy* saat pemrosesan data intensif berlangsung. Pemanfaatan *dart* dalam penelitian ini diwujudkan melalui integrasi *flutter* dan *google ml kit* untuk mendeteksi buah serta sayuran secara *on-device* dengan respons cepat (Ikhsan et al., 2025).

Dalam konteks penelitian ini, *dart* berperan sebagai bahasa inti yang menghubungkan antarmuka *flutter* dengan modul deteksi gambar dan pemrosesan

machine learning. Dengan kompilasi cepat, dukungan *asynchronous*, serta arsitektur yang ramah pengembangan, *dart* memungkinkan sistem deteksi berjalan ringan dan responsif di perangkat *mobile*. Kesimpulannya, *dart* bukan hanya bahasa pemrograman pendukung, tetapi merupakan pondasi logika aplikasi yang memastikan seluruh fitur berjalan optimal dan stabil.

2.4. *Firebase CLI*

Firebase cli merupakan *command line interface* yang disediakan oleh *firebase* untuk mempermudah pengembangan, pengujian, dan *deployment* layanan *backend* pada aplikasi *mobile* maupun *web*. Melalui *firebase cli*, pengembang dapat melakukan proses autentikasi proyek, men-*deploy hosting*, mengaktifkan *cloud functions*, mengelola *firestore database*, hingga melakukan pengujian lokal sebelum sistem dipublikasikan. *Firebase cli* dirancang sebagai alat utama untuk *workflow* pengembangan modern karena memungkinkan proses *deploy* hanya dengan satu perintah, cepat, dan efisien (Rahmawati & Sari, 2024).

Firebase cli juga mendukung sinkronisasi proyek secara langsung dengan *firebase console*, sehingga perubahan yang dilakukan pada struktur data, konfigurasi autentikasi, maupun layanan penyimpanan bisa diterapkan secara *real-time* tanpa pengaturan manual. Pemanfaatan *command-based tools* seperti *firebase cli* mempercepat integrasi sistem berbasis *cloud*, terutama dalam aplikasi yang memerlukan koneksi *server* ringan dan respons cepat pada perangkat *mobile*. Hal ini sangat relevan dalam pengembangan sistem deteksi buah dan sayuran yang memerlukan penyimpanan *log* hasil, aktivitas pengguna, serta integrasi *chatbot* (Abdi, 2024).

Selain itu, *firebase cli* menyediakan fitur *emulator suite* yang memungkinkan pengujian *firestore*, *authentication*, *hosting*, hingga *functions* secara *offline* di perangkat pengembang. Dengan adanya simulasi lokal ini, proses *debugging* lebih fleksibel dan aman karena tidak mempengaruhi data asli pada *server*. Mekanisme *local emulation* memberikan keuntungan signifikan bagi aplikasi riset berbasis *ai*, sebab pengujian dapat dilakukan berulang sebelum *model* ataupun *database* dirilis ke produksi (Sarker, 2021).

Berdasarkan uraian tersebut dapat disimpulkan bahwa *firebase cli* adalah komponen kunci dalam proses pengembangan *backend* aplikasi, terutama ketika diperlukan sistem yang ringan, cepat, serta mudah dikontrol melalui terminal. Dalam penelitian ini, *firebase cli* digunakan sebagai *platform deployment* dan pengelolaan *database* untuk mendukung penyimpanan hasil klasifikasi buah dan sayuran, integrasi *chatbot*, serta memastikan aplikasi dapat dikembangkan dan diuji secara efisien.

2.5. *Google ML Kit*

Google ml kit merupakan serangkaian *api machine learning* yang dirancang untuk berjalan secara *on-device*, sehingga mampu melakukan pemrosesan dan klasifikasi gambar tanpa koneksi *server*. Teknologi ini mendukung berbagai fungsi seperti *image labeling, object detection, face detection, text recognition,* hingga *barcode scanning* yang dapat diimplementasikan langsung pada aplikasi *android* maupun *ios*. Keunggulan utama *ml kit* terletak pada kemampuannya memproses citra secara lokal dengan latensi rendah sehingga sangat cocok diterapkan untuk aplikasi *real-time* (Tariq, 2025).

Dalam pengembangan aplikasi *mobile vision*, *google ml kit* banyak dimanfaatkan untuk mendukung proses identifikasi objek berbasis kamera *smartphone* secara efisien. Integrasi teknologi ini memungkinkan aplikasi melakukan analisis *visual* secara cepat tanpa membebani proses pengembangan dengan pelatihan model yang kompleks. Penggunaan *api* siap pakai seperti *image labeling* membantu pengembang mempercepat implementasi sistem deteksi objek pada perangkat *mobile*. Pendekatan ini sangat relevan bagi aplikasi yang membutuhkan respons cepat dan kemudahan integrasi teknologi *visual* (Batubara et al., 2025).

Ml kit juga mendukung pemrosesan *on-device hybrid*, di mana fungsi dasar dapat dijalankan *offline* sementara fitur lanjutan dapat dikombinasikan dengan *cloud ml* untuk meningkatkan akurasi. Arsitektur *machine learning* berbasis perangkat langsung memberikan keuntungan signifikan dalam efisiensi, privasi pengguna, serta konsumsi sumber daya yang rendah. Hal ini menjadikan *ml kit valid*

sebagai teknologi yang stabil dan *scalable* untuk aplikasi klasifikasi buah dan sayuran menggunakan kamera *mobile* (Rahayu Dewi et al., 2021).

Berdasarkan pemaparan tersebut, *google ml kit* dapat disimpulkan sebagai solusi *machine learning* ringan namun efektif untuk kebutuhan deteksi gambar berbasis *smartphone*. Dalam penelitian ini, *ml kit* digunakan sebagai mesin klasifikasi utama yang melakukan *labeling* objek buah dan sayuran secara *real-time* melalui kamera maupun gambar unggahan, sehingga mendukung tujuan sistem yang praktis, cepat, dan mudah diakses pengguna.

2.6. *Machine Learning*

Machine learning (ML) merupakan cabang dari kecerdasan buatan yang memungkinkan sistem belajar dari data tanpa harus diprogram secara eksplisit. Teknologi ini bekerja dengan mengenali pola pada *dataset,* kemudian digunakan untuk melakukan prediksi maupun pengambilan keputusan berdasarkan pola yang telah dipelajari. *Machine learning* menjadi fondasi utama dalam berbagai aplikasi modern, mulai dari sistem rekomendasi otomatis, analisis citra digital, hingga pengembangan sistem cerdas yang mampu meningkatkan performa berdasarkan data yang terus bertambah (Javaid et al., 2023).

Dalam bidang visi komputer, *Machine learning* banyak diterapkan menggunakan pendekatan *deep learning*, salah satunya melalui arsitektur *convolutional neural network* (*CNN*). *CNN* dirancang untuk mengekstraksi fitur *visual* seperti warna, tekstur, bentuk, hingga tepi objek secara otomatis sehingga sangat efektif digunakan dalam proses deteksi dan klasifikasi gambar. Kemampuan *cnn* dalam mempelajari pola visual kompleks menjadikannya cocok untuk mengenali objek biologis seperti buah dan sayuran yang memiliki karakteristik visual beragam. Pendekatan ini banyak digunakan pada sistem klasifikasi citra modern karena mampu menghasilkan tingkat akurasi yang tinggi dalam proses pengenalan objek (Khattab & Zaharia, 2020).

Penelitian mengenai klasifikasi buah menunjukkan bahwa penerapan *machine learning* berbasis *convolutional neural network* (*CNN*) mampu menghasilkan tingkat akurasi yang tinggi dalam proses identifikasi visual. *Model cnn* dapat mengenali karakteristik objek seperti warna, bentuk, dan tekstur secara

otomatis sehingga efektif digunakan untuk klasifikasi citra buah dan sayuran. Hasil penelitian memperlihatkan bahwa pendekatan *deep learning* sangat relevan diterapkan pada sistem *mobile vision*, terutama ketika dikombinasikan dengan teknologi *on-device processing* yang memungkinkan proses deteksi berjalan langsung pada perangkat pengguna (Kikuchi et al., 2022).

Berdasarkan kajian tersebut, *machine learning* memiliki peran sentral dalam pengembangan sistem deteksi buah dan sayuran. Dengan kemampuan mempelajari pola citra dan mengekstraksi fitur *visual* secara otomatis, *ml* memungkinkan aplikasi melakukan klasifikasi secara *real-time* melalui kamera *smartphone*. Konsep ini menjadi dasar utama bagi sistem yang dibangun dalam penelitian ini.

2.7. *Prototype*

Metode *prototype* merupakan pendekatan pengembangan perangkat lunak yang dilakukan dengan membangun versi awal sistem untuk memperoleh umpan balik pengguna sebelum sistem akhir dikembangkan secara penuh. *Model* ini memungkinkan proses pengembangan berlangsung secara iteratif melalui evaluasi dan penyempurnaan berulang berdasarkan kebutuhan pengguna yang ditemukan selama proses pengujian. Pendekatan *prototype* banyak digunakan dalam pengembangan aplikasi modern karena mampu meningkatkan fleksibilitas perancangan sistem serta membantu pengembang memahami kebutuhan pengguna secara lebih akurat sejak tahap awal pengembangan (Maryani et al., 2022).

Proses *prototype* umumnya terdiri atas beberapa tahapan, yaitu identifikasi kebutuhan pengguna, perancangan awal (*quick design*), pembangunan prototipe, evaluasi pengguna, serta penyempurnaan sistem berdasarkan masukan yang diperoleh selama proses pengembangan. Karakteristik iteratif ini menjadikan metode *prototype* sangat sesuai digunakan pada pengembangan aplikasi yang masih memerlukan penyesuaian kebutuhan dan perubahan fitur selama proses pembangunan sistem (Trisnadoli & Yanti, 2022).

Selain itu, berbagai penelitian menunjukkan bahwa metode *prototype* mampu mengurangi risiko kegagalan pengembangan sistem karena pengguna dapat melakukan evaluasi terhadap fungsi dan tampilan sistem sejak tahap awal. Proses

umpan balik yang dilakukan secara berulang memungkinkan pengembang melakukan perbaikan sebelum sistem memasuki tahap implementasi penuh. Dengan pendekatan tersebut, kebutuhan pengguna dapat diakomodasi dengan lebih baik sehingga kualitas sistem yang dihasilkan menjadi lebih sesuai dengan tujuan pengembangan dan memiliki tingkat penerimaan pengguna yang lebih tinggi (Hindrichsen & Eyre, 2023).

Sebagai kesimpulan, penggunaan metode *prototype* dalam penelitian ini sangat relevan karena memungkinkan pembangunan aplikasi deteksi buah dan sayuran dilakukan secara iteratif, dengan prototipe diuji langsung pada pengguna dan hasil evaluasi menjadi dasar penyempurnaan sistem. Dengan *model* ini, risiko kesalahan bisa diminimalkan dan aplikasi yang dihasilkan diharapkan lebih sesuai dengan kebutuhan nyata pengguna sebelum peluncuran *final*.

2.8. *Chatbot*

Chatbot merupakan program komputer yang dirancang untuk mensimulasikan percakapan manusia secara otomatis melalui antarmuka teks maupun suara. Teknologi ini umumnya memanfaatkan *natural language processing (NLP)* untuk memahami masukan pengguna dan menghasilkan respons yang relevan sesuai konteks percakapan. *Chatbot* digunakan untuk menyediakan informasi secara cepat, konsisten, dan *real-time* tanpa memerlukan keterlibatan manusia secara langsung. Kemampuan *chatbot* sebagai solusi efektif dalam meningkatkan aksesibilitas, interaktivitas, dan efisiensi layanan berbasis teknologi (Gökçearslan et al., 2024).

Dalam aplikasi *mobile* maupun *web* modern, *chatbot* umumnya dibangun dengan mengombinasikan teknologi *natural language processing* (*NLP*) dan algoritma pendukung untuk memproses masukan pengguna, mengenali maksud pertanyaan (*intent recognition*), serta menghasilkan respons yang sesuai berdasarkan basis pengetahuan yang tersedia. Penerapan *chatbot* memungkinkan penyampaian informasi secara otomatis dengan waktu respons yang cepat dan konsisten. Selain itu, penggunaan *chatbot* dapat meningkatkan efisiensi layanan informasi, serta mengurangi beban pekerjaan administratif (Mahmoud S Abdallah, 2025).

Di samping itu, implementasi *chatbot* dalam ranah pendidikan memerlukan perancangan yang tidak hanya berfokus pada kemampuan percakapan, tetapi juga pada dukungan terhadap proses pembelajaran yang efektif. *Chatbot* edukatif perlu dirancang dengan memperhatikan aspek penyampaian materi, pemahaman konteks bahasa, serta kemampuan memberikan umpan balik yang sesuai dengan kebutuhan pengguna. Pendekatan tersebut memungkinkan *chatbot* berfungsi sebagai media pembelajaran interaktif yang mampu meningkatkan keterlibatan pengguna dalam proses belajar (Snekha & Ayyanathan, 2023).

Dengan mempertimbangkan aspek teknis, fungsionalitas, dan etis *chatbot* menjadi komponen penting dalam sistem informasi *modern*. Untuk penelitian ini, *chatbot* akan diintegrasikan ke dalam aplikasi deteksi buah dan sayuran sebagai antarmuka interaktif bagi pengguna yang ingin memperoleh informasi atau penjelasan hasil deteksi menjadikannya sebagai fitur tambahan yang memperkaya pengalaman pengguna dan meningkatkan kegunaan aplikasi secara keseluruhan.

2.9. Buah dan Sayuran

Buah dan sayuran merupakan komoditas pangan alami yang berasal dari berbagai bagian tanaman yang dapat dikonsumsi oleh manusia. Kedua kelompok pangan ini mengandung berbagai zat gizi penting seperti vitamin, mineral, serat, serta senyawa bioaktif yang berperan dalam menjaga kesehatan tubuh. Selain memiliki nilai gizi yang tinggi, buah dan sayuran juga kaya akan *fitonutrien* yang berfungsi sebagai antioksidan alami dan membantu melindungi tubuh dari berbagai risiko penyakit. Keberagaman warna, bentuk, ukuran, dan tekstur yang dimiliki menjadikan buah dan sayuran sering digunakan sebagai objek penelitian berbasis kecerdasan buatan (Devirgiliis et al., 2024).

Selain itu, perkembangan teknologi perangkat *mobile* memungkinkan proses identifikasi buah dan sayuran dilakukan secara langsung melalui kamera *smartphone*. Pemanfaatan teknologi *computer vision* dan *machine learning* memungkinkan sistem mengenali karakteristik *visual* objek berdasarkan bentuk, warna, tekstur, dan pola tertentu. Dalam penelitian ini, aplikasi *nutriscan* dikembangkan sebagai sistem pendeteksi buah dan sayuran berbasis perangkat *mobile* yang mampu melakukan identifikasi objek secara *real-time*, menampilkan

hasil klasifikasi, serta memberikan informasi pendukung terkait objek yang terdeteksi. (Appadoo et al., 2023).

Penelitian modern memanfaatkan *dataset* citra buah dan sayuran sebagai bahan utama dalam proses pelatihan model klasifikasi berbasis *machine learning*. Keberhasilan proses klasifikasi sangat dipengaruhi oleh kualitas data citra, jumlah dan keragaman *dataset*, serta teknik augmentasi yang diterapkan selama pelatihan *model*. Selain itu, tahapan prapemrosesan (*preprocessing*) seperti normalisasi pencahayaan, *cropping*, *resizing*, dan peningkatan kontras berperan penting dalam meningkatkan kualitas citra yang digunakan. (Aharony et al., 2022).

Secara keseluruhan, buah dan sayuran merupakan objek yang sangat relevan untuk penelitian berbasis *computer vision* karena memiliki karakteristik visual yang beragam, seperti warna, bentuk, ukuran, dan tekstur yang dapat dikenali melalui teknologi pengolahan citra digital. Oleh karena itu, pemanfaatan teknologi deteksi otomatis pada buah dan sayuran menjadi landasan yang kuat dalam pengembangan aplikasi *nutriscan*, yang bertujuan membantu pengguna melakukan identifikasi objek secara cepat, akurat, dan praktis melalui perangkat *mobile*.

2.10. Penelitian Terdahulu

Kajian penelitian terdahulu merupakan proses penelaahan terhadap berbagai penelitian yang memiliki keterkaitan dengan topik yang sedang diteliti. Melalui kajian tersebut, peneliti dapat mengidentifikasi pendekatan, metode, teknologi, serta hasil penelitian yang telah dilakukan sebelumnya sehingga dapat diketahui kelebihan, keterbatasan, dan peluang pengembangan dari penelitian yang ada. Dalam konteks pengembangan aplikasi deteksi buah dan sayuran, kajian penelitian terdahulu digunakan untuk membandingkan berbagai teknologi seperti *computer vision*, *machine learning*, aplikasi *mobile*, serta fitur pendukung yang telah diterapkan pada penelitian sejenis (Topal, 2025).

Penelitian terkait deteksi objek pada perangkat *mobile* terus berkembang seiring kemajuan teknologi *computer vision* dan *machine learning*. Berbagai penelitian telah memanfaatkan model klasifikasi citra, *object detection*, serta teknologi pemrosesan berbasis perangkat (*on-device processing*) untuk menghasilkan sistem yang mampu melakukan identifikasi objek secara otomatis

melalui kamera *smartphone*. Oleh karena itu, kombinasi *flutter* sebagai *framework* lintas *platform* dan *google ml kit* sebagai teknologi *image labeling* menjadi pendekatan yang relevan dalam pengembangan aplikasi deteksi buah dan sayuran berbasis *mobile* (Bradley, 2022).

2.1. Tabel Penelitian Terdahulu

No	Peneliti & Tahun	Judul / Fokus Penelitian	Metode / Teknologi	Hasil Penelitian
1	(Batubara et al., 2025)	*Academic Chatbot Based on Natural Language Processing for Student Services*	*Natural Language Processing (NLP) + Chatbot*	*Chatbot* berbasis *nlp* mampu memberikan layanan informasi akademik secara otomatis, cepat, dan responsif sehingga membantu meningkatkan efisiensi pelayanan kepada pengguna.
2	(Mahmoud S Abdallah, 2025)	*How Conversational AI Chatbots Support and Reinforce Self-Regulated Language Learning*	*Conversational AI + Chatbot + NLP*	*Chatbot* berbasis *ai* mampu mendukung proses pembelajaran mandiri melalui interaksi yang adaptif, responsif, dan sesuai dengan kebutuhan pengguna.
3	(Ikhsan et al., 2025)	*AI Chatbot Implementation for Academic Assistance*	*NLP + Web Chatbot*	*Chatbot* meningkatkan efisiensi pelayanan info akademik & respons lebih cepat dibanding layanan manual.
4	(Tariq, 2025)	*Mobile Application Performance Testing Development*	*Mobile Application Testing + Quality Assurance Framework*	Pengujian performa aplikasi *mobile* diperlukan untuk memastikan stabilitas sistem, kecepatan respons, efisiensi penggunaan sumber daya, dan kualitas pengalaman pengguna pada berbagai kondisi penggunaan.
5	(Gökçearslan et al., 2024)	*Benefits, Challenges, and Methods of Artificial Intelligence (AI) Chatbots in Education*	*Artificial Intelligence Chatbot + Natural Language Processing (NLP)*	*Chatbot* berbasis *ai* mampu meningkatkan interaktivitas, aksesibilitas informasi, dan efektivitas layanan pembelajaran melalui respons otomatis yang cepat dan kontekstual.
6	(Rahmawati & Sari, 2024)	Aplikasi Deteksi Buah Berbasis *Mobile Vision*	*CNN + Mobile Vision*	Sistem mendeteksi kualitas buah dengan akurasi tinggi & respons cepat pada perangkat *mobile*.
7	(Abdi, 2024)	Tinjauan Langkah Pengujian Perangkat Lunak	*Black Box Testing + Equivalence Partitioning*	Efektif untuk menguji fungsi sistem berdasarkan *input–output* tanpa melihat kode program, sehingga mampu menemukan kesalahan fungsional secara efisien.
8	(Saputra et al., 2023)	*Object Detection* Untuk Mendeteksi Citra Buah-Buahan	*YOLO + Object Detection*	Metode *yolo* mampu mendeteksi dan mengidentifikasi objek buah secara otomatis dengan tingkat akurasi yang baik melalui

		Menggunakan Metode *Yolo*		pemrosesan citra *digital* secara *real-time*.
9	(Appadoo et al., 2023)	*FruVegy: An Android App for the Automatic Identification of Fruits*	*Computer Vision + Machine Learning + Android Application*	Aplikasi *android* mampu mengidentifikasi buah dan sayuran secara otomatis menggunakan teknologi *computer vision* dan *machine learning* melalui kamera perangkat *mobile*.
10	(Snekha & Ayyanathan, 2023)	*An Educational CRM Chatbot for Learning Management System*	*Educational Chatbot + Learning Management System (LMS)*	*Chatbot* edukatif mampu membantu penyampaian informasi dan meningkatkan interaksi pengguna dalam proses pembelajaran melalui layanan otomatis yang responsif.
11	(Topal, 2025)	*Research on Voscreen in English Language Education*	*Systematic Literature Review (SLR) + Content Analysis*	Kajian literatur membantu mengidentifikasi tren penelitian, kesenjangan penelitian (*research gap*), serta peluang pengembangan penelitian pada bidang yang diteliti.
12	(Fajri & Fitria, 2023)	Pengembangan *Real-time Object Detection*	*Computer Vision + (CNN) + (SBC)*	Sistem deteksi objek mampu berjalan secara *real-time* pada perangkat *single-board computer* dengan tingkat akurasi yang baik dan waktu respons yang relatif cepat
13	(Gulzar, 2023)	*Fruit Image Classification Model*	*MobileNetV2 + Deep Transfer Learning Technique*	Dengan *transfer learning* mampu meningkatkan akurasi klasifikasi citra buah secara signifikan dengan *model* yang ringan dan efisien untuk perangkat *mobile.*
14	(Widya Ningsih & Habibah Nurfauziah, 2023)	Perbandingan *Model Waterfall* dan Metode *Prototype* untuk Pengembangan Aplikasi pada Sistem Informasi	*Waterfall* dan *Prototype*	Metode *prototype* lebih fleksibel dibanding *waterfall* karena evaluasi dan perbaikan sistem secara bertahap berdasarkan pengguna.
15	(Azahari et al., 2022)	*The Development of an Online Food Ordering System for JomMakan Restaurant*.	*Web-based System + Database + Prototype*	Sistem pemesanan makanan *online* mempermudah proses pemesanan dan meningkatkan efisiensi operasional restoran.
16	(Rachma & Muhlas, 2022)	*Comparison of Models in Research And Development (R&D)*	*Waterfall + Prototype + Android Application*	Metode *prototype* lebih fleksibel dan efektif untuk pengembangan aplikasi *android* dibandingkan *waterfall*.
17	(Zaidi et al., 2022)	*A survey of Modern Deep Learning Based Object Detection Models*	*Deep Learning + CNN + Object Detection Models (YOLO, SSD, R-CNN)*	*Model deep learning* efektif untuk deteksi objek dengan akurasi tinggi dan mendukung proses *real-time*.

18	(Haider, 2021)	Evaluation of cross-platform technology Flutter from the user's perspective	*Flutter + Cross-Platform Mobile Development*	*Flutter* mampu mendukung pengembangan aplikasi lintas *platform* dengan antarmuka yang konsisten, performa yang baik, serta efisiensi pengembangan dibandingkan pembuatan aplikasi secara terpisah pada setiap *platform.*
19	(Sarker, 2021)	*Machine Learning Techniques for Image Recognition*	*CNN / Supervised ML*	*Ml modern* lebih akurat dibanding metode klasik & mampu meningkatkan deteksi *visual* signifikan.
20	(Kusnetsoff, 2021)	*Mobile Real-Time Object Detection with Flutter*	*Flutter + On-Device ML*	*Flutter* terbukti mampu mendeteksi objek *real-time* dengan performa mendekati *native*.
21	(Ayu et al., 2021)	*Prototyping Model in Information System Development*	*Prototyping + Information System*	Metode *prototype* memungkinkan pengembangan sistem dilakukan secara iteratif sehingga kebutuhan pengguna dapat diidentifikasi dan disesuaikan secara lebih akurat selama proses pengembangan.
22	(Lorenzoni et al., 2021)	*Machine Learning Model Development from a Software Engineering Perspective*	*Software Engineering+ Machine Learning Development Lifecycle*	Pengembangan *model machine learning* memerlukan pendekatan rekayasa perangkat lunak yang terstruktur agar proses perancangan, implementasi, pengujian, dan pemeliharaan *model* dapat dilakukan secara efektif.
23	(Haristiani & Rifai, 2021)	*Chatbot-based Application Development & Implementation as an Autonomous Language Learning Medium*	*Chatbot + Natural Language Processing (NLP)*	*Chatbot* mampu mendukung pembelajaran mandiri melalui interaksi dua arah secara otomatis, memberikan umpan balik yang cepat, serta meningkatkan keterlibatan pengguna dalam proses pembelajaran.
24	(Khattab & Zaharia, 2020)	*A Comprehensive Review on CNN for Image Classification*	*CNN + Deep Learning*	*Cnn* efektif digunakan untuk klasifikasi citra karena mampu mengenali fitur *visual* secara otomatis dan meningkatkan akurasi pengenalan objek.
25	(Laurence, 2020)	*AI and Machine Learning for On-Device Development*	*On-Device Machine Learning + AI*	*On-device ai* memungkinkan proses deteksi dan klasifikasi berjalan lebih cepat, efisien, serta dapat digunakan tanpa koneksi internet secara terus-menerus.

Sumber: Penulis (2026).

Berdasarkan tinjauan terhadap penelitian terdahulu, dapat diketahui bahwa pengembangan sistem berbasis *computer vision* pada perangkat *mobile* mengalami perkembangan yang cukup pesat dalam beberapa tahun terakhir. Berbagai

penelitian telah berhasil menerapkan teknologi *machine learning*, *object detection*, *chatbot*, serta pemrosesan berbasis perangkat (*on-device processing*) untuk meningkatkan kemampuan identifikasi objek dan penyampaian informasi kepada pengguna. Kondisi ini menjadi dasar bagi pengembangan aplikasi *nutriscan* yang mengombinasikan *flutter*, *google ml kit*, *voice output*, dan *chatbot* dalam satu aplikasi terpadu (Drolia et al., 2022).

Selain aspek teknologi yang digunakan, pendekatan metodologi pengembangan sistem juga menjadi faktor penting dalam keberhasilan implementasi aplikasi *mobile*. Berbagai penelitian menunjukkan bahwa metode pengembangan yang bersifat iteratif memungkinkan pengembang melakukan evaluasi dan penyempurnaan sistem secara berkelanjutan berdasarkan umpan balik pengguna. Pendekatan tersebut sangat relevan diterapkan pada aplikasi berbasis *mobile vision* karena memerlukan pengujian antarmuka, validasi fungsi deteksi, serta evaluasi performa secara langsung pada perangkat yang digunakan. (Riad et al., 2021).

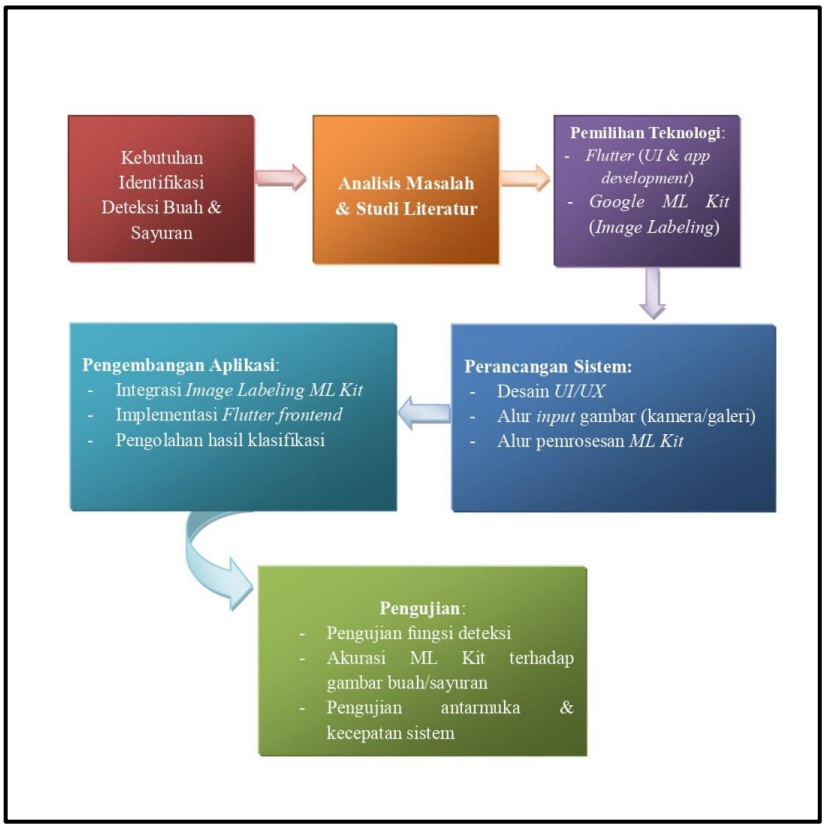
Jika dibandingkan dengan penelitian terdahulu tersebut, penelitian ini memiliki fokus pengembangan yang lebih spesifik. Penelitian kamu memadukan *flutter* sebagai *platform* utama, *google ml kit* sebagai sistem deteksi *on-device*, serta objek penelitian yang dipersempit pada buah dan sayuran. Beda lainnya proyek ini nggak hanya menargetkan akurasi deteksi, tapi juga kecepatan proses, penggunaan ringan di *smartphone*, serta integrasi antarmuka *modern* yang mudah digunakan. Dengan demikian, penelitian ini tidak hanya mengadopsi temuan sebelumnya, tetapi juga mengembangkan sistem yang lebih aplikatif dengan fitur yang relevan untuk penggunaan sehari-hari.

2.11. Kerangka Berpikir Penelitian

Kerangka berpikir penelitian merupakan gambaran konseptual yang digunakan untuk menjelaskan hubungan antara permasalahan, teori, metode, dan tahapan penelitian yang dilakukan. Kerangka berpikir berfungsi sebagai pedoman dalam menyusun alur penelitian secara logis dan sistematis sehingga setiap tahapan yang dilaksanakan memiliki keterkaitan yang jelas dengan tujuan penelitian. Melalui kerangka berpikir, peneliti dapat mengidentifikasi langkah-langkah yang

diperlukan untuk menyelesaikan permasalahan yang diteliti serta memastikan bahwa pendekatan metodologis dan teknologi yang digunakan sesuai dengan kebutuhan penelitian (Nadtayay et al., 2025).

Berdasarkan kerangka berpikir yang disusun pada penelitian ini, proses diawali dengan identifikasi kebutuhan aplikasi deteksi buah dan sayuran, kemudian dilanjutkan dengan analisis masalah dan studi literatur untuk menentukan teknologi yang sesuai. Pemilihan *flutter* sebagai *framework* pengembangan aplikasi serta *google ml kit* sebagai teknologi *image labeling* dilakukan berdasarkan hasil kajian literatur dan kebutuhan sistem. Tahapan berikutnya mencakup perancangan sistem, pengembangan aplikasi, hingga pengujian fungsi, akurasi, dan performa aplikasi untuk memastikan sistem bekerja sesuai dengan tujuan penelitian.



Sumber: Penulis (2026).

(Gambar 2.1. Kerangka Bepikir Penelitian)

Pada Gambar 2.1. menegaskan bahwa penelitian berjalan berurutan, logis, dan berbasis pengembangan iteratif, mulai dari kebutuhan, kajian pustaka,

pemilihan teknologi, perancangan, pembangunan aplikasi hingga evaluasi akhir. Kerangka berpikir pada penelitian ini menjelaskan alur sistematis dalam pengembangan aplikasi deteksi buah dan sayuran menggunakan *flutter* serta *google ml kit*. Penelitian diawali dengan proses identifikasi kebutuhan guna mengetahui urgensi sistem dan masalah yang muncul pada proses klasifikasi buah maupun sayuran secara manual. Dari tahap ini, ditentukan bahwa diperlukan sebuah aplikasi *mobile* yang mampu mendeteksi objek secara otomatis, cepat, dan mudah digunakan.

Tahap selanjutnya adalah analisis masalah dan studi literatur sebagai landasan teori utama. Pada tahap ini dilakukan penelusuran jurnal, *e-book*, penelitian terdahulu, serta dokumentasi teknis berkaitan dengan *flutter*, *image labeling ml kit, machine learning*, serta penerapan deteksi objek pada perangkat *mobile*. Studi literatur ini berfungsi memperkuat konsep serta memastikan bahwa teknologi yang dipilih relevan dengan tujuan penelitian.

Berdasarkan hasil analisis dan kajian pustaka, penelitian kemudian mengarah pada pemilihan teknologi yang akan digunakan. *Flutter* dipilih sebagai *framework* karena bersifat *cross-platform*, ringan, dan mampu menghasilkan performa *ui* yang responsif. *Google ml kit* dipilih sebagai model pendeteksi gambar secara *on-device* melalui fitur *image labeling*. Kombinasi keduanya memungkinkan proses klasifikasi berlangsung cepat tanpa bergantung pada *server* eksternal, sekaligus menjaga efisiensi *resource* perangkat.

Tahap berikutnya adalah perancangan sistem meliputi desain antarmuka *(UI/UX)*, alur input gambar melalui kamera atau galeri, serta perancangan alur pemrosesan data oleh *ml kit*. Setelah perancangan selesai, penelitian berlanjut pada tahap pengembangan aplikasi dengan melakukan integrasi label deteksi, implementasi *frontend flutter*, serta pengolahan hasil klasifikasi agar dapat ditampilkan kepada pengguna.

Tahap terakhir yaitu pengujian untuk memastikan sistem berfungsi sesuai tujuan. Pengujian dilakukan pada fitur kamera/galeri, akurasi *model ml kit* dalam mengenali buah dan sayuran, serta kecepatan deteksi pada perangkat *smartphone*. Selain itu, diuji juga kenyamanan antarmuka dan stabilitas sistem agar aplikasi layak digunakan sebagai media identifikasi objek secara cepat.

BAB III
METODOLOGI PENELITIAN

3.1. Objek Penelitian, Subjek Penelitian dan Jadwal Penelitian

3.1.1. Objek Penelitian

Objek penelitian dalam penelitian ini adalah aplikasi *nutriscan*, yaitu aplikasi deteksi buah dan sayuran berbasis *flutter* yang memanfaatkan teknologi *google ml kit* (*image labeling*) sebagai sistem pengenalan objek secara *visual*. Aplikasi ini dirancang untuk mengidentifikasi berbagai jenis buah dan sayuran melalui kamera maupun unggahan gambar, kemudian menampilkan hasil klasifikasi secara *real-time* pada perangkat *mobile*. Pengembangan aplikasi dilakukan untuk menghasilkan sistem yang mampu melakukan proses identifikasi objek secara cepat, stabil, dan sesuai dengan kebutuhan pengguna (Bonesso et al., 2025).

Fitur yang menjadi objek penelitian meliputi proses pengambilan gambar, pemrosesan citra, klasifikasi objek menggunakan *google ml kit*, penyajian hasil deteksi, *confidence score*, serta *voice output* yang menyampaikan hasil identifikasi kepada pengguna. Selain itu, penelitian juga berfokus pada integrasi antarmuka pengguna (*user interface*) yang mendukung kemudahan penggunaan aplikasi sehingga proses identifikasi objek dapat dilakukan secara efektif pada perangkat *mobile* (Muneeb et al., 2022).

Ruang lingkup objek penelitian dibatasi pada pengembangan aplikasi yang berfungsi untuk mendeteksi dan mengklasifikasikan buah serta sayuran menggunakan teknologi *google ml kit* melalui fitur *image labeling*. Selain itu, penelitian difokuskan pada implementasi aplikasi berbasis *android* menggunakan *framework flutter* dengan penambahan fitur *voice output* sebagai media penyampaian hasil deteksi, tanpa melakukan analisis lanjutan berbasis *natural language processing* (*NLP*) (Y. Huang et al., 2021).

Secara keseluruhan, objek penelitian berfokus pada pengembangan aplikasi *nutriscan* sebagai media identifikasi buah dan sayuran pada perangkat *mobile*. Aplikasi tersebut mengintegrasikan *flutter* dan *google ml kit* untuk menghasilkan proses deteksi yang praktis, responsif, dan mudah digunakan. Ruang lingkup yang

telah ditetapkan menjadi dasar dalam pelaksanaan pengembangan serta pengujian fungsi aplikasi agar hasil yang diperoleh tetap sesuai dengan tujuan penelitian.

3.1.2. Subjek Penelitian

Subjek penelitian dalam penelitian ini adalah pengguna yang terlibat secara langsung dalam proses uji coba aplikasi *nutriscan*. Pengguna berperan sebagai responden yang mengoperasikan aplikasi melalui perangkat *android* dan memberikan data berdasarkan pengalaman penggunaan aplikasi. Keterlibatan pengguna diperlukan untuk mengetahui kemudahan pengoperasian, kejelasan informasi, serta tingkat kepuasan terhadap fungsi-fungsi yang tersedia dalam aplikasi. Pengujian aplikasi *mobile* yang melibatkan pengguna dapat menghasilkan informasi mengenai efektivitas, efisiensi, dan kepuasan pengguna selama menyelesaikan tugas tertentu pada aplikasi (Weichbroth, 2024).

Subjek penelitian dipilih dengan mempertimbangkan kemampuan dasar dalam menggunakan perangkat *smartphone* berbasis *android* serta kesediaan untuk mengikuti seluruh tahapan pengujian aplikasi. Setiap pengguna diminta menjalankan beberapa aktivitas, seperti membuka aplikasi, mengambil gambar melalui kamera, memilih gambar dari galeri, melakukan deteksi buah atau sayuran, serta melihat hasil identifikasi yang ditampilkan. Pelaksanaan pengujian berbasis tugas memungkinkan peneliti mengamati interaksi pengguna dan mengetahui keberhasilan pengguna dalam menjalankan fungsi-fungsi yang tersedia pada aplikasi (Shahil-Feroz et al., 2023).

Setelah menggunakan aplikasi, responden memberikan penilaian melalui kuesioner berdasarkan indikator yang telah ditetapkan. Penilaian tersebut mencakup kecepatan tampilnya hasil deteksi, kejelasan informasi klasifikasi, kemudahan penggunaan kamera dan galeri, fungsi *voice output*, tampilan antarmuka, serta kenyamanan penggunaan aplikasi secara keseluruhan. Pelaksanaan pengujian berbasis tugas memungkinkan aktivitas pengguna, waktu penyelesaian, kesalahan, dan hasil penilaian terhadap antarmuka aplikasi dicatat sebagai data penelitian (C. J. Huang & Kolopitawondal, 2023).

Secara keseluruhan, subjek penelitian merupakan pengguna aplikasi *nutriscan* yang menjadi sumber data primer dalam pengujian sistem. Keterlibatan

pengguna diharapkan dapat memberikan gambaran mengenai cara kerja aplikasi ketika digunakan secara langsung serta membantu memperoleh data yang relevan mengenai fungsi, kemudahan, dan tingkat kepuasan penggunaan aplikasi. Jumlah responden dan karakteristiknya akan ditetapkan sesuai dengan kebutuhan pengujian serta ketentuan penelitian yang digunakan.

3.1.3. Jadwal Penelitian

Jadwal penelitian disusun berdasarkan alur kerja mulai dari persiapan, pengembangan, hingga evaluasi. Tabel berikut merangkum urutannya:

Tabel 3.1 Jadwal Penelitian

No	Kegiatan	Bulan Ke-1	Bulan Ke-2	Bulan Ke-3	Bulan Ke-4	Bulan Ke-5	Bulan Ke-6
1	Studi literatur & Pengumpulan data teori	√	√	-	-	-	-
2	Analisis kebutuhan sistem	√	√	-	-	-	-
3	Perancangan *UI/UX* aplikasi	-	√	√	-	-	-
4	Pengembangan sistem (*Flutter + ML Kit*)	-	-	√	√	√	-
5	Pengujian Aplikasi	-	-	-	√	√	√
6	Penyusunan laporan penelitian	-	-	-	-	√	√

Sumber: Penulis, 2026

3.2. Metode Penelitian

Penelitian ini menggunakan pendekatan kuantitatif untuk memperoleh data yang dapat diukur dan dianalisis secara sistematis. Dalam penelitian kuantitatif, variabel merupakan atribut atau karakteristik yang memiliki nilai berbeda dan digunakan untuk menggambarkan objek yang diteliti. Variabel dalam penelitian ini disusun berdasarkan tujuan pengembangan aplikasi *nutriscan* yang memanfaatkan *flutter* dan *google ml kit* untuk mendeteksi buah serta sayuran. Variabel tersebut mencakup kinerja aplikasi, fungsionalitas sistem, dan kemudahan penggunaan aplikasi oleh pengguna (Andrade, 2021).

Setiap variabel dijabarkan ke dalam definisi operasional, dimensi, indikator, dan butir pernyataan agar proses pengumpulan data dapat dilakukan secara terarah. Definisi operasional menjelaskan batasan variabel secara spesifik sesuai dengan konteks penelitian, sedangkan indikator digunakan sebagai dasar untuk menyusun

instrumen pengukuran. Penentuan variabel dan definisi operasional juga membantu memastikan bahwa data yang dikumpulkan memiliki hubungan yang sesuai dengan rumusan masalah serta tujuan penelitian (Barroga & Matanguihan, 2022).

Berdasarkan penjabaran tersebut, variabel penelitian digunakan sebagai dasar dalam penyusunan kuesioner dan pengukuran tingkat kepuasan pengguna terhadap aplikasi *nutriscan*. Setiap indikator kemudian diterjemahkan menjadi butir pernyataan yang dinilai menggunakan skala penilaian dari sangat memuaskan sampai tidak memuaskan. Uraian mengenai variabel, definisi operasional, dimensi, indikator, dan skala pengukuran disajikan pada Tabel 3.2.

Tabel 3.2 Operasional Variabel Penelitian

No	Variabel	Definisi Operasional	Dimensi	Indikator Penilaian	Skala Pengukuran
1	Persepsi Pengguna terhadap Kinerja Aplikasi *NutriScan*	Penilaian pengguna terhadap kemampuan aplikasi dalam menampilkan hasil deteksi buah dan sayuran secara tepat, cepat, lancar, dan mudah dipahami.	Ketepatan deteksi buah	Aplikasi mampu mengenali buah dengan benar.	Sangat Memuaskan, Memuaskan, Cukup Memuaskan, Kurang Memuaskan, Tidak Memuaskan
2			Ketepatan deteksi sayuran	Aplikasi mampu mengenali sayuran dengan benar.	Sangat Memuaskan, Memuaskan, Cukup Memuaskan, Kurang Memuaskan, Tidak Memuaskan
3			Kecepatan hasil deteksi	Hasil deteksi ditampilkan dengan cepat.	Sangat Memuaskan, Memuaskan, Cukup Memuaskan, Kurang Memuaskan, Tidak Memuaskan
4			Kelancaran pemindaian	Proses pemindaian berjalan tanpa jeda yang mengganggu.	Sangat Memuaskan, Memuaskan, Cukup Memuaskan, Kurang Memuaskan, Tidak Memuaskan
5			Keberhasilan identifikasi	Objek yang diuji dapat dikenali sesuai kategori yang tersedia.	Sangat Memuaskan, Memuaskan, Cukup Memuaskan, Kurang Memuaskan, Tidak Memuaskan
6			Kejelasan *confidence score*	Tingkat keyakinan hasil deteksi dianggap	Sangat Memuaskan, Memuaskan, Cukup Memuaskan, Kurang

				sesuai dengan objek yang dikenali.	Memuaskan, Tidak Memuaskan
7	Kepuasan Pengguna terhadap Fungsionalitas dan *Usability* Aplikasi	Penilaian pengguna terhadap fungsi fitur, kemudahan pengoperasian, kejelasan tampilan, dan kenyamanan penggunaan aplikasi NutriScan.	Fungsi kamera dan galeri	Fitur kamera dan galeri berfungsi dengan baik.	Sangat Memuaskan, Memuaskan, Cukup Memuaskan, Kurang Memuaskan, Tidak Memuaskan
8			Fungsi *voice output*	Fitur *voice output* berjalan dengan baik.	Sangat Memuaskan, Memuaskan, Cukup Memuaskan, Kurang Memuaskan, Tidak Memuaskan
9			Antarmuka pengguna	Tampilan aplikasi mudah dipahami dan digunakan.	Sangat Memuaskan, Memuaskan, Cukup Memuaskan, Kurang Memuaskan, Tidak Memuaskan
10			Pengalaman pengguna	Aplikasi memberikan pengalaman penggunaan yang nyaman secara keseluruhan.	Sangat Memuaskan, Memuaskan, Cukup Memuaskan, Kurang Memuaskan, Tidak Memuaskan

Sumber: Penulis, 2026

Berdasarkan Tabel 3.2, variabel penelitian terdiri atas persepsi pengguna terhadap kinerja aplikasi *nutriscan* serta kepuasan pengguna terhadap fungsionalitas dan *usability* aplikasi. Persepsi terhadap kinerja aplikasi diukur berdasarkan ketepatan hasil deteksi buah dan sayuran, kecepatan penyajian hasil, kelancaran proses pemindaian, keberhasilan identifikasi objek, serta kejelasan informasi *confidence score* yang ditampilkan oleh aplikasi.

Kepuasan pengguna terhadap fungsionalitas dan *usability* aplikasi diukur berdasarkan fungsi kamera dan galeri, fungsi *voice output*, kemudahan memahami antarmuka pengguna, serta kenyamanan penggunaan aplikasi secara keseluruhan. Data diperoleh melalui kuesioner yang diberikan kepada responden setelah menggunakan fitur-fitur utama aplikasi *nutriscan*. Setiap indikator dinilai menggunakan skala lima tingkat, yaitu sangat memuaskan, memuaskan, cukup memuaskan, kurang memuaskan, dan tidak memuaskan.

Secara keseluruhan, operasionalisasi variabel tersebut menjadi dasar dalam penyusunan instrumen kuesioner dan pengumpulan data mengenai penilaian pengguna terhadap aplikasi *nutriscan*. Adapun akurasi deteksi secara objektif, waktu respons, nilai *confidence score*, dan keberhasilan fungsi sistem tetap diukur secara terpisah melalui pengujian teknis aplikasi.

3.3. Sumber Data

Sumber data yang digunakan dalam penelitian ini terdiri atas data primer dan data sekunder. Data primer diperoleh secara langsung melalui pengujian aplikasi dan pengisian kuesioner oleh pengguna, sedangkan data sekunder diperoleh melalui kajian terhadap berbagai sumber literatur ilmiah dan dokumentasi teknis. Kedua sumber data tersebut digunakan untuk mendukung proses pengembangan, pengujian, dan analisis aplikasi *nutriscan* berbasis *flutter* dan *google ml kit*.

3.3.1. Data Primer

Data primer dalam penelitian ini diperoleh secara langsung melalui pengujian aplikasi *nutriscan* pada perangkat *smartphone* berbasis *android* dan penyebaran kuesioner kepada pengguna yang terlibat sebagai responden. Pengujian aplikasi dilakukan menggunakan sejumlah gambar buah dan sayuran yang kategorinya telah diketahui. Dari proses pengujian tersebut diperoleh data kuantitatif berupa jumlah hasil identifikasi yang benar dan salah, tingkat keberhasilan deteksi, nilai *confidence score*, serta waktu respons aplikasi dalam menampilkan hasil identifikasi. Data primer dapat dikumpulkan melalui eksperimen, survei, wawancara, dan kuesioner sesuai dengan kebutuhan serta tujuan penelitian (Taherdoost, 2021).

Selain data hasil pengujian teknis, data primer juga diperoleh melalui kuesioner yang diisi oleh responden setelah menggunakan aplikasi *nutriscan*. Penilaian responden meliputi ketepatan hasil deteksi menurut pengguna, kecepatan penyajian hasil, kelancaran proses pemindaian, kejelasan *confidence score*, fungsi kamera dan galeri, fungsi *voice output*, kemudahan memahami antarmuka, serta kenyamanan penggunaan aplikasi. Jawaban responden diukur menggunakan lima

kategori penilaian, yaitu sangat memuaskan, memuaskan, cukup memuaskan, kurang memuaskan, dan tidak memuaskan (Milosz et al., 2024).

Data hasil pengujian teknis dan data kuesioner dianalisis secara terpisah karena keduanya memiliki tujuan pengukuran yang berbeda. Data teknis digunakan untuk mengetahui kinerja aplikasi berdasarkan hasil pengujian secara langsung, sedangkan data kuesioner digunakan untuk mengetahui persepsi dan tingkat kepuasan pengguna terhadap fungsionalitas serta kemudahan penggunaan aplikasi *nutriscan*.

3.3.2. Data Sekunder

Data sekunder dalam penelitian ini diperoleh dari berbagai sumber ilmiah dan dokumentasi teknis yang relevan dengan pengembangan aplikasi *nutriscan*. Sumber data tersebut meliputi jurnal ilmiah, buku elektronik, *prosiding*, repositori penelitian, penelitian terdahulu, serta dokumentasi resmi yang membahas *flutter*, *google ml kit*, *machine learning*, pengolahan citra digital, pengembangan aplikasi *mobile*, dan metode pengembangan sistem *prototype*. Tinjauan literatur dalam bidang ilmu komputer dilakukan dengan mengumpulkan, mengidentifikasi, dan menganalisis penelitian yang tersedia secara sistematis agar dapat memberikan gambaran mengenai perkembangan serta kesenjangan penelitian pada suatu topik (Carrera-Rivera et al., 2022)

Data sekunder digunakan sebagai landasan dalam menentukan konsep, teknologi, metode pengembangan, dan teknik pengujian yang diterapkan pada aplikasi *nutriscan*. Pemilihan sumber dilakukan dengan memperhatikan kesesuaian isi dengan topik penelitian, tahun penerbitan, identitas penulis, kredibilitas penerbit, ketersediaan dokumen lengkap, serta kejelasan metode dan hasil penelitian. Proses penilaian kualitas sumber diperlukan karena mutu kajian ilmiah dapat berbeda, sehingga sumber yang digunakan harus diperiksa secara terstruktur agar mampu mendukung dasar teori dan keputusan teknis dalam penelitian perangkat lunak (Usman et al., 2023).

Berdasarkan uraian tersebut, data sekunder berperan sebagai dasar teoritis dan referensi teknis yang mendukung proses perancangan, pengembangan, serta pengujian aplikasi *nutriscan*. Penggunaan sumber yang relevan dan kredibel

membantu memastikan bahwa setiap tahapan penelitian dilaksanakan berdasarkan konsep ilmiah dan perkembangan teknologi yang sesuai. Data sekunder tersebut selanjutnya digunakan untuk melengkapi data primer yang diperoleh melalui pengujian aplikasi dan penilaian pengguna.

3.4. Teknik Pengumpulan Data

Teknik pengumpulan data dalam penelitian ini dilakukan melalui observasi terstruktur, pengujian sistem, kuesioner, dan dokumentasi. Teknik tersebut digunakan untuk memperoleh data kuantitatif yang mendukung proses pengembangan serta pengujian aplikasi *nutriscan* berbasis *flutter* dan *google ml kit*. Data yang dikumpulkan meliputi hasil pengamatan objek uji, hasil pengujian teknis aplikasi, penilaian pengguna, dan berbagai dokumen pendukung penelitian. Pemilihan teknik pengumpulan data perlu disesuaikan dengan tujuan, jenis data, ruang lingkup, dan pendekatan penelitian yang digunakan (Mwita, 2022).

3.4.1. Observasi Terstruktur

Observasi dilakukan secara terstruktur dengan mengamati objek buah dan sayuran yang digunakan sebagai gambar uji serta proses aplikasi dalam melakukan identifikasi objek. Pengamatan dilakukan menggunakan lembar observasi yang memuat jenis objek, kondisi pencahayaan, jarak pengambilan gambar, sumber gambar dari kamera atau galeri, hasil klasifikasi, nilai *confidence score*, dan keberhasilan aplikasi dalam menampilkan hasil deteksi. Observasi kuantitatif dilakukan secara sistematis menggunakan parameter dan lembar pencatatan yang telah ditentukan agar hasil pengamatan dapat dicatat dalam bentuk angka, tanda centang, atau skor (Ellis, 2024).

Hasil observasi selanjutnya dicatat secara sistematis ke dalam lembar pengamatan untuk memudahkan proses pengelompokan dan analisis data. Data yang diperoleh digunakan untuk membandingkan hasil deteksi pada setiap objek uji berdasarkan kondisi pengambilan gambar yang berbeda, seperti pencahayaan, jarak, sudut pengambilan, dan sumber gambar. Melalui pencatatan tersebut, peneliti dapat mengetahui kondisi yang mendukung maupun menghambat proses identifikasi buah dan sayuran pada aplikasi *nutriscan*.

3.4.2. Pengujian Sistem

Pengujian sistem dilakukan untuk mengetahui kesesuaian fungsi aplikasi *nutriscan* dengan rancangan dan kebutuhan sistem yang telah ditetapkan. Pengujian mencakup fungsi kamera, pemilihan gambar dari galeri, proses *image labeling*, penyajian hasil klasifikasi, penampilan *confidence score*, serta fungsi *voice output*. Setiap fungsi diuji menggunakan skenario yang memuat masukan, hasil yang diharapkan, hasil yang diperoleh, dan status berhasil atau tidak berhasil. Pengujian *black box* dilakukan dengan mengamati respons dan keluaran aplikasi berdasarkan masukan yang diberikan tanpa memeriksa struktur kode program (Saman & Indah Ratnasari, 2022).

Pengujian deteksi dilakukan menggunakan sejumlah gambar buah dan sayuran yang kategorinya telah diketahui sebelumnya. Hasil pengujian dicatat untuk memperoleh jumlah identifikasi benar dan salah, tingkat keberhasilan deteksi, waktu respons aplikasi, dan nilai *confidence score* yang ditampilkan. Data tersebut digunakan untuk mengukur kinerja teknis aplikasi secara objektif. Nilai *confidence score* dicatat sebagai tingkat keyakinan sistem terhadap hasil klasifikasi dan tidak langsung dianggap sebagai bukti bahwa hasil deteksi sudah benar.

3.4.3. Kuesioner

Kuesioner diberikan kepada responden setelah menggunakan fitur-fitur utama aplikasi *nutriscan*. Kuesioner berisi pernyataan tertutup yang disusun berdasarkan indikator pada operasional variabel penelitian, meliputi ketepatan deteksi menurut pengguna, kecepatan penyajian hasil, kelancaran pemindaian, kejelasan *confidence score*, fungsi kamera dan galeri, fungsi *voice output*, tampilan antarmuka, dan kenyamanan penggunaan aplikasi. Kuesioner tertutup memberikan pilihan jawaban yang telah ditentukan sehingga data responden dapat diubah menjadi skor dan dianalisis secara kuantitatif (Rathi & Ronald, 2022).

Jawaban responden menggunakan lima kategori penilaian, yaitu sangat memuaskan dengan skor 5, memuaskan dengan skor 4, cukup memuaskan dengan skor 3, kurang memuaskan dengan skor 2, dan tidak memuaskan dengan skor 1. Hasil kuesioner digunakan untuk mengetahui tingkat kepuasan pengguna terhadap kinerja, fungsionalitas, antarmuka, dan pengalaman penggunaan aplikasi *nutriscan*.

3.4.4. Dokumentasi

Dokumentasi dilakukan dengan mengumpulkan dan menyimpan berbagai bukti yang berkaitan dengan pelaksanaan penelitian. Dokumentasi tersebut meliputi gambar buah dan sayuran yang digunakan sebagai objek uji, lembar hasil observasi, tabel hasil pengujian sistem, jawaban kuesioner, tangkapan layar aplikasi, serta catatan selama proses pengembangan. Dokumen dapat digunakan sebagai sumber data pendukung apabila dipilih, disusun, dan diperiksa sesuai dengan kebutuhan penelitian (Ramazan, 2021).

Secara keseluruhan, penggunaan observasi terstruktur, pengujian sistem, kuesioner, dan dokumentasi memungkinkan penelitian memperoleh data yang sesuai dengan pendekatan kuantitatif. Observasi terstruktur dan pengujian sistem digunakan untuk memperoleh data teknis mengenai kinerja aplikasi, sedangkan kuesioner digunakan untuk memperoleh data penilaian pengguna. Dokumentasi berfungsi sebagai bukti pendukung agar proses pengumpulan dan analisis data dapat dilakukan secara terukur, sistematis, dan dapat dipertanggungjawabkan.

3.5. Metode Pengembangan Sistem

Metode pengembangan sistem yang digunakan dalam penelitian ini adalah metode *prototype*, yaitu pendekatan pengembangan perangkat lunak yang dilakukan dengan membangun model awal sistem untuk diuji dan diperbaiki secara berulang hingga sesuai dengan kebutuhan pengguna. Metode ini dipilih karena memungkinkan rancangan aplikasi ditampilkan sejak tahap awal, sehingga kekurangan fungsi, antarmuka, dan alur penggunaan dapat diketahui sebelum sistem dikembangkan menjadi bentuk akhir. Pendekatan *prototype* bersifat iteratif karena rancangan dikembangkan, diuji, dan disempurnakan berdasarkan hasil evaluasi yang diperoleh selama proses pengembangan (Syarif & Risdiansyah, 2024).

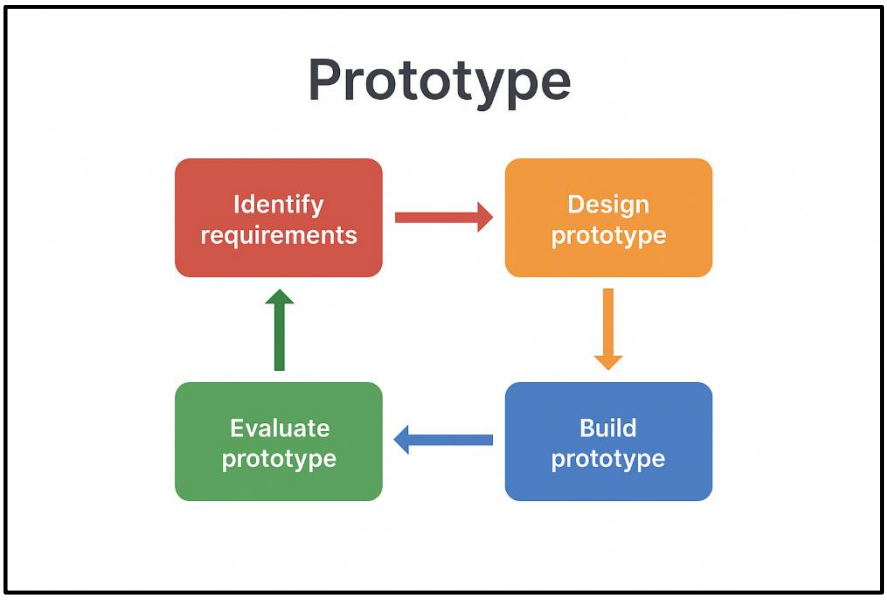
Tahap pertama adalah identifikasi kebutuhan (*identify requirements*) dan perancangan *prototype* (*design prototype*). Identifikasi kebutuhan dilakukan dengan menentukan tujuan pengembangan, karakteristik objek buah dan sayuran, kebutuhan pengguna, fungsi utama aplikasi, serta perangkat lunak dan perangkat keras yang digunakan. Berdasarkan hasil identifikasi tersebut, selanjutnya dibuat

rancangan awal aplikasi yang meliputi desain antarmuka pengguna, struktur halaman, navigasi, tombol kamera dan galeri, halaman hasil identifikasi, tampilan *confidence score*, serta fitur *voice output*. Tahap identifikasi kebutuhan dan rancangan cepat digunakan untuk memperoleh gambaran awal mengenai fungsi serta tampilan sistem yang akan dikembangkan (Kurniawan et al., 2023).

Tahap berikutnya adalah pembangunan *prototype* (*build prototype*). Pada tahap ini, rancangan awal diterapkan ke dalam aplikasi berbasis *android* menggunakan *framework flutter*. Proses pembangunan mencakup pembuatan antarmuka, penerapan fitur pengambilan gambar melalui kamera dan galeri, integrasi *google ml kit* melalui fitur *image labeling*, pengolahan hasil klasifikasi, penampilan *confidence score*, serta penerapan *voice output*. Pembangunan dilakukan secara bertahap dengan berfokus pada penerapan fungsi-fungsi utama hingga menghasilkan versi awal aplikasi *nutriscan* yang dapat menjalankan proses identifikasi buah dan sayuran serta digunakan dalam tahap pengujian (Fahmi, 2024).

Tahap terakhir adalah evaluasi *prototype* (*evaluate prototype*). Evaluasi dilakukan melalui pengujian fungsi aplikasi, pengujian hasil identifikasi terhadap kumpulan gambar buah dan sayuran, pengukuran waktu respons, pencatatan *confidence score*, serta penilaian pengguna terhadap antarmuka dan kemudahan penggunaan aplikasi. Apabila hasil evaluasi menunjukkan adanya fungsi yang belum berjalan sesuai rancangan atau hasil identifikasi yang belum memenuhi kebutuhan penelitian, maka *prototype* diperbaiki dan diuji kembali. Proses evaluasi dan penyempurnaan dilakukan secara berulang untuk mengurangi kekurangan serta meningkatkan kesesuaian aplikasi dengan kebutuhan pengguna (Nabawi et al., 2024).

Secara keseluruhan, penerapan metode *prototype* memberikan alur pengembangan yang terstruktur sekaligus fleksibel dalam membangun aplikasi *nutriscan*. Setiap tahap saling berkaitan dan dapat diulang apabila ditemukan kekurangan pada hasil pengembangan. Melalui proses tersebut, aplikasi diharapkan mampu menjalankan fungsi identifikasi buah dan sayuran secara responsif, stabil, mudah digunakan, dan sesuai dengan tujuan penelitian.



Sumber: Penulis, 2026

Gambar 3.1 Model Konseptual *Prototype*

Pada Gambar 3.1 ditunjukkan bahwa proses pengembangan sistem dimulai dari tahap *identify requirements* atau identifikasi kebutuhan. Tahap ini dilakukan untuk menentukan kebutuhan pengguna, fitur aplikasi, objek yang akan dideteksi, serta teknologi yang digunakan. Kebutuhan yang telah diperoleh menjadi dasar pada tahap *design prototype*, yaitu penyusunan rancangan awal antarmuka dan alur kerja aplikasi sebelum diterapkan ke dalam sistem.

Rancangan tersebut selanjutnya diterapkan pada tahap *build prototype* untuk menghasilkan versi awal aplikasi *nutriscan* yang dapat menjalankan fungsi utama. Versi awal kemudian memasuki tahap *evaluate prototype* melalui pengujian fungsi, hasil identifikasi, kecepatan sistem, dan penilaian pengguna. Apabila masih ditemukan kekurangan, proses dikembalikan ke tahap identifikasi kebutuhan untuk dilakukan perbaikan. Siklus tersebut dilaksanakan hingga prototype memenuhi kebutuhan dan dapat digunakan sebagai aplikasi akhir.

3.6. Alat Bantu Perancangan

Alat bantu perancangan digunakan untuk memvisualisasikan alur kerja sistem, struktur komponen, serta interaksi antar proses agar pengembangan menjadi lebih terarah dan mudah dipahami. Pada penelitian ini, alat bantu yang digunakan meliputi:

3.6.1. *Flowchart*

Dipakai untuk menggambarkan alur logika dari proses kerja aplikasi secara sistematis. *Flowchart* membantu peneliti memahami urutan langkah mulai dari input, proses, hingga *output*.

Tabel 3.3 *Flowchart* Proses Aplikasi Deteksi Buah dan Sayuran

No	Simbol Flowchart	Nama Simbol	Keterangan
1	⬭	*Terminator*	Menunjukkan awal dan akhir proses aplikasi
2	▰	*Input/Output*	Menunjukkan proses masukan gambar melalui kamera atau galeri serta keluaran berupa hasil deteksi.
3	▭	*Process*	Menunjukkan proses pengolahan gambar dan klasifikasi objek menggunakan *google ml kit*.
4	◆	*Decision*	Menunjukkan pengambilan keputusan berdasarkan kondisi tertentu, seperti menentukan apakah objek berhasil dikenali sebagai buah atau sayuran.
5	⟶	*Flowline*	Menunjukkan arah dan urutan aliran proses dari satu tahapan menuju tahapan berikutnya.

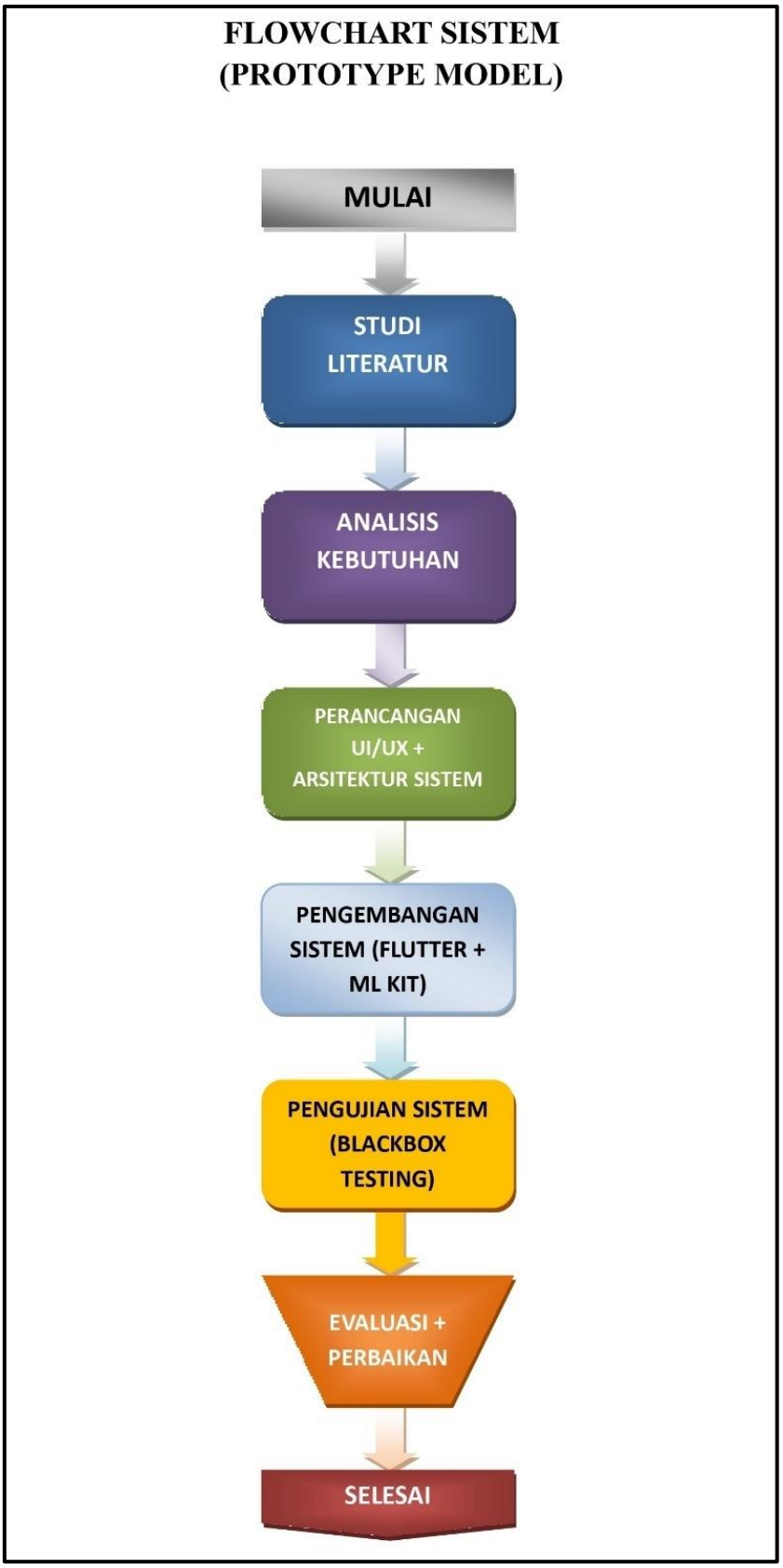
Sumber: (Pebriani et al., 2023)

Berdasarkan Tabel 3.3, simbol-simbol *flowchart* digunakan untuk menggambarkan alur kerja aplikasi *nutriscan* secara sistematis, mulai dari proses masukan gambar hingga penyajian hasil identifikasi. Simbol *terminator* menunjukkan awal dan akhir sistem, simbol *input/output* menggambarkan proses pengambilan atau pemilihan gambar serta penyajian hasil, simbol *process* menunjukkan pemrosesan gambar, simbol *decision* digunakan untuk menentukan hasil berdasarkan kondisi tertentu, sedangkan *flowline* menunjukkan hubungan dan

arah perpindahan antartahapan. Penggunaan simbol yang terstruktur membantu menyajikan alur sistem secara jelas dan ringkas sehingga proses yang kompleks lebih mudah dipahami serta dianalisis (Windarti et al., 2025).

Flowchart juga berfungsi sebagai alat bantu perancangan dan dokumentasi sistem sebelum proses implementasi dilakukan. Alur yang telah dirancang memberikan gambaran mengenai urutan aktivitas, proses masukan, pengolahan data, percabangan keputusan, dan keluaran aplikasi. Penggunaan simbol *terminator*, *process*, *decision*, dan *input*/*output* pada perancangan sistem membantu pengembang menghubungkan setiap tahapan secara logis sehingga implementasi pada *framework flutter* dapat dilakukan secara lebih terarah (Imran et al., 2024).

Secara keseluruhan, penggunaan *flowchart* dalam penelitian ini memberikan dasar yang jelas dalam proses perancangan dan pengembangan aplikasi *nutriscan*. Alur proses yang terdefinisi membantu peneliti memahami hubungan antarfungsi, menemukan kemungkinan kesalahan logika, serta menentukan skenario pengujian sistem. Dengan demikian, aplikasi dapat dikembangkan secara lebih terstruktur, mudah dipahami, dan sesuai dengan kebutuhan penelitian.



Sumber: Penulis, 2026

Gambar 3.2 *Flowchart* Sistem

Gambar 3.2 menunjukkan alur pengembangan aplikasi *nutriscan* menggunakan *model prototype*. Proses dimulai dengan studi literatur untuk mempelajari teori, teknologi, dan penelitian terdahulu yang berkaitan dengan *flutter*, *google ml kit*, pengolahan citra, serta aplikasi deteksi buah dan sayuran. Hasil studi literatur selanjutnya digunakan pada tahap analisis kebutuhan untuk menentukan kebutuhan pengguna, fungsi utama aplikasi, perangkat yang digunakan, dan teknologi yang akan diterapkan dalam proses pengembangan sistem.

Setelah kebutuhan sistem ditentukan, proses dilanjutkan dengan perancangan *ui*/*ux* dan arsitektur sistem. Tahap ini mencakup penyusunan struktur halaman, navigasi aplikasi, fitur kamera dan galeri, halaman hasil deteksi, tampilan *confidence score*, fitur *voice output*, serta alur pemrosesan gambar menggunakan *google ml kit*. Rancangan tersebut kemudian diterapkan pada tahap pengembangan sistem menggunakan *framework flutter* dan teknologi *google ml kit* melalui fitur *image labeling*.

Aplikasi yang telah dikembangkan selanjutnya diuji menggunakan metode *black-box testing* untuk memastikan setiap fitur dapat berjalan sesuai dengan fungsi yang telah ditentukan. Pengujian dilakukan terhadap fungsi kamera, galeri, proses identifikasi gambar, penyajian hasil klasifikasi, *confidence score*, dan *voice output*. Apabila ditemukan fungsi yang belum berjalan sesuai kebutuhan, sistem memasuki tahap evaluasi dan perbaikan. Proses pengembangan, pengujian, serta perbaikan dilakukan secara berulang hingga aplikasi memenuhi kebutuhan penelitian dan dapat dinyatakan selesai.

3.6.2. Alat Bantu Sistem

Alat bantu sistem merupakan sejumlah perangkat dan teknologi yang digunakan untuk mendukung proses pemodelan, perancangan, implementasi, serta pengujian aplikasi *nutriscan*. Penggunaan alat bantu tersebut bertujuan agar setiap tahapan pengembangan dapat dilaksanakan secara terstruktur, mulai dari menggambarkan kebutuhan dan alur kerja sistem hingga mengelola konfigurasi serta data aplikasi. Pemodelan sistem juga membantu pengembang menyajikan

kebutuhan dan interaksi pengguna dalam bentuk *visual* sebelum sistem diterapkan ke dalam kode program (Ahmad et al., 2023).

1. Desain Sistem dan Arsitektur Sistem

Perancangan sistem dilakukan menggunakan *flowchart*, *use case diagram*, dan diagram arsitektur sistem. *Flowchart* digunakan untuk menggambarkan urutan proses aplikasi, sedangkan *use case diagram* digunakan untuk memperlihatkan hubungan antara pengguna dan fungsi yang tersedia dalam aplikasi *nutriscan*. Diagram arsitektur sistem digunakan untuk menjelaskan hubungan antarkomponen, seperti antarmuka aplikasi *flutter*, masukan gambar dari kamera atau galeri, proses *image labeling* pada *google ml kit*, serta penyajian hasil identifikasi. Pemodelan menggunakan notasi yang sesuai membantu menjelaskan kebutuhan dan integrasi fungsi sistem sebelum memasuki tahap implementasi (Rospricilia & Ma'ady, 2024).

2. Basis Data Menggunakan Cloud Firestore dari *Firebase CLI* (*Database)*

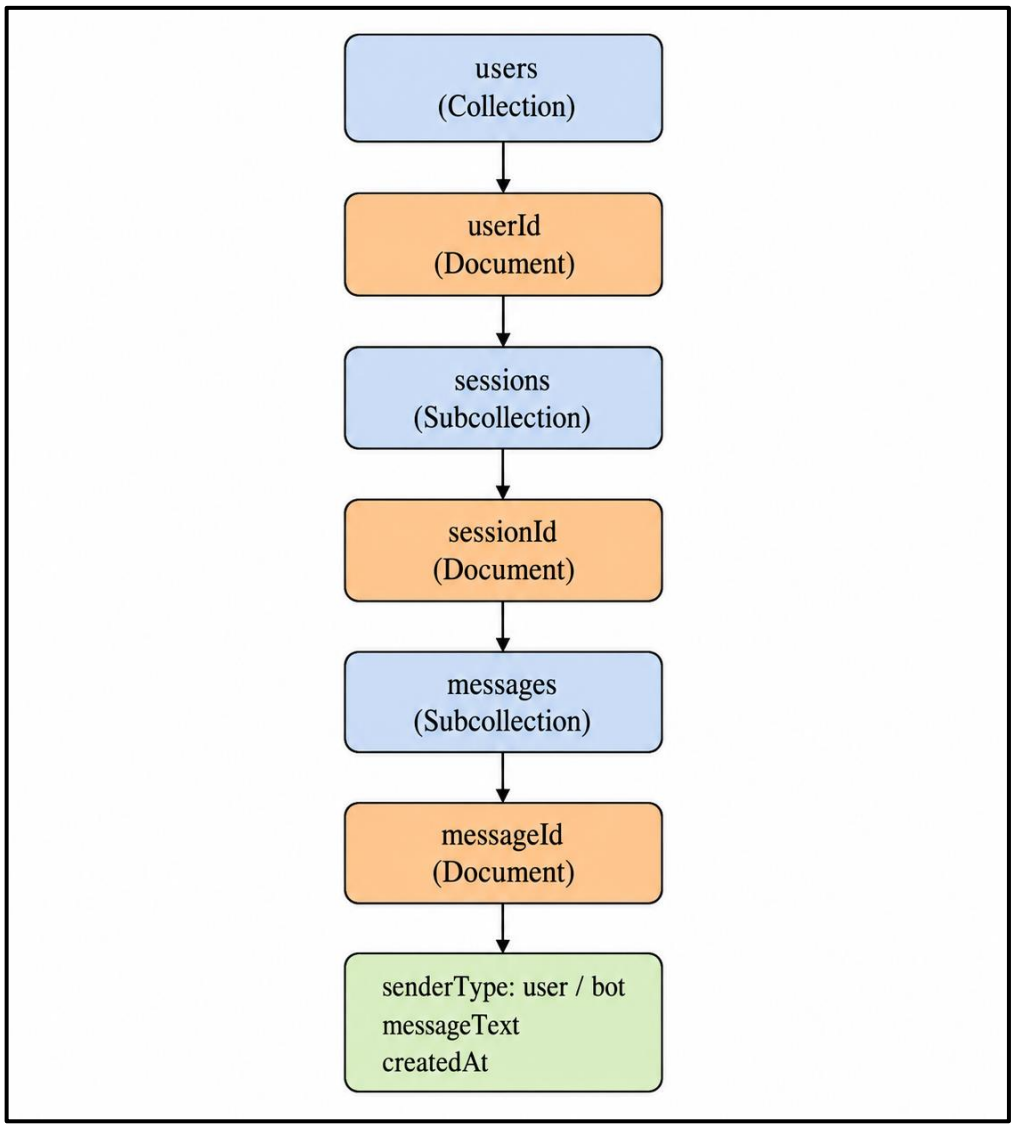
Basis data aplikasi menggunakan *cloud firestore*, yaitu basis data berbasis dokumen yang digunakan untuk menyimpan data aplikasi dalam struktur koleksi dan dokumen. Pada aplikasi *nutriscan*, *cloud firestore* dapat digunakan untuk menyimpan data yang diperlukan oleh sistem, seperti informasi pengguna, riwayat hasil deteksi, atau data pendukung fitur aplikasi. Struktur penyimpanan dirancang berdasarkan kebutuhan sistem agar proses pembacaan dan penulisan data dapat dilakukan secara efisien. Perancangan struktur data pada *cloud firestore* perlu memperhatikan jumlah pembacaan dokumen, ukuran respons, biaya akses, dan waktu pemrosesan agar penggunaan basis data menjadi lebih optimal (Kesavan et al., 2023).

3. Konfigurasi Proyek Menggunakan *Firebase CLI*

Firebase cli digunakan sebagai alat bantu berbasis baris perintah untuk menghubungkan lingkungan pengembangan lokal dengan proyek *firebase*. Alat ini digunakan untuk melakukan autentikasi akun, menginisialisasi layanan *firebase*, melihat daftar proyek, menjalankan *emulator* lokal, serta menerapkan konfigurasi layanan yang digunakan oleh aplikasi. Dengan demikian, *firebase cli* berfungsi

sebagai perangkat pengelolaan dan konfigurasi proyek, bukan sebagai tempat penyimpanan data aplikasi (Nguyen, 2022).

Secara keseluruhan, alat bantu sistem tersebut digunakan sesuai dengan fungsinya masing-masing. *flowchart*, *use case diagram*, dan diagram arsitektur digunakan pada tahap perancangan, *cloud firestore* digunakan sebagai basis data, sedangkan *firebase cli* digunakan untuk membantu konfigurasi dan pengelolaan proyek *firebase*. Pemisahan fungsi tersebut membuat proses pengembangan aplikasi *nutriscan* menjadi lebih jelas, sistematis, dan mudah diterapkan.



Sumber: Hasil perancangan peneliti (2026) berdasarkan *model* data *Cloud Firestore*, *Google Firebase*

Gambar 3.3 Struktur Basis Data *Cloud Firestore* di *Firebase* CLI

Gambar 3.3 menunjukkan rancangan struktur basis data *cloud firestore* yang digunakan untuk menyimpan riwayat interaksi pengguna dengan fitur *chatbot* pada aplikasi *nutriscan*. Struktur penyimpanan dimulai dari koleksi *users* yang memuat dokumen setiap pengguna berdasarkan *userid*. Pada *cloud firestore*, data disimpan dalam dokumen yang dikelompokkan ke dalam koleksi dan dapat dikembangkan menjadi struktur hierarkis menggunakan subkoleksi.

Setiap dokumen *userId* memiliki subkoleksi *sessions* yang digunakan untuk mengelompokkan riwayat percakapan pengguna. Di dalam subkoleksi tersebut terdapat dokumen sessionId yang mewakili satu sesi percakapan. Selanjutnya, setiap dokumen sesi memiliki subkoleksi *messages* yang menyimpan seluruh pesan pada sesi tersebut berdasarkan dokumen *messageid*. Pola koleksi, dokumen, dan subkoleksi memungkinkan data percakapan dikelompokkan secara hierarkis sesuai pengguna dan sesi penggunaan aplikasi.

Setiap dokumen *messageid* memiliki beberapa *field*, yaitu *sendertype* untuk membedakan pesan pengguna dan balasan *bot*, *messagetext* untuk menyimpan isi pesan, serta *createdat* untuk mencatat waktu pengiriman pesan. Ketiga *field* tersebut merupakan rancangan khusus peneliti yang disesuaikan dengan kebutuhan fitur *chatbot nutriscan*, bukan struktur wajib yang ditentukan oleh *cloud firestore*. Struktur tersebut memungkinkan aplikasi mengambil, mengurutkan, dan menampilkan kembali pesan berdasarkan pengguna serta sesi percakapan tertentu.

Secara keseluruhan, rancangan basis data tersebut membantu memisahkan data setiap pengguna, sesi, dan pesan secara terstruktur. Susunan ini menjadi dasar dalam proses penyimpanan serta pengambilan riwayat percakapan *chatbot* pada aplikasi *nutriscan* berbasis *flutter*. *Cloud firestore* sendiri dirancang sebagai basis data *nosql* tanpa *server* yang mendukung pengembangan aplikasi *mobile* dan pembaruan data secara waktu nyata.

4. Rancangan Antarmuka Pengguna (*User Interface Design*)

Perancangan antarmuka pengguna aplikasi *nutriscan* dilakukan menggunakan *figma* untuk menghasilkan rancangan *visual* dan *prototype* sebelum diterapkan ke dalam kode program. Tahap perancangan meliputi penentuan struktur halaman, tata letak komponen, navigasi, warna, ikon, tombol, ukuran teks, serta hubungan antartampilan aplikasi. Penggunaan *figma* membantu menyajikan

gambaran awal antarmuka aplikasi *mobile* sehingga kesesuaian desain, kebutuhan pengguna, dan alur interaksi dapat diperiksa sebelum memasuki tahap implementasi (Tawasuli & Huda, 2023).

Rancangan antarmuka *nutriscan* mencakup halaman *splash screen* (intro), *login* (masuk) dan *signup* (pendaftaran), halaman utama, pemindaian langsung, pemilihan gambar dari galeri, hasil identifikasi, riwayat deteksi, *chatbot*, serta profil pengguna. Setiap halaman dirancang dengan memperhatikan konsistensi tata letak, kejelasan informasi, kemudahan navigasi, dan ukuran komponen yang sesuai untuk perangkat *mobile*. Rancangan dari *figma* kemudian diimplementasikan menggunakan komponen antarmuka *flutter* dan disesuaikan dengan fungsi setiap halaman. Flutter menyediakan komponen untuk membangun serta menjalankan tugas-tugas antarmuka pengguna pada aplikasi *android* melalui satu kerangka pengembangan (Białkowski & Smołka, 2022).

Pada halaman pemindaian dan hasil deteksi, komponen antarmuka difokuskan pada kemudahan pengguna dalam mengambil atau memilih gambar, menjalankan proses identifikasi, serta memahami hasil yang diberikan oleh aplikasi. Informasi yang ditampilkan mencakup nama objek, nilai *confidence score*, tombol untuk mendengarkan *voice output*, dan akses menuju informasi tambahan melalui fitur *chatbot*. Penempatan komponen tersebut disusun agar fungsi utama aplikasi dapat ditemukan dan digunakan tanpa melalui tahapan navigasi yang terlalu panjang (von Wangenheim & Dirschnabel, 2023).

Secara keseluruhan, rancangan antarmuka pengguna menjadi acuan dalam proses implementasi tampilan aplikasi *nutriscan* menggunakan *flutter*. Penyusunan desain *visual* sebelum pengodean membantu menjaga konsistensi antartampilan, mengurangi perubahan struktur pada tahap implementasi, dan memastikan setiap fitur dapat diakses dengan mudah. Dengan rancangan yang terstruktur, aplikasi diharapkan mempunyai tampilan yang jelas, responsif, dan nyaman digunakan pada perangkat *android*.

3.7. Pengujian Sistem

Pengujian sistem dilakukan menggunakan pendekatan *black box testing*, yaitu metode pengujian perangkat lunak yang memfokuskan pada fungsi dan hasil

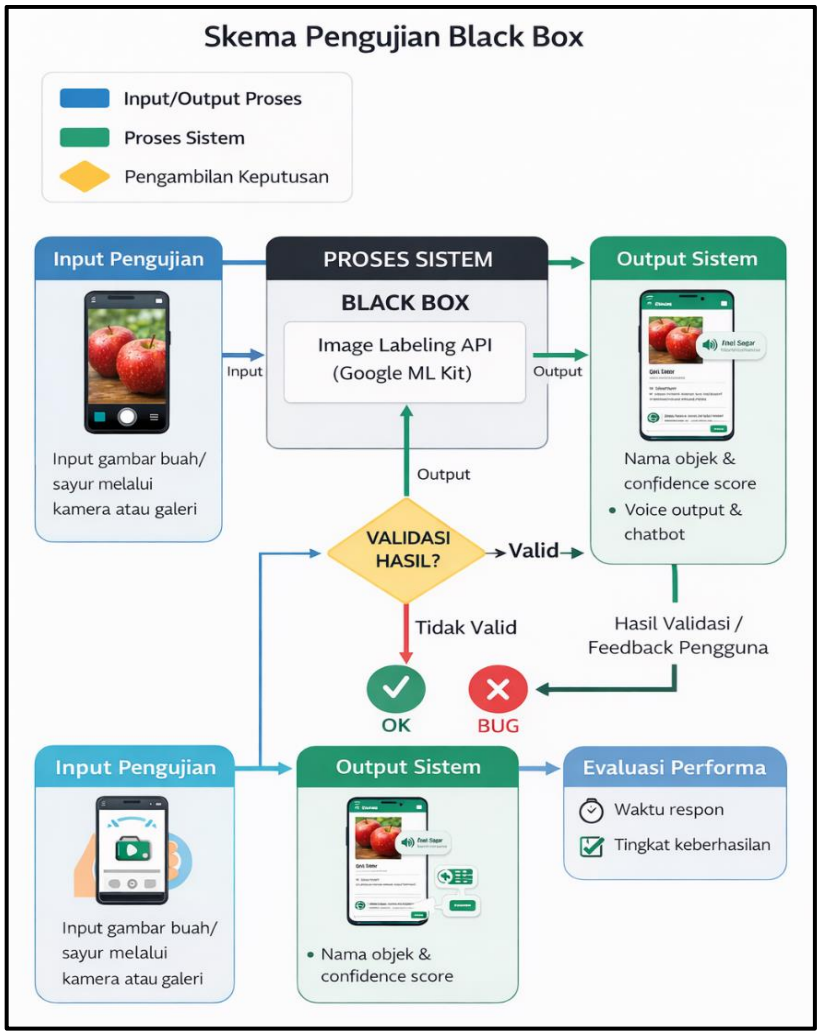
output tanpa mempertimbangkan struktur internal kode. Metode ini ideal ketika sistem diuji secara kuantitatif dari sisi pengguna untuk *input* diberikan, kemudian *output* diperiksa apakah sesuai spesifikasi atau kebutuhan (Kim et al., 2021).

Dalam praktiknya, pengujian dilakukan dengan membuat beberapa kasus uji (*test cases*) yang merepresentasikan situasi nyata: misalnya gambar buah dikenali dengan benar, gambar non-buah dikategorikan "tidak dikenal", atau uji keberagaman (variasi buah/sayur, kondisi pencahayaan, ukuran gambar). Teknik seperti *equivalence partitioning* atau *boundary value analysis* dapat digunakan untuk mengurangi jumlah pengujian namun tetap menjangkau seluruh domain input secara representatif. Studi pada aplikasi *mobile* sebelumnya membuktikan bahwa metode *black box* dengan *equivalence partitioning* cukup efektif untuk menguji fungsi dengan efisien (Ibrahim & Lee, 2023).

Untuk aplikasi berbasis *mobile* termasuk sistem deteksi buah dan sayuran serta pengujian ini juga perlu memperhatikan aspek *non*-fungsional seperti kecepatan respons, konsistensi hasil *across device*, dan stabilitas *ui* saat *input/output* diproses. Artikel tentang pengujian performa pada aplikasi *mobile* menyebut bahwa kombinasi pengujian fungsional dengan uji performa menjadi penting agar aplikasi tidak hanya benar secara logika, melainkan juga nyaman dan layak digunakan di banyak perangkat (Giani & Martin, 2021).

Sebagai kesimpulan, metode *black box testing* dipadukan dengan teknik *equivalence partitioning* dan pengujian performa menjadi pilihan paling cocok untuk penelitian ini. Pendekatan ini memungkinkan verifikasi objektif atas fungsi deteksi, keandalan hasil, serta respons aplikasi tanpa perlu membuka kode sumber. Hasil pengujian bisa dinyatakan dalam angka (akurasi, persentase keberhasilan, waktu respon), sehingga mendukung karakter penelitian kuantitatif yang kamu pilih.

Untuk memperjelas proses pengujian yang dilakukan, penelitian ini dilengkapi dengan skema pengujian *black box testing*. Skema ini menggambarkan alur pengujian sistem berdasarkan fungsi yang terlihat oleh pengguna tanpa melibatkan struktur kode internal aplikasi. Pendekatan ini bertujuan memastikan bahwa setiap fitur utama aplikasi berjalan sesuai dengan kebutuhan dan spesifikasi yang telah ditetapkan.



Sumber: Penulis, 2026

Gambar 3.4 Skema Pengujian Fungsional Aplikasi *FreshScan* Berbasis *Black Box Testing*

Pada Gambar 3.4 menunjukkan skema pengujian fungsional aplikasi *nutriscan* menggunakan metode *black box testing*. Pengujian dilakukan dengan memfokuskan pada kesesuaian antara input yang diberikan pengguna dan *output* yang dihasilkan sistem tanpa melihat struktur kode program. Adapun tahapan pengujian yang dilakukan adalah sebagai berikut:

1. *Input* Pengujian: Pengguna memberikan *input* berupa citra buah atau sayuran melalui kamera atau galeri pada perangkat *mobile*.
2. Proses Sistem (*Black Box*): Sistem memproses citra menggunakan *image labeling api* dari *google ml kit* tanpa melibatkan pengujian terhadap kode internal aplikasi.

3. *Output* Sistem: Aplikasi menampilkan hasil berupa nama objek, nilai *confidence score*, serta menyediakan fitur *voice output* dan *chatbot* sebagai informasi tambahan.
4. Validasi Hasil: *Output* yang dihasilkan dievaluasi apakah sesuai dengan *input* yang diberikan. Jika sesuai, hasil dinyatakan *valid;* jika tidak, dicatat sebagai kesalahan (*bug*).
5. Evaluasi Performa: Pengujian juga mencakup pengukuran waktu respon sistem dan tingkat keberhasilan deteksi sebagai indikator performa aplikasi.

BAB IV
ANALISA DAN PERANCANGAN

4.1. Analisa Sistem

Tahap analisa sistem merupakan langkah awal yang dilakukan sebelum proses perancangan dan pengembangan aplikasi dimulai. Analisa ini bertujuan untuk mengidentifikasi kondisi sistem yang sedang berjalan saat ini, mengevaluasi kelemahan yang ada, serta menentukan kebutuhan sistem baru yang akan dikembangkan. Dengan melakukan analisa secara menyeluruh, proses pengembangan aplikasi *nutriscan* dapat diarahkan secara tepat sasaran agar menghasilkan solusi yang relevan, efisien, dan sesuai dengan kebutuhan pengguna.

4.1.1. Analisa dan Evaluasi Sistem Berjalan

a. Analisa Sistem Berjalan

Analisa sistem berjalan dilakukan untuk memahami kondisi dan proses identifikasi buah dan sayuran yang selama ini dijalankan oleh pengguna sebelum aplikasi *nutriscan* dikembangkan. Berdasarkan hasil observasi dan wawancara awal kepada calon pengguna, ditemukan bahwa proses identifikasi buah dan sayuran masih dilakukan secara manual, yaitu dengan mengandalkan pengetahuan *visual* pribadi, bertanya kepada orang lain, atau melakukan pencarian melalui mesin telusur di internet. Kondisi ini mengandung beberapa kelemahan yang teridentifikasi, antara lain:

1. Proses identifikasi lambat dan tidak efisien. Pengguna harus membuka *browser*, mengetikkan deskripsi objek, dan menyaring hasil secara manual. Proses ini membutuhkan waktu yang tidak sedikit, terutama bagi pengguna yang belum familiar dengan teknologi.
2. Hasil tidak akurat dan bergantung pada subjektivitas. Identifikasi berbasis ingatan *visual* bersifat subyektif dan rawan keliru, khususnya untuk buah dan sayuran yang memiliki kemiripan bentuk atau warna, seperti terong ungu dengan terong hijau, atau mangga muda dengan jambu.

3. Tidak tersedia fitur aksesibilitas audio. Sistem manual maupun pencarian *web* tidak menyediakan *output* suara (*voice output*) secara otomatis, sehingga kurang mendukung pengguna dengan keterbatasan baca atau anak-anak yang belum lancar membaca.
4. Tidak ada interaksi lanjutan berbasis percakapan. Setelah menemukan nama objek, pengguna tidak memiliki fasilitas untuk menggali informasi lebih lanjut secara interaktif melalui *chatbot*.
5. Ketergantungan koneksi internet secara penuh. Pencarian informasi melalui *web* membutuhkan koneksi internet yang stabil, sehingga tidak dapat digunakan di lingkungan dengan jaringan terbatas.

Berdasarkan identifikasi kelima masalah di atas, dapat disimpulkan bahwa sistem berjalan memiliki keterbatasan yang cukup mendasar, baik dari sisi kecepatan, akurasi, aksesibilitas, maupun interaktivitas. Dikaitkan dengan metode pengembangan sistem *prototype* yang digunakan dalam penelitian ini, kondisi sistem berjalan menunjukkan adanya kebutuhan mendesak untuk menghadirkan sistem baru yang mampu melakukan identifikasi objek secara otomatis, cepat, *on-device*, dan interaktif tanpa harus bergantung pada proses manual atau koneksi internet yang terus-menerus.

b. Evaluasi Sistem Berjalan

Setelah permasalahan pada sistem berjalan teridentifikasi, tahap selanjutnya adalah mengevaluasi dan merumuskan solusi yang tepat untuk mengatasi setiap kelemahan tersebut. Evaluasi ini menjadi dasar dalam menentukan fitur-fitur yang akan dikembangkan pada aplikasi *nutriscan*. Tabel berikut merangkum pemetaan antara masalah sistem berjalan dan solusi yang diusulkan.

Tabel 4.1 Evaluasi Sistem Berjalan

No	Masalah Sistem Berjalan	Solusi yang Diusulkan
1	Identifikasi manual lambat dan tidak efisien	Deteksi otomatis menggunakan *google ml kit (image labelling)* yang mampu menghasilkan label objek dalam hitungan detik
2	Hasil tidak akurat, bergantung subjektivitas	Memanfaatkan *model machine learning* pada *google ml kit* yang terlatih dengan *dataset* besar sehingga menghasilkan *confidence score* terukur

3	Tidak ada fitur aksesibilitas *audio*	Mengintegrasikan fitur *voice output* menggunakan *flutter tts* (*Text-to-Speech*) untuk menyampaikan hasil deteksi secara *audio*
4	Tidak ada interaksi lanjutan	Mengintegrasikan fitur *chatbot* sebagai media tanya jawab lanjutan terkait hasil identifikasi buah dan sayuran
5	Bergantung penuh pada koneksi internet	Menggunakan *on-device processing* sehingga fitur deteksi utama dapat berjalan tanpa koneksi internet

Sumber: Penulis, 2026

Berdasarkan Tabel 4.1 setiap permasalahan yang ditemukan pada sistem berjalan telah dipetakan ke dalam solusi teknis yang konkret. Solusi-solusi tersebut kemudian diimplementasikan dalam bentuk aplikasi *mobile nutriscan* berbasis *flutter* yang mengintegrasikan *google ml kit* sebagai mesin klasifikasi *visual*, *voice output* sebagai media penyampaian hasil secara *audio*, serta *chatbot* sebagai antarmuka interaksi lanjutan. Dengan pendekatan metode *prototype*, seluruh solusi ini dikembangkan secara bertahap melalui proses evaluasi dan perbaikan berulang hingga menghasilkan sistem yang sesuai dengan kebutuhan pengguna.

4.1.2. Analisa Kebutuhan

Analisa kebutuhan dilakukan untuk menentukan secara rinci apa saja yang dibutuhkan oleh sistem agar dapat berfungsi dengan baik dan memenuhi tujuan pengembangan. Analisa ini mencakup kebutuhan pengguna, kebutuhan data, kebutuhan fungsional, kebutuhan *non*-fungsional, serta spesifikasi perangkat keras dan lunak yang diperlukan. Hasil dari analisa kebutuhan ini menjadi landasan utama dalam tahap perancangan sistem yang akan dilakukan selanjutnya.

a. Kebutuhan Pengguna (*User*)

Identifikasi kebutuhan pengguna dilakukan untuk menentukan siapa saja yang akan berinteraksi dengan sistem dan apa yang mereka butuhkan dari aplikasi *nutriscan*. Pengguna sistem dikategorikan ke dalam dua aktor sebagaimana ditunjukkan pada tabel berikut.

Tabel 4.2 Kebutuhan Pengguna

No	Aktor	Deskripsi
1	Pengguna Umum (*User*)	Masyarakat umum, pelajar, anak-anak, atau konsumen yang ingin mengidentifikasi jenis buah dan sayuran melalui kamera atau galeri *smartphone* secara cepat dan mudah

2	*Admin* / *Developer*	Pengembang yang mengelola konfigurasi sistem, memperbarui integrasi *api*, serta memantau performa aplikasi melalui *firebase console*

Sumber: Penulis, 2026

Berdasarkan Tabel 4.2 aktor utama pada sistem ini adalah Pengguna Umum, karena seluruh fitur fungsional aplikasi yaitu deteksi gambar, *voice output*, dan *chatbot* dirancang untuk memenuhi kebutuhan interaksi pengguna secara langsung. Sementara itu, *admin*/*developer* berperan di balik layar untuk memastikan sistem tetap berjalan optimal.

b. Kebutuhan Data

Kebutuhan data menjelaskan jenis-jenis data yang diperlukan oleh sistem *nutriscan* agar dapat berfungsi secara optimal, mulai dari data yang menjadi *input* proses deteksi hingga data yang disimpan untuk keperluan riwayat dan interaksi *chatbot*. Identifikasi kebutuhan data yang tepat sangat penting untuk memastikan alur informasi dalam sistem berjalan secara efisien dan terstruktur.

Tabel 4.3 Kebutuhan Data Sistem *NutriScan*

No	Jenis Data	Sumber Data	Tujuan
1	Citra gambar buah/sayuran	Input pengguna (kamera / galeri)	Diproses oleh *google ml kit* untuk menghasilkan label dan *confidence score*
2	Label hasil deteksi	*Google ML Kit Image Labeling API*	Ditampilkan kepada pengguna sebagai hasil identifikasi objek
3	*Confidence score*	*Google ML Kit Image Labeling API*	Menunjukkan tingkat keyakinan hasil deteksi kepada pengguna
4	Data riwayat deteksi	Aplikasi (*Firestore Database*)	Menyimpan *log* hasil deteksi pengguna untuk keperluan referensi atau histori
5	Data percakapan *chatbot*	*Input* pengguna + *Firestore*	Menyimpan sesi percakapan *chatbot* untuk memberikan respons kontekstual
6	Data pengguna (opsional)	*Input* pengguna saat registrasi	Digunakan untuk personalisasi riwayat dan pengalaman pengguna

Sumber: Penulis, 2026

Berdasarkan Tabel 4.3 data utama yang menjadi inti sistem adalah citra gambar buah dan sayuran sebagai *input*, serta label hasil deteksi beserta *confidence score* sebagai output utama yang ditampilkan kepada pengguna. Data riwayat deteksi dan percakapan *chatbot* disimpan di *firebase firestore* untuk mendukung fitur histori dan interaksi lanjutan.

c. Kebutuhan Fungsional

Kebutuhan fungsional mendefinisikan fungsi-fungsi apa saja yang wajib tersedia dan dapat dijalankan oleh sistem *nutriscan*. Setiap fungsi diidentifikasi berdasarkan kebutuhan pengguna yang telah dianalisis sebelumnya dan diberikan kode unik sebagai acuan dalam proses pengembangan dan pengujian sistem.

Tabel 4.4 Kebutuhan Fungsional Sistem *NutriScan*

No	Kode	Fungsi	Deskripsi
1	F-01	Registrasi Akun	Pengguna baru dapat membuat akun menggunakan *email* dan *password* melalui *firebase authentication*
2	F-02	*Login*	Pengguna terdaftar dapat masuk ke aplikasi menggunakan *email* dan *password* yang telah didaftarkan
3	F-03	*Logout*	Pengguna yang sedang *login* dapat keluar dari sesi aktif dan diarahkan kembali ke halaman *login*
4	F-04	Deteksi *via* Kamera	Pengguna dapat mengambil gambar secara langsung menggunakan kamera perangkat untuk dideteksi
5	F-05	Deteksi *via* Galeri	Pengguna dapat mengunggah gambar dari galeri perangkat untuk dideteksi
6	F-06	*Image Labeling*	Sistem memproses gambar menggunakan *google ml kit* dan menampilkan label serta *confidence score*
7	F-07	*Voice Output*	Sistem membacakan hasil deteksi secara *audio* menggunakan fitur *text-to-speech*
8	F-08	*Chatbot* Interaktif	Pengguna dapat berinteraksi melalui *chatbot* untuk mendapatkan informasi tambahan terkait objek yang terdeteksi
9	F-09	Riwayat Deteksi	Sistem menyimpan dan menampilkan riwayat hasil deteksi yang pernah dilakukan pengguna
10	F-10	Halaman *Splash & Onboarding*	Aplikasi menampilkan halaman pembuka dan panduan singkat penggunaan saat pertama kali dijalankan
11	F-11	Notifikasi Tidak Terdeteksi	Sistem menampilkan pesan khusus apabila objek tidak dapat dikenali sebagai buah atau sayuran

Sumber: Penulis, 2026

Berdasarkan Tabel 4.4 terdapat 11 fungsi utama yang harus dipenuhi oleh sistem *nutriscan*. Fungsi F-01 hingga F-03 mencakup sistem autentikasi pengguna yang memastikan data riwayat dan percakapan tersimpan secara personal per akun. Fungsi F-04 hingga F-06 merupakan inti dari sistem deteksi, sementara F-07 dan F-08 merupakan fitur pendukung yang meningkatkan aksesibilitas dan interaktivitas aplikasi. Fungsi F-09 hingga F-11 memastikan pengalaman pengguna yang lebih lengkap melalui fitur riwayat, tampilan awal, dan penanganan kondisi gagal deteksi.

d. Kebutuhan *Non*-Fungsional

Selain kebutuhan fungsional, sistem *nutriscan* juga harus memenuhi sejumlah kebutuhan *non*-fungsional yang berkaitan dengan kualitas sistem secara keseluruhan. Kebutuhan non-fungsional menentukan bagaimana sistem harus beroperasi, bukan hanya apa yang harus dilakukan oleh sistem. Aspek-aspek ini mencakup performa, keamanan, kemudahan penggunaan, dan kompatibilitas perangkat.

Tabel 4.5 Kebutuhan Non-Fungsional

No	Aspek	Kebutuhan
1	Performa	Proses deteksi harus menghasilkan respons dalam waktu ≤ 3 detik pada perangkat dengan spesifikasi minimum
2	Keamanan	Data pengguna disimpan secara aman di *firebase* dengan aturan akses (*security rules*) yang sesuai
3	Kemudahan Penggunaan (*Usability*)	Antarmuka dirancang sederhana dan intuitif sehingga dapat digunakan oleh semua kalangan, termasuk anak-anak dan lansia
4	Ketersediaan (*Availability*)	Fitur deteksi utama dapat berjalan secara *on-device* tanpa koneksi internet; fitur *chatbot* memerlukan koneksi
5	Kompatibilitas	Aplikasi mendukung perangkat *android* minimal versi 8.0 (*oreo*) *api* level 26 ke atas
6	Skalabilitas	Arsitektur sistem memungkinkan penambahan fitur baru tanpa mengubah struktur inti aplikasi
7	Aksesibilitas	Fitur *voice output* memastikan hasil deteksi dapat diakses oleh pengguna dengan keterbatasan baca

Sumber: Penulis, 2026

Berdasarkan Tabel 4.5 aspek performa menjadi prioritas utama mengingat aplikasi ini dirancang untuk penggunaan *real-time*. Batas waktu respons ≤ 3 detik ditetapkan sebagai standar minimum agar pengguna tidak mengalami ketidaknyamanan saat menunggu hasil deteksi. Aspek ketersediaan (*availability*) juga menjadi perhatian khusus karena fitur deteksi utama harus tetap dapat digunakan meskipun pengguna berada di area dengan koneksi internet yang terbatas.

e. Analisa Perangkat Keras dan Lunak (*Hardware*/*Software*)

Analisa perangkat keras dan lunak bertujuan untuk menentukan spesifikasi minimum yang dibutuhkan agar aplikasi *nutriscan* dapat berjalan dengan baik, baik dari sisi pengguna maupun dari sisi pengembang. Analisa ini dibagi menjadi tiga bagian, yaitu spesifikasi perangkat keras untuk pengguna, spesifikasi perangkat

keras untuk pengembangan, dan spesifikasi perangkat lunak yang digunakan dalam proses pengembangan sistem.

Tabel 4.6 Spesifikasi Perangkat Keras Minimum untuk Pengguna

No	Komponen	Spesifikasi Minimum
1	Perangkat	*Smartphone Android*
2	Sistem Operasi	*Android* 8.0 (*Oreo*) / *API* Level 26
3	*RAM*	2 *GB*
4	Penyimpanan Internal	100 *MB* (ruang kosong untuk instalasi)
5	Kamera	Kamera belakang minimal 5 *MP*
6	Koneksi	*Wi-Fi* / Data seluler (untuk fitur *chatbot*)

Sumber: Penulis, 2026

Tabel 4.6 menunjukkan spesifikasi minimum perangkat yang diperlukan oleh pengguna untuk menjalankan aplikasi *nutriscan*. Spesifikasi ini ditetapkan berdasarkan kebutuhan sistem deteksi *on-device* menggunakan *google ml kit* yang memerlukan kemampuan pemrosesan gambar secara lokal di perangkat. Kamera belakang dengan resolusi minimal 5 *mp* (*Mega Pixel*) diperlukan untuk memastikan kualitas gambar yang cukup baik sehingga proses *image labeling* dapat menghasilkan hasil deteksi yang akurat.

Tabel 4.7 Spesifikasi Perangkat Keras Pengembangan

No	Komponen	Spesifikasi
1	*Prosesor*	*Intel Core i5* / *AMD Ryzen 5* atau setara
2	*RAM*	8 *GB* (minimum)
3	Penyimpanan	256 *GB SSD*
4	Sistem Operasi	*Windows 10/11 64-bit*

Sumber: Penulis, 2026

Tabel 4.7 menjabarkan spesifikasi perangkat keras yang digunakan oleh pengembang selama proses pembangunan aplikasi. Spesifikasi ini diperlukan untuk mendukung proses kompilasi *flutter*, menjalankan *emulator android*, serta mengelola proyek *firebase* secara bersamaan tanpa mengalami kendala performa.

Tabel 4.8 Spesifikasi Perangkat Lunak Pengembangan

No	Perangkat Lunak	Versi / Keterangan
1	*Flutter SDK*	3.x (*stable channel*)
2	*Dart SDK*	*Bundled* dengan *flutter*
3	*Android Studio* / *VS Code*	Versi terbaru
4	*Google ML Kit* (*google_mlkit_image_labeling*)	Versi terbaru di *pub.dev*

5	*Firebase CLI*	Versi terbaru
6	*Flutter TTS* (*flutter_tts*)	Untuk fitur *voice output*
7	*Android SDK*	*Api level 26* ke atas
8	*Git*	Untuk *version control*

Sumber: Penulis, 2026

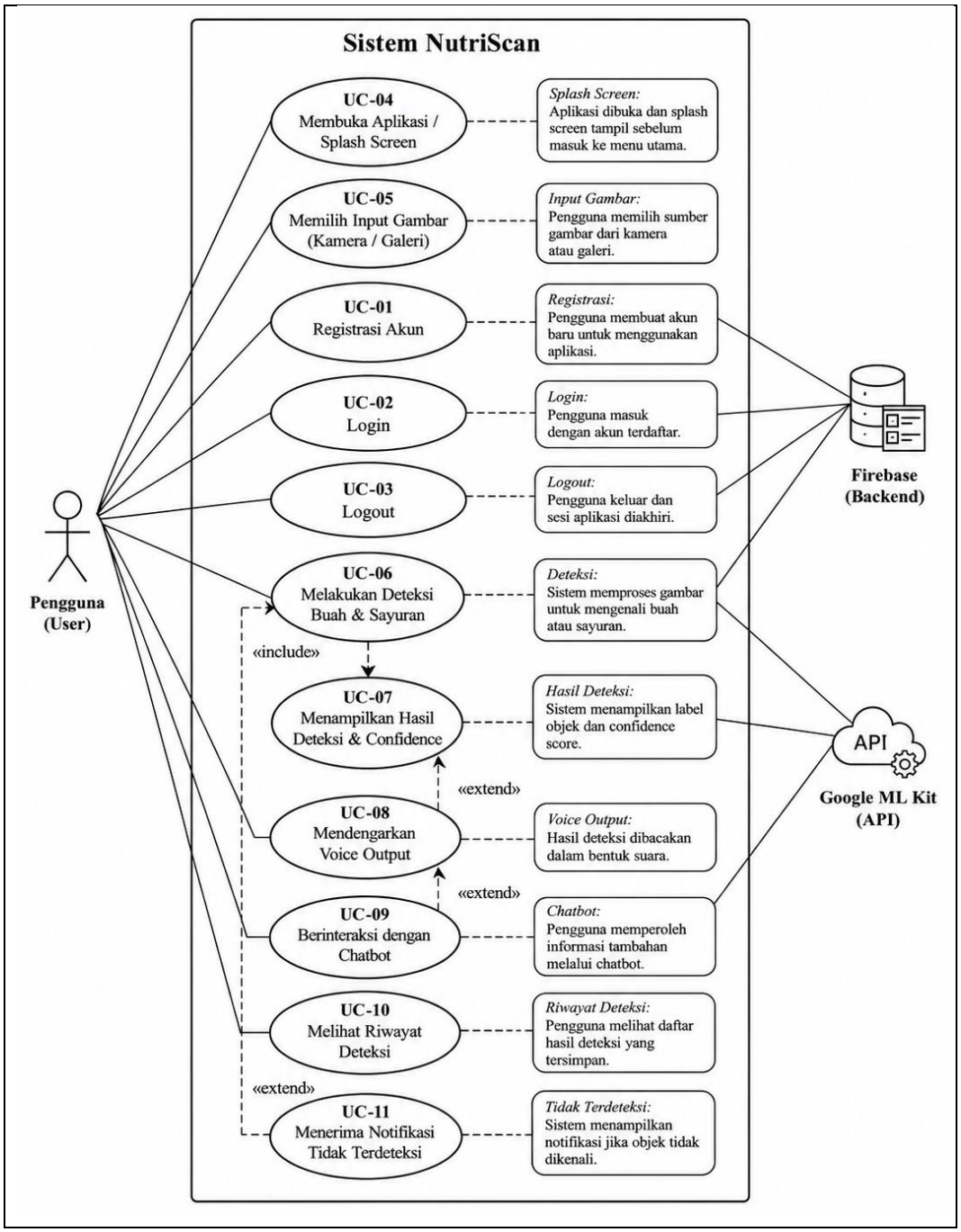
Berdasarkan Tabel 4.8 perangkat lunak yang digunakan dalam pengembangan aplikasi *nutriscan* terdiri atas komponen inti dan komponen pendukung. *Flutter sdk* dan *dart sdk* merupakan fondasi utama pengembangan, sementara *google ml kit* menjadi komponen kunci untuk fitur deteksi objek. *Firebase cli* digunakan untuk mengelola layanan *backend* berbasis *cloud*, sedangkan *git* berperan sebagai alat *version control* untuk menjaga konsistensi dan keterlacakan perubahan kode selama proses pengembangan berlangsung. Secara keseluruhan, kombinasi perangkat lunak ini membentuk ekosistem pengembangan yang lengkap dan terintegrasi untuk menghasilkan aplikasi *nutriscan* yang fungsional dan berkualitas.

4.2. Perancangan Sistem (Desain) yang diusulkan

4.2.1. Perancangan Sistem (Logis)

a. *Use Case Diagram*

Use case diagram menggambarkan interaksi antara aktor dengan sistem melalui fungsi-fungsi utama yang tersedia dalam aplikasi *nutriscan*. Terdapat tiga aktor yang terlibat dalam sistem ini, yaitu Pengguna sebagai aktor utama yang berinteraksi langsung dengan seluruh fitur aplikasi, *google ml kit* sebagai sistem eksternal yang memproses citra dan mengembalikan label hasil deteksi, serta *firebase* sebagai sistem eksternal yang menyimpan data riwayat deteksi dan sesi *chatbot*.



Sumber: Penulis, 2026

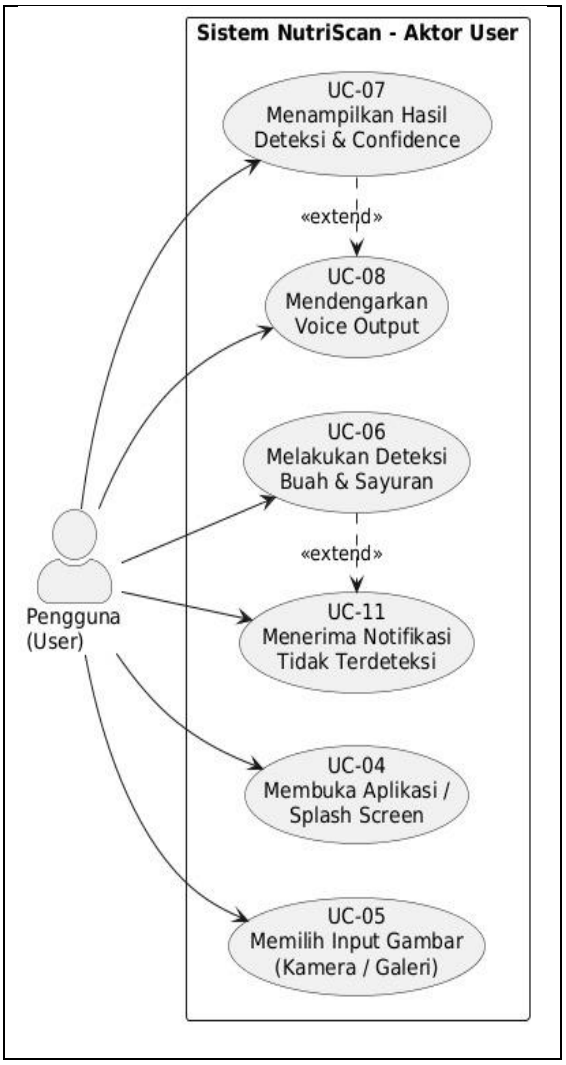
Gambar 4.1 *Use Case* Diagram Aplikasi *NutriScan*

Berdasarkan Gambar 4.1 sistem *nutriscan* memiliki satu aktor utama, yaitu pengguna (*user*), serta dua sistem pendukung, yaitu *firebase* sebagai *backend* dan *google ml kit* sebagai *api*. Pengguna dapat menjalankan sebelas fungsi utama yang meliputi membuka aplikasi melalui *splash screen*, memilih sumber gambar dari kamera atau galeri, melakukan registrasi, *login*, *logout*, mendeteksi buah dan

sayuran, melihat hasil deteksi beserta *confidence score*, mendengarkan *voice output*, berinteraksi dengan *chatbot*, melihat riwayat deteksi, dan menerima notifikasi apabila objek tidak berhasil dikenali.

Firebase berperan dalam mendukung pengelolaan autentikasi pengguna, sesi akun, penyimpanan riwayat deteksi, serta data interaksi yang diperlukan aplikasi. Sementara itu, *google ml kit* digunakan pada proses identifikasi gambar melalui fitur *image labeling*. Gambar yang diperoleh dari kamera atau galeri diproses oleh *google ml kit* untuk menghasilkan label objek dan nilai *confidence score*, kemudian hasil tersebut ditampilkan kepada pengguna melalui antarmuka aplikasi *nutriscan*.

Relasi <<*include*>> menunjukkan bahwa proses UC-06 melakukan deteksi buah dan sayuran selalu menyertakan uc-07 menampilkan hasil deteksi dan *confidence*, karena hasil klasifikasi harus ditampilkan setelah gambar selesai diproses. Adapun relasi <<*extend*>> menunjukkan bahwa fitur UC-08 mendengarkan *voice output*, UC-09 berinteraksi dengan *chatbot*, dan UC-11 menerima notifikasi tidak terdeteksi dijalankan secara kondisional. *Voice output* dan *chatbot* dapat digunakan setelah hasil deteksi ditampilkan, sedangkan notifikasi tidak terdeteksi muncul apabila sistem tidak berhasil mengenali gambar sebagai buah atau sayuran. Secara keseluruhan, diagram tersebut menggambarkan hubungan antara pengguna, fungsi-fungsi aplikasi, *firebase*, dan *google ml kit* dalam mendukung proses kerja *nutriscan*.

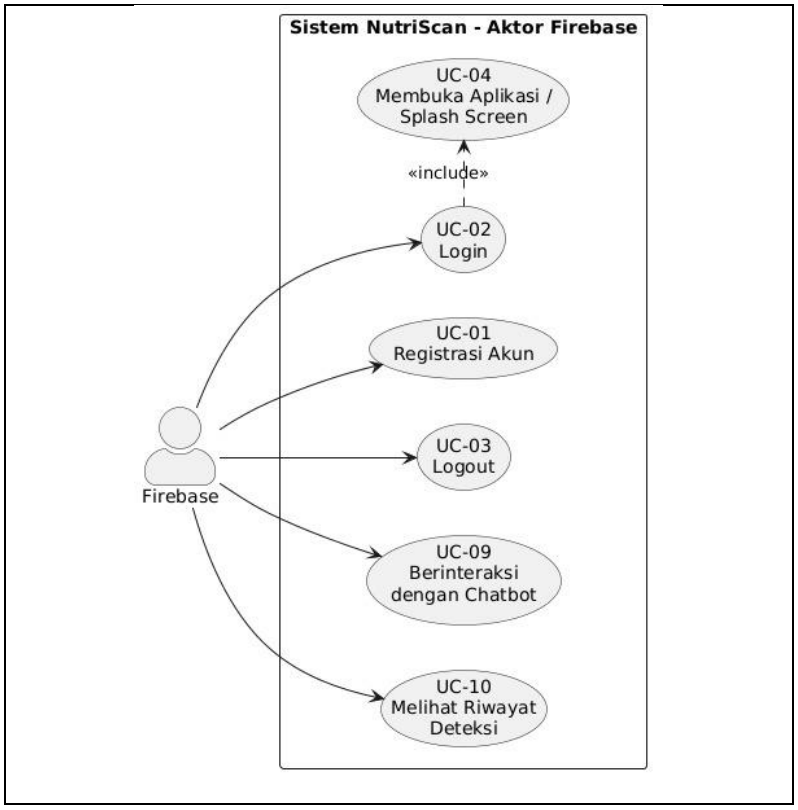


Sumber: Penulis, 2026

Gambar 4.2 *Use Case* Diagram pada *User* di Aplikasi *NutriScan*

Berdasarkan Gambar 4.2 pengguna (*User*) menjadi aktor utama dalam enam *use case* sistem *nutriscan*. Pada UC-04 (*Splash Screen*), aplikasi menampilkan layar pembuka sambil memeriksa ketersediaan sesi autentikasi lokal; jika sesi *valid* maka pengguna diarahkan ke halaman utama, sedangkan jika tidak ada sesi atau token sudah kadaluarsa maka pengguna diarahkan ke halaman *login* — *use case* ini selalu dipicu secara otomatis setelah UC-02 *login* berhasil. Pada UC-05 (Memilih *Input* Gambar), pengguna memilih sumber gambar melalui dialog pilihan kamera atau galeri;

1. sistem meminta izin akses terlebih dahulu dan jika izin ditolak maka tampil pesan permintaan izin, sementara jika pengguna membatalkan proses maka sistem kembali ke halaman utama.
2. Pada UC-06 (Deteksi Buah & Sayuran), sistem meneruskan gambar yang dipilih ke *model ml kit* untuk diklasifikasikan, dan hasilnya selalu diteruskan ke UC-07; jika *confidence* di bawah 70%, sistem secara otomatis memicu UC-11.
3. Pada UC-07 (Tampil Hasil Deteksi), sistem menyajikan nama buah/sayuran, *confidence score.*

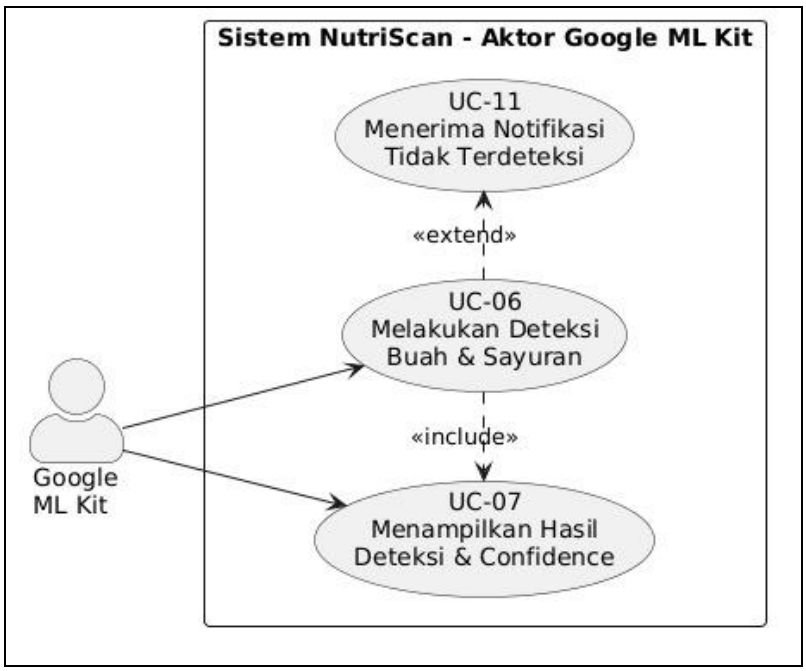


Sumber: Penulis, 2026

Gambar 4.3 *Use Case* Diagram pada *Firebase* di Aplikasi *NutriScan*

Berdasarkan Gambar 4.3 *firebase* berperan sebagai aktor pendukung dalam enam *use case* sistem *nutriscan*. Pada UC-01 (Registrasi Akun), *firebase authentication* menerima dan menyimpan data akun baru setelah pengguna mengisi formulir pendaftaran; jika *email* sudah terdaftar atau koneksi terputus, sistem menampilkan pesan kesalahan yang sesuai, yaitu:

1. Pada UC-02 (*Login*), *firebase* memverifikasi kredensial pengguna dan mengembalikan token sesi, yang kemudian disimpan secara lokal sebelum sistem secara otomatis memicu UC-04 (*Splash Screen*) sebagai transisi ke halaman utama; bila token yang tersimpan sudah kadaluarsa, sesi dihapus dan pengguna diarahkan kembali ke halaman *login*.
2. Pada UC-03 (*Logout*), *firebase* mencabut token autentikasi setelah pengguna mengkonfirmasi aksi *logout*, dan sesi lokal tetap dihapus meskipun koneksi internet tidak tersedia.
3. Pada UC-09 (Berinteraksi dengan *Chatbot*), *firebase* meneruskan pertanyaan pengguna beserta konteks hasil deteksi ke *gemini api* untuk menghasilkan respons berbasis *ai*; *use case* ini bersifat opsional tetapi sangat diperlukan jika *user* belum memahami di bagian halaman hasil deteksinya.



Sumber: Penulis, 2026

Gambar 4.4 *Use Case* Diagram pada *Google* ML Kit di Aplikasi *NutriScan*

Berdasarkan Gambar 4.4 *google ml kit* berperan sebagai aktor pendukung dalam tiga *use case* sistem *nutriscan* dengan mode inferensi *on-device* sehingga tidak memerlukan koneksi internet. Pada UC-06 (Melakukan Deteksi Buah & Sayuran), sistem meneruskan gambar yang telah dipilih pengguna melalui UC-05 ke *model* klasifikasi kustom *ml kit image labeling*; *model* memproses gambar dan

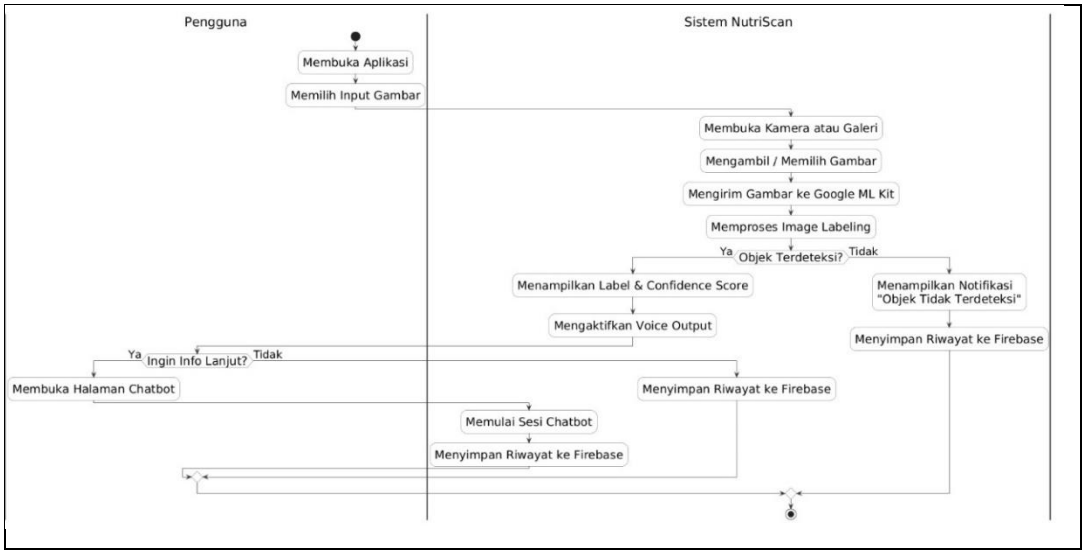
mengembalikan daftar label beserta *confidence score*, kemudian sistem mengambil label dengan nilai tertinggi — jika gambar *corrupt* atau terlalu kecil sistem menampilkan pesan *error* dan meminta gambar baru, sedangkan jika *model* belum siap sistem menampilkan indikator loading. Hasil dari UC-06 selalu diteruskan ke UC-07 melalui relasi *include*. Pada UC-07 (Menampilkan Hasil Deteksi & *Confidence*), *ml kit* menyediakan data label dan *confidence score* yang kemudian disajikan sistem dalam format persentase bersama informasi nilai gizi; jika data gizi tidak ditemukan di *database* maka ditampilkan pesan "Data gizi tidak tersedia".

b. ***Activity Diagram***

Activity diagram menggambarkan alur aktivitas pada setiap proses utama dalam sistem *nutriscan*. Terdapat dua *activity diagram* yang disusun dalam penelitian ini, yaitu diagram untuk proses deteksi objek dan diagram untuk proses percakapan *chatbot*.

1. *Activity Diagram I* – Proses Deteksi Objek

Activity diagram proses deteksi menggambarkan seluruh langkah dari pengguna membuka aplikasi, memilih input gambar, hingga sistem menampilkan hasil deteksi dan memberikan opsi interaksi lanjutan melalui *voice output* dan *chatbot*.



Sumber: Penulis, 2026

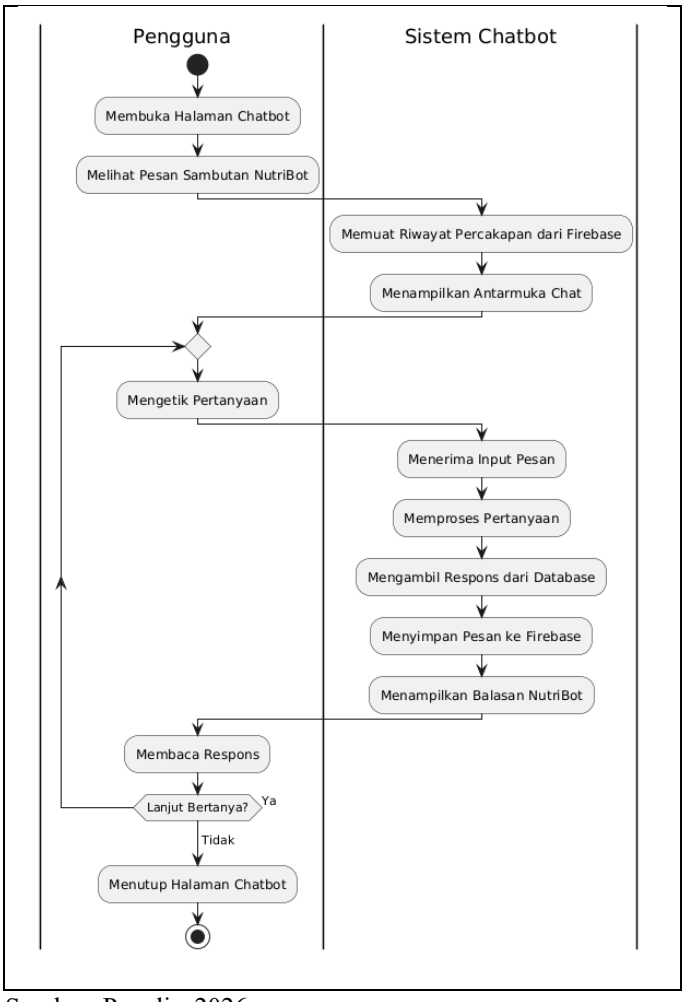
Gambar 4.5 *Activity* Diagram – Proses Deteksi Buah dan Sayuran

Berdasarkan Gambar 4.5 alur proses deteksi dimulai dari pengguna membuka aplikasi dan memilih metode *input* gambar. Sistem kemudian memproses citra melalui *google ml kit*. Terdapat dua kondisi percabangan utama:

- Apabila objek berhasil terdeteksi maka sistem menampilkan label dan *confidence score* serta mengaktifkan *voice output*, sedangkan apabila tidak terdeteksi sistem menampilkan notifikasi.
- Seluruh hasil deteksi disimpan ke *firebase firestore* sebagai riwayat pengguna.

2. *Activity Diagram II* – Proses *Chatbot*

Activity diagram proses *chatbot* menggambarkan alur aktivitas percakapan antara pengguna dengan *chatbot nutribot*, mulai dari membuka halaman *chatbot* hingga pengguna menutup sesi percakapan.



Sumber: Penulis, 2026

Gambar 4.6 *Activity* Diagram – Proses Percakapan *Chatbot NutriBot*

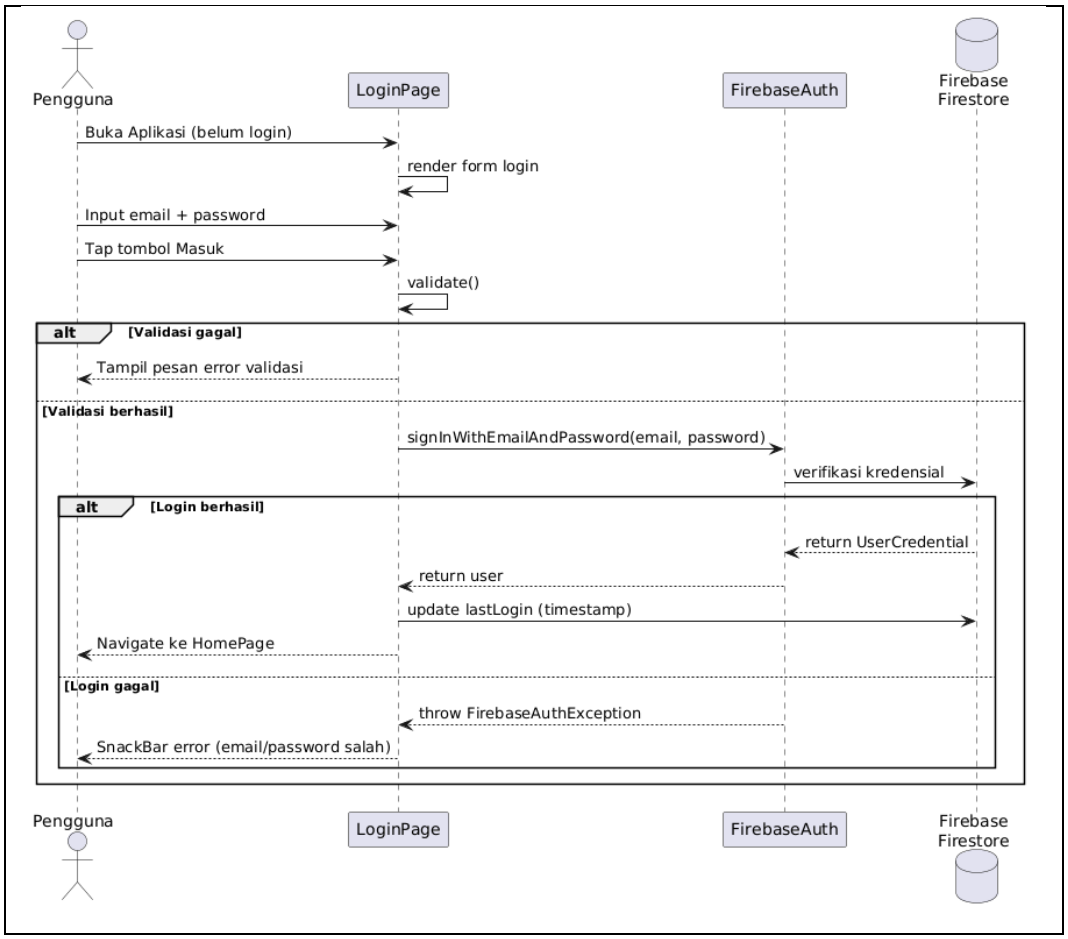
Berdasarkan Gambar 4.6 alur proses *chatbot* dimulai dari pengguna membuka halaman *chatbot* dan sistem memuat riwayat percakapan dari *firebase*. Pengguna dapat mengirim pertanyaan secara berulang selama sesi berlangsung. Setiap pesan yang dikirim diproses oleh sistem, respons diambil dari *database*, disimpan ke *firebase*, dan ditampilkan kepada pengguna sebagai balasan *nutribot*. Proses berulang selama pengguna masih ingin bertanya dan berakhir ketika pengguna menutup halaman *chatbot*.

c. *Sequence Diagram*

Sequence diagram menampilkan urutan interaksi antar komponen sistem dalam skenario utama aplikasi *nutriscan* secara kronologis. Terdapat dua *sequence diagram* yang disusun, yaitu skenario deteksi via kamera dan skenario penggunaan *chatbot*.

1. *Sequence Diagram I – Login*

Sequence diagram berikut menggambarkan urutan interaksi antar komponen dalam skenario pengguna melakukan *login* ke aplikasi *nutriscan* menggunakan *email* dan *password*.



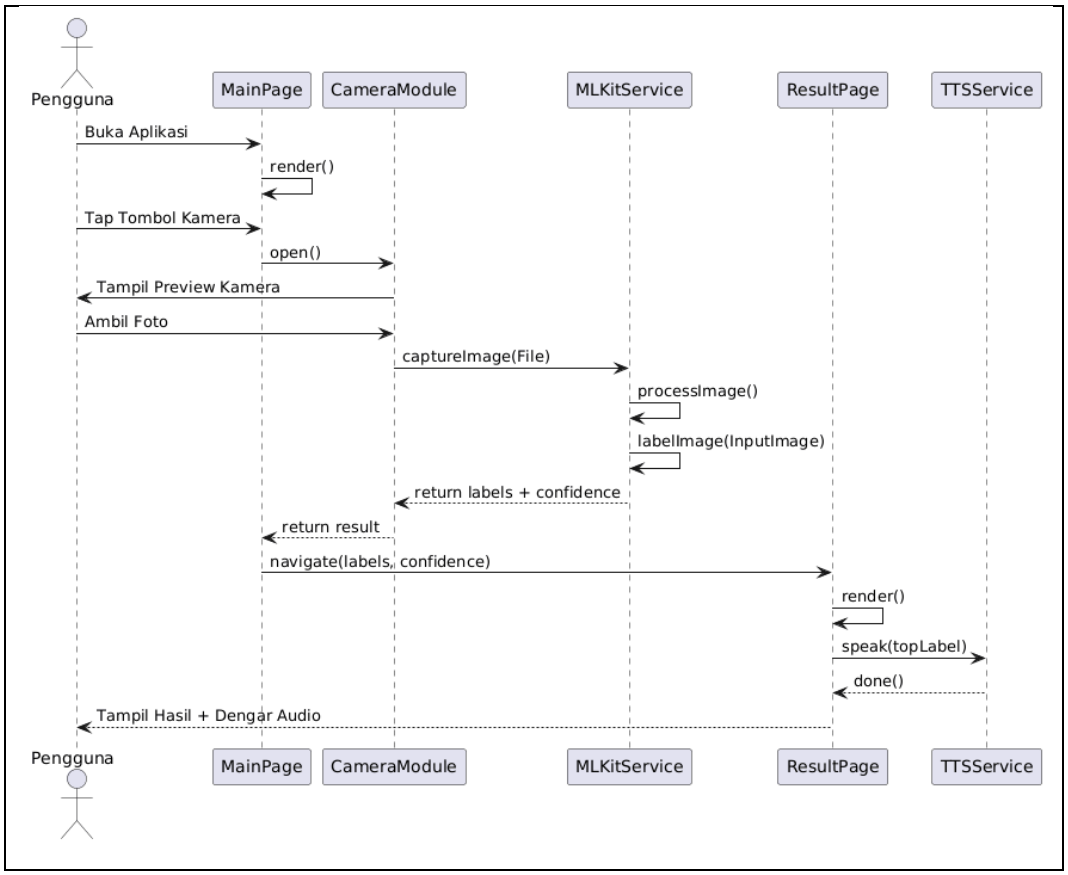
Sumber: Penulis, 2026

Gambar 4.7 *Sequence* Diagram – Proses *Login*

Berdasarkan diagram, proses login diawali dengan pengguna membuka aplikasi pada kondisi belum *login* sehingga sistem menampilkan `*LoginPage*`. Setelah pengguna mengisi *email* dan *password* lalu menekan tombol "Masuk", sistem melakukan validasi lokal terlebih dahulu. Apabila validasi berhasil, sistem memanggil `*FirebaseAuth.signInWithEmailAndPassword()*`. Jika kredensial *valid*, *Firebase* mengembalikan `*UserCredential*` dan pengguna diarahkan ke `*HomePage*`. Apabila gagal, `*FirebaseAuthException*` ditangkap dan pesan *error* ditampilkan melalui `*SnackBar*`.

2. *Sequence Diagram II* – Deteksi via Kamera

Sequence diagram ini menggambarkan komunikasi berurutan antara pengguna, *MainPage*, *CameraModule*, *MLKitService*, *ResultPage*, dan *TTSService* dalam skenario pengguna melakukan deteksi objek melalui kamera perangkat.



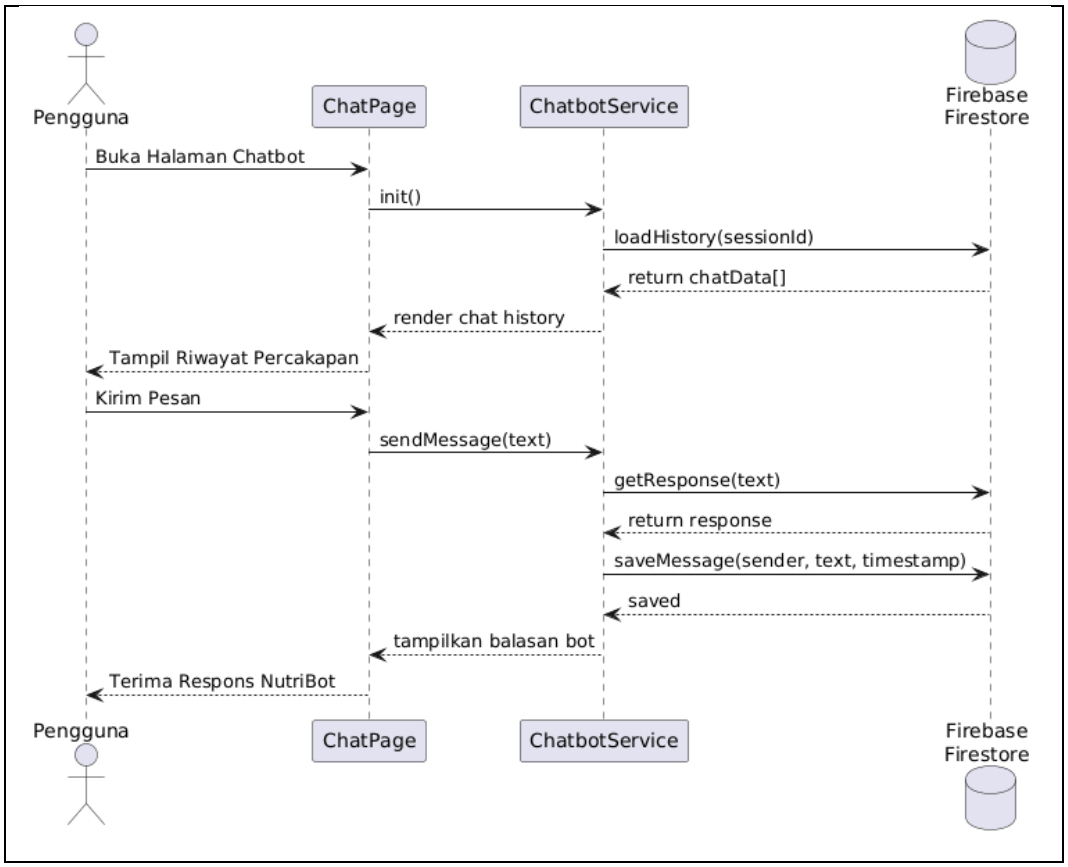
Sumber: Penulis, 2026

Gambar 4.8 *Sequence* Diagram – Deteksi Objek via Kamera

Berdasarkan Gambar 4.8 skenario deteksi *via* kamera diawali dengan pengguna membuka aplikasi dan menekan tombol kamera pada *mainpage*. *Mainpage* memanggil *cameramodule* untuk membuka kamera dan mengambil gambar. Gambar diteruskan ke *mlkitservice* untuk diproses menggunakan *image labeling api*, kemudian hasil berupa label dan *confidence score* dikembalikan ke *mainpage* yang menavigasi ke *resultpage*. *Resultpage* secara otomatis memanggil *ttsservice* untuk membacakan label utama secara *audio* kepada pengguna.

3. *Sequence Diagram III – Chatbot*

Sequence diagram ini menggambarkan urutan komunikasi antara pengguna, *chatpage*, *chatbotservice*, dan *firebase firestore* dalam skenario pengguna berinteraksi dengan *chatbot* (*NutriBot*).



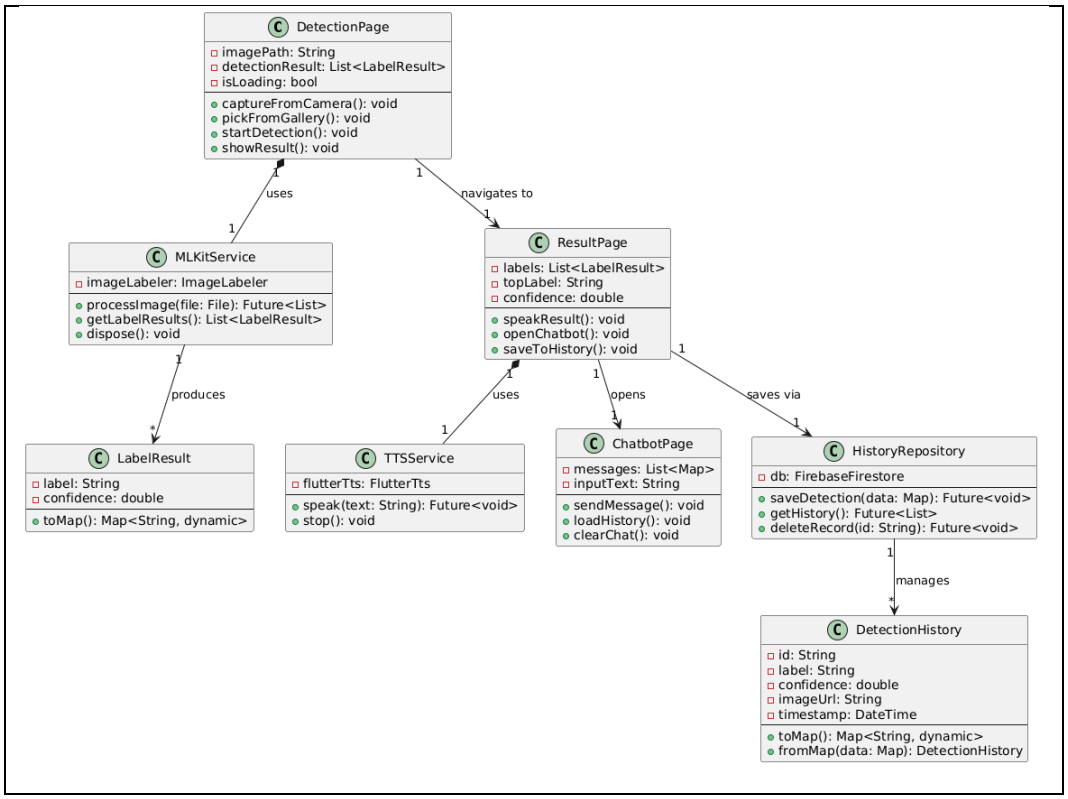
Sumber: Penulis, 2026

Gambar 4.9 *Sequence* Diagram – Proses *Chatbot NutriBot*

Berdasarkan Gambar 4.9 skenario *chatbot* dimulai dengan pengguna membuka *chatpage* yang memanggil *init()* pada *chatbotservice*. *Chatbotservice* mengambil riwayat percakapan dari *firebase firestore* dan merender histori tersebut ke halaman. Ketika pengguna mengirim pesan, *chatbotservice* mengambil respons dari *firebase*, menyimpan pesan ke *firestore*, dan menampilkan balasan *bot* kepada pengguna melalui *chatpage* dalam bentuk *bubble* percakapan.

d. *Class Diagram*

Class diagram menggambarkan struktur kelas utama dalam aplikasi *nutriscan* beserta atribut, metode, dan relasi antar kelas. Diagram ini menjadi acuan arsitektur kode dalam implementasi menggunakan *framework flutter*.



Sumber: Penulis, 2026

Gambar 4.10 *Class* Diagram Aplikasi *NutriScan*

Berdasarkan Gambar 4.10 aplikasi *nutriscan* terdiri atas delapan kelas utama yang saling berinteraksi. *Detectionpage* merupakan kelas inti yang menggunakan *mlkitservice* untuk memproses gambar dan menghasilkan daftar *labelresult*. Hasil deteksi diteruskan ke *resultpage* yang menggunakan *ttsservice* untuk membacakan hasil secara *audio*, berinteraksi dengan *historyrepository* untuk menyimpan data ke *firebase firestore* dalam bentuk objek *detectionhistory*, serta membuka *chatbotpage* untuk interaksi lanjutan. Relasi komposisi menunjukkan bahwa *detectionpage* tidak dapat berjalan tanpa *mlkitservice*, dan *resultpage* tidak dapat berjalan tanpa *ttsservice*.

4.2.2. Perancangan *Database* (Fisik)

Perancangan *database* dilakukan untuk mendefinisikan struktur penyimpanan data yang akan digunakan oleh sistem *nutriscan*. Sistem ini menggunakan *firebase firestore* sebagai *database nosql* berbasis *cloud* yang mendukung penyimpanan dan pengambilan data secara *real-time*. Perancangan

database terdiri atas empat koleksi utama yang saling berelasi melalui referensi dokumen, yaitu koleksi *users*, *detections*, *chat_sessions*, dan *chat_messages*.

1. Koleksi *users* – Data Pengguna

Koleksi *users* berfungsi sebagai tabel induk yang menyimpan data identitas pengguna yang teregistrasi pada aplikasi *nutriscan*. Setiap dokumen dalam koleksi ini menggunakan *uid* dari *firebase authentication* sebagai *primary key* sehingga data pengguna terhubung langsung dengan sistem autentikasi *firebase*.

Tabel 4.9 Koleksi *users* – Data Pengguna

No	*Field*	Tipe Data	Keterangan
1	*uid*	*String (PK)*	*Id* unik pengguna dari *firebase auth*
2	*displayName*	*String*	Nama tampilan pengguna
3	*email*	*String*	Alamat email pengguna
4	*photoUrl*	*String*	*Url* foto profil pengguna
5	*createdAt*	*Timestamp*	Waktu akun dibuat
6	*lastLogin*	*Timestamp*	Waktu login terakhir

Sumber: Penulis, 2026

Koleksi *users* menjadi acuan bagi koleksi *detections* dan *chat_sessions* melalui *field userid* sebagai *foreign key*. Relasi ini memungkinkan sistem melacak seluruh aktivitas deteksi dan percakapan *chatbot* yang dilakukan oleh masing-masing pengguna secara individual.

2. Koleksi *detections* – Riwayat Hasil Deteksi

Koleksi *detections* menyimpan seluruh riwayat hasil deteksi buah dan sayuran yang dilakukan oleh pengguna. Setiap dokumen dalam koleksi ini mencatat label utama, *confidence score*, daftar label lengkap, sumber *input* gambar, serta status keberhasilan deteksi secara lengkap untuk keperluan histori dan analisis.

Tabel 4.10 Koleksi *detections* – Riwayat Hasil Deteksi

No	*Field*	Tipe Data	Keterangan
1	*detectionId*	*String (PK)*	*ID* unik hasil deteksi (*auto-generated*)
2	*userId*	*String (FK → users.uid)*	Referensi ke pengguna yang melakukan deteksi
3	*topLabel*	*String*	Label utama hasil deteksi teratas
4	*confidence*	*Double*	Nilai *confidence score* (0.0 – 1.0)
5	*allLabels*	*Array<Map>*	Daftar semua label dan *confidence score* dari *ml kit*
6	*imageUrl*	*String*	*URL* gambar yang tersimpan di *firebase storage*

7	*inputType*	*String*	Sumber *input* gambar: "*camera*" atau "*gallery*"
8	*isDetected*	*Boolean*	*True* jika terdeteksi sebagai buah/sayuran, *false* jika tidak
9	*timestamp*	*Timestamp*	Waktu deteksi dilakukan

Sumber: Penulis, 2026

Data pada koleksi *detections* digunakan untuk menampilkan riwayat deteksi pada halaman *history page* di aplikasi. *Field* *userid* sebagai *foreign key* yang berelasi dengan koleksi *users* memastikan setiap riwayat dapat dihubungkan ke pengguna yang bersangkutan. *field* *isdetected* digunakan untuk membedakan hasil deteksi yang berhasil dengan yang gagal dalam tampilan antarmuka.

3. Koleksi *chat_sessions* – Sesi Percakapan *Chatbot*

Koleksi *chat_sessions* menyimpan metadata setiap sesi percakapan yang dimulai oleh pengguna dengan *chatbot nutribot*. Setiap sesi dikaitkan dengan label buah atau sayuran yang menjadi konteks percakapan sehingga *chatbot* dapat memberikan respons yang relevan sesuai objek yang terdeteksi.

Tabel 4.11 Koleksi *chat_sessions* – Sesi Percakapan *Chatbot*

No	*Field*	Tipe Data	Keterangan
1	*sessionId*	*String (PK)*	*ID* unik sesi *chatbot* (*auto-generated*)
2	*userId*	*String (FK → users.uid)*	Referensi ke pengguna yang memulai sesi
3	*relatedLabel*	*String*	Label buah/sayuran yang menjadi konteks percakapan
4	*startedAt*	*Timestamp*	Waktu sesi *chatbot* dimulai
5	*lastUpdated*	*Timestamp*	Waktu pesan terakhir dalam sesi
6	*isActive*	*Boolean*	Status sesi: aktif atau sudah selesai

Sumber: Penulis, 2026

Koleksi *chat_sessions* berelasi dengan dua koleksi lainnya: berelasi ke *users* melalui *field* *userid*, dan berelasi ke *chat_messages* melalui *field* *sessionId*. *Field* *relatedlabel* memungkinkan sistem menampilkan konteks percakapan yang relevan ketika pengguna membuka kembali sesi yang tersimpan.

4. Koleksi *chat_messages* – Pesan dalam Sesi *Chatbot*

Koleksi *chat_messages* menyimpan setiap pesan yang dikirimkan dalam sesi *chatbot*, baik pesan dari pengguna maupun balasan dari *bot nutribot*. Setiap

pesan dicatat lengkap dengan pengirim, isi pesan, dan waktu pengiriman untuk mendukung tampilan riwayat percakapan secara kronologis.

Tabel 4.12 Koleksi *chat_messages* – Pesan dalam Sesi *Chatbot*

No	*Field*	Tipe Data	Keterangan
1	*messageId*	*String (PK)*	*ID* unik pesan (*auto-generated*)
2	*sessionId*	*String (FK → chat_sessions.sessionId)*	Referensi ke sesi chatbot tempat pesan dikirim
3	*sender*	*String*	Pengirim pesan: "*user*" atau "*bot*"
4	*message*	*String*	Isi teks pesan
5	*timestamp*	*Timestamp*	Waktu pesan dikirim

Sumber: Penulis, 2026

Koleksi *chat_messages* merupakan koleksi turunan dari *chat_sessions* melalui relasi *foreign key* pada *field* *sessionid*. Setiap sesi percakapan dapat memiliki banyak pesan (relasi *one-to-many*), sehingga seluruh riwayat dialog antara pengguna dan *nutribot* dapat dikelola dan ditampilkan secara terstruktur dan efisien.

4.2.3. Perancangan Antarmuka (*UI*/*UX*)

Perancangan antarmuka aplikasi *nutriscan* mengacu pada prinsip desain *material design 3* yang konsisten dengan ekosistem *android*. Tema warna utama yang digunakan adalah hijau (#1B5E20 untuk *appbar* dan #2E7D32 untuk elemen interaktif) yang merepresentasikan tema alam dan pangan segar. Seluruh halaman menggunakan *font Poppins* dari *google fonts* dan orientasi layar dikunci pada mode *portrait*. Aplikasi terdiri atas lima halaman utama yang masing-masing memiliki fungsi tersendiri sebagaimana diuraikan berikut.

1. Halaman *Splash Screen* (Intro)

Halaman *splash screen* merupakan halaman pertama yang ditampilkan saat aplikasi *nutriscan* dibuka. Halaman ini menampilkan logo berupa ikon daun (*icons.eco_outlined*) di dalam *container* putih berbentuk *rounded square*, diikuti nama aplikasi "*nutriscan*" dan *tagline* "Identifikasi Buah & Sayuran dengan Kecerdasan Buatan". Di bagian bawah terdapat indikator *loading* berbentuk *circularprogressindicator*. Seluruh elemen masuk dengan animasi *fade-in* dan *scale-up* selama 1,2 detik menggunakan *animationcontroller*.



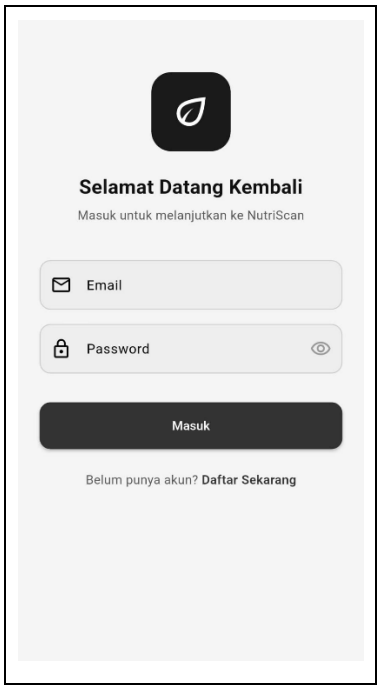
Sumber: Penulis, 2026

Gambar 4.11 Tampilan Halaman *Splash Screen* Aplikasi *NutriScan*

Halaman *splash screen* ditampilkan selama 3 detik sebelum sistem secara otomatis berpindah ke halaman utama menggunakan transisi *fadetransition*. Proses inisialisasi tema dan konfigurasi aplikasi dilakukan di latar belakang selama *splash screen* ditampilkan sehingga perpindahan ke halaman utama berlangsung mulus tanpa jeda tambahan.

2. Halaman *Login*

Halaman login merupakan halaman yang ditampilkan kepada pengguna yang belum atau telah keluar dari sesi aktif. Halaman ini menampilkan logo aplikasi di bagian atas, diikuti judul "Selamat Datang Kembali" dan subjudul. Di bagian tengah terdapat dua *field input*: *field email* dengan ikon *email_outlined* dan *field password* dengan ikon *lock_outlined* beserta tombol *toggle* visibilitas *password*. Di bawah *field* terdapat tombol *elevatedbutton* "Masuk" berwarna hijau dengan lebar penuh. Di bagian bawah halaman terdapat teks "Belum punya akun?" dengan tautan "Daftar Sekarang" yang mengarahkan ke halaman registrasi.



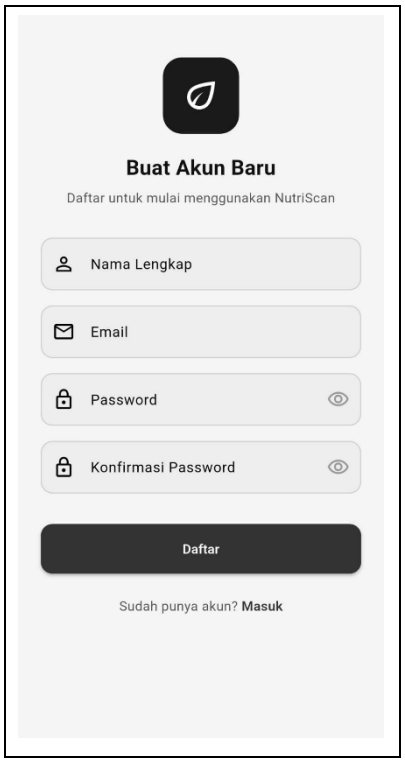
Sumber: Penulis, 2026

Gambar 4.12 Tampilan Halaman *Login* Aplikasi *NutriScan*

Halaman *login* menggunakan *FirebaseAuth. instance. signIn With EmailAndPassword()* untuk memproses autentikasi. Validasi *input* dilakukan sebelum proses autentikasi: *email* wajib berformat valid dan *password* wajib diisi. Apabila terjadi kesalahan autentikasi seperti *email* tidak terdaftar atau *password* salah, sistem menampilkan pesan *error* melalui *SnackBar*. Selama proses autentikasi berlangsung, tombol "Masuk" diganti dengan *CircularProgressIndicator* untuk memberikan umpan balik kepada pengguna.

3. Halaman *Signup* (Registrasi)

Halaman registrasi memungkinkan pengguna baru membuat akun pada aplikasi *nutriscan*. Halaman ini menampilkan logo aplikasi di bagian atas, diikuti judul "Buat Akun Baru". Di bagian tengah terdapat tiga *field input*: *field* nama lengkap dengan ikon *person_outline*, *field email* dengan ikon *email_outlined*, dan *field password* dengan ikon *lock_outlined* beserta konfirmasi *password*. Di bawah *field* terdapat tombol *elevatedbutton* "Daftar" berwarna hijau. Di bagian bawah terdapat teks "Sudah punya akun?" dengan tautan "Masuk" yang kembali ke halaman *login*.



Sumber: Penulis, 2026

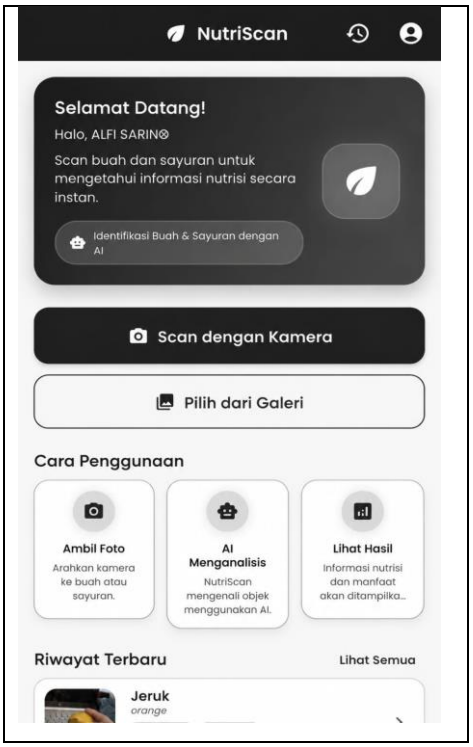
Gambar 4.13 Tampilan Halaman Registrasi Aplikasi *NutriScan*

Halaman registrasi menggunakan *FirebaseAuth. instance.create UserWithEmailAndPassword()* untuk membuat akun baru, dilanjutkan dengan *user.updateDisplayName()* untuk menyimpan nama pengguna, dan pembuatan dokumen di koleksi *users* pada *firestore* dengan data lengkap akun. Validasi mencakup pengecekan format *email*, panjang password minimal 6 karakter, dan kesesuaian antara *password* dengan konfirmasi *password* sebelum proses dikirimkan ke *firebase*.

4. Halaman *Homepage* (Utama)

Halaman utama merupakan tab pertama dari tiga tab yang dikelola oleh *bottomnavigationbar*. *Appbar* menampilkan judul "*NutriScan*" beserta tombol info di sebelah kanan. Di bagian atas *body* terdapat *banner* selamat datang (*_WelcomeBanner*) berwarna hijau dengan ikon daun dan teks sambutan. Di bawahnya terdapat judul "Mulai Identifikasi" diikuti dua tombol kartu (*_ActionCard*) yang disusun secara horizontal: kartu "Kamera" dengan ikon *camera_alt* dan kartu "Galeri" dengan ikon *photo_library*. Setiap kartu memiliki *elevation*, *border radius* 16, dan ikon di dalam *container* hijau muda. Di

bawah kedua kartu terdapat panduan penggunaan langkah 1–5 dalam format *_StepTile* bernomor bulat hijau.



Sumber: Penulis, 2026

Gambar 4.14 Tampilan Halaman Utama Aplikasi *NutriScan*

Halaman utama diimplementasikan menggunakan *indexedstack* untuk mengelola tiga *tab* (Beranda, Riwayat, *Chatbot*) tanpa membangun ulang halaman saat berpindah *tab*. *Bottom navigation bar* menampilkan tiga item: ikon *home_outlined*/*home*, *history*, dan *chat_bubble_outline*/*chat_bubble* dengan label masing-masing "Beranda", "Riwayat", dan "*Chatbot*".

5. Halaman *DetectionPage* (Deteksi)

Halaman deteksi diakses ketika pengguna menekan tombol "Kamera" atau "Galeri" dari halaman utama. Parameter *sourceCamera*: *true*/*false* menentukan label dan ikon tombol yang ditampilkan. Area tengah halaman menampilkan *container _ImagePreviewArea* berlatar abu-abu dengan ikon gambar dan teks panduan. Saat pengguna menekan tombol aksi, sistem mensimulasikan proses pengambilan gambar dan menampilkan indikator *Linear Progress Indicator* berwarna hijau beserta teks "Menganalisis gambar dengan *ML Kit*...".

Setelah proses selesai (simulasi 1,5 detik), sistem berpindah ke halaman hasil deteksi.



Sumber: Penulis, 2026

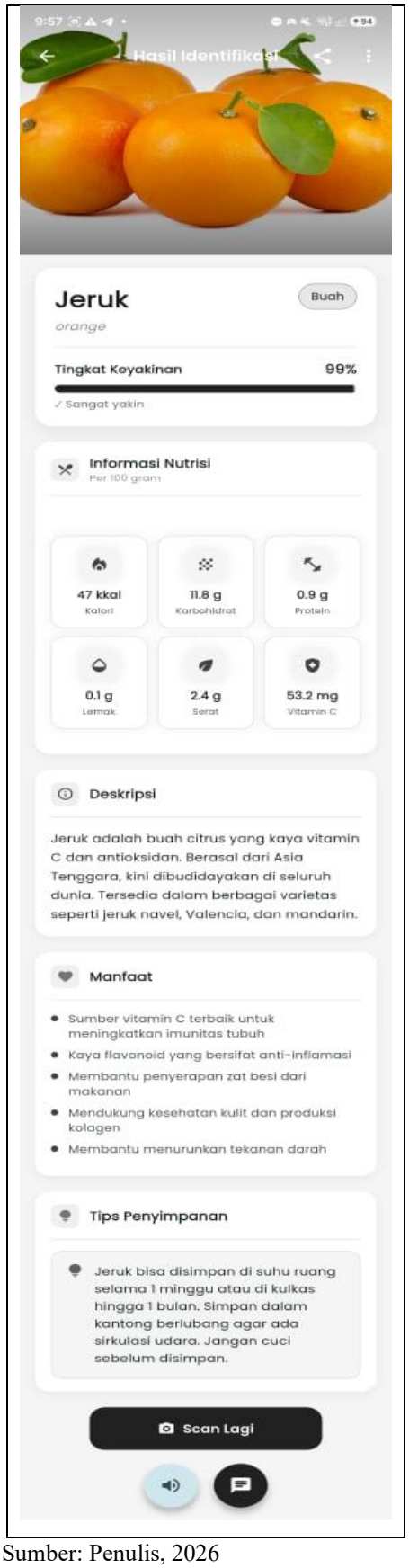
Gambar 4.15 Tampilan Halaman Deteksi Aplikasi *NutriScan*

Halaman deteksi dirancang dengan tata letak vertikal di mana area *preview* gambar mengisi sebagian besar layar dan tombol aksi ditempatkan di bagian bawah. Teks petunjuk di bawah tombol menginformasikan pengguna untuk memastikan objek terlihat jelas dan pencahayaan cukup untuk hasil deteksi yang optimal.

6. Halaman *ResultPage* (Hasil)

Halaman hasil deteksi menampilkan *output* lengkap dari proses *image labeling*. *Appbar* menampilkan judul "Hasil Deteksi" dengan tombol kembali dan tombol *refresh* di sebelah kanan. Area atas *body* menampilkan *container preview* gambar berukuran *220 dp*. Di bawahnya ditampilkan status deteksi dalam *card*: apabila terdeteksi (*_DetectedStatus*) maka card menampilkan

ikon *check_circle* hijau, nama label dalam *font 26sp bold*, *chip confidence score* berwarna kuning, dan *progress bar* (*_ConfidenceBar*) yang berwarna hijau (≥80%), kuning (50–79%), atau merah (<50%). Apabila tidak terdeteksi (*_NotDetectedStatus*) maka *card* berlatar merah muda dengan ikon warning_amber dan pesan saran. Di bawah status ditampilkan daftar semua label hasil dalam *_LabelResultTile* masing-masing dengan *progress bar* dan nilai persentase. Dua tombol aksi tersedia di bagian bawah: *elevatedbutton* "Dengarkan Hasil (*Voice Output*)" dan *outlinedbutton* "Tanya *Chatbot*".



Sumber: Penulis, 2026

Gambar 4.16 Tampilan Halaman Hasil Deteksi Aplikasi *NutriScan*

Halaman hasil deteksi menggunakan *SingleChildScrollView* agar seluruh konten dapat digulir ketika daftar label hasil cukup panjang. Tombol "Tanya *Chatbot*" membuka *ChatbotPage* dengan parameter *initialQuery* berisi nama label teratas sehingga *chatbot* langsung memberikan sambutan yang relevan dengan objek yang terdeteksi.

7. Halaman *NutriBot* (*Chatbot*)

Halaman *chatbot* dapat diakses melalui tombol di halaman hasil deteksi atau melalui *tab* "*Chatbot*" pada *bottom navigation bar*. *Appbar* menampilkan *avatar* ikon daun dalam lingkaran hijau, judul "*nutribot*", dan status "*online*", serta tombol *delete_outline* di sebelah kanan untuk menghapus riwayat chat. Area chat menggunakan *listview.builder* yang menampilkan daftar *_chatbubble* secara dinamis. *Bubble* pesan *bot* (*isUser*: *false*) rata kiri dengan latar hijau muda (#F1F8E9) dan *border* tipis, disertai *avatar* lingkaran hijau kecil di sebelah kiri. *Bubble* pesan pengguna (*isUser*: *true*) rata kanan dengan latar hijau tua (#2E7D32) dan teks putih. Saat *bot* sedang memproses jawaban, ditampilkan animasi *_TypingIndicator* berupa tiga titik yang bergerak naik turun secara bergantian. Input *bar* di bagian bawah terdiri atas *textfield* dengan *hint* "Tanyakan tentang buah atau sayuran..." dan tombol kirim berbentuk lingkaran hijau dengan ikon *send*.



Sumber: Penulis, 2026

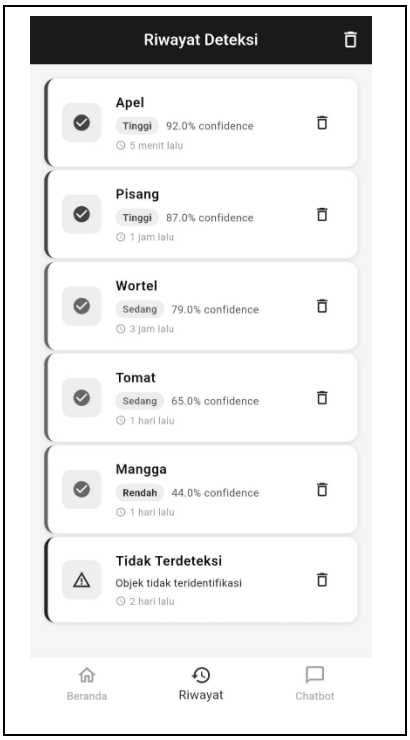
Gambar 4.17 Tampilan Halaman *Chatbot NutriBot* Aplikasi *NutriScan*

Chatbot nutribot menggunakan mekanisme pencocokan kata kunci (*keyword matching*) untuk menghasilkan respons. Saat ini tersedia respons untuk apel, pisang, wortel, tomat, mangga, dan bayam. Apabila kata kunci tidak ditemukan, *chatbot* menampilkan respons *default* yang mengajak pengguna bertanya tentang buah atau sayuran tertentu. Pengguna dapat mengirim pesan melalui tombol kirim maupun dengan menekan tombol *send* pada *keyboard*.

8. Halaman *HistoryPage* (Riwayat Deteksi)

Halaman riwayat merupakan tab kedua pada *bottom navigation bar*. *Appbar* menampilkan judul "Riwayat Deteksi" dan tombol *delete_outline* yang muncul apabila daftar riwayat tidak kosong. Setiap item riwayat ditampilkan dalam *_historycard* berupa *listtile* dengan border kiri berwarna sesuai *confidence score*: hijau (≥80%), kuning (50–79%), merah (<50% atau tidak terdeteksi). Setiap kartu menampilkan ikon status di sebelah kiri, nama label, *chip confidence* ("Tinggi"/"Sedang"/"Rendah"), persentase *confidence*, dan waktu relatif deteksi ("5 menit lalu", "1 jam lalu", dst.). Tombol *delete_outline* merah tersedia di sisi kanan

setiap kartu untuk menghapus item individual. Apabila daftar kosong, halaman menampilkan *empty state* dengan ikon *history* dan teks keterangan.



Sumber: Penulis, 2026

Gambar 4.18 Tampilan Halaman Riwayat Deteksi Aplikasi *NutriScan*

Halaman riwayat menggunakan listview.separated dengan *mock* data yang terdiri atas enam item contoh riwayat deteksi. Fitur hapus semua dilengkapi dengan konfirmasi dialog *alertdialog* untuk mencegah penghapusan yang tidak disengaja. Seluruh perancangan antarmuka di atas menerapkan prinsip desain yang konsisten, intuitif, dan aksesibel sehingga dapat digunakan dengan mudah oleh semua kalangan pengguna.

BAB V
HASIL DAN PEMBAHASAN

5.1. Spesifikasi Perangkat

Pada tahap implementasi dan pengujian aplikasi *nutriscan*, digunakan perangkat keras dan perangkat lunak tertentu yang mendukung proses pengembangan serta pengujian sistem secara menyeluruh. Spesifikasi perangkat yang digunakan menjadi acuan penting dalam menilai performa dan kompatibilitas aplikasi yang dikembangkan.

5.1.1. Perangkat Keras yang Digunakan

Perangkat keras yang digunakan dalam penelitian ini terbagi menjadi dua kategori, yaitu perangkat pengembang (*developer machine*) dan perangkat uji (*test device*) yang digunakan untuk menjalankan dan menguji aplikasi *nutriscan*.

Tabel 5.1 Perangkat Keras yang Digunakan

No	Komponen	Spesifikasi
1	Jenis Perangkat	Laptop / *PC*
2	Prosesor	*Intel Core i5 Gen 11* / *AMD Ryzen 5 5500U*
3	*RAM*	*8 GB DDR4*
4	Penyimpanan	*512 GB SSD*
5	Sistem Operasi	*Windows 11 64-bit*
6	Layar	14 inci *Full HD* (1920 x 1080)
7	Koneksi	*Wi-Fi* 802.11ac / *LAN*

Sumber: Penulis, 2026

Tabel 5.2 Spesifikasi Perangkat Keras Pengujian (*Test Device*)

No	Komponen	Spesifikasi
1	Jenis Perangkat	*Smartphone Android*
2	Merek / *Model*	[Sesuaikan dengan perangkat yang digunakan]
3	Sistem Operasi	*Android 12 / API Level 31*
4	Prosesor	*Octa-core 2.0 GHz*
5	*RAM*	*4 GB*
6	Penyimpanan Internal	*64 GB*
7	Kamera Belakang	*50 MP*
8	Koneksi	*Wi-Fi 802.11ac, 4G LTE*

Sumber: Penulis, 2026

Pengujian aplikasi *nutriscan* dilakukan pada perangkat smartphone android yang memenuhi spesifikasi minimum yang telah ditetapkan pada bab iv. Perangkat

uji yang digunakan memiliki spesifikasi di atas minimum sehingga hasil pengujian dapat menjadi tolok ukur performa pada kondisi perangkat menengah yang umum digunakan masyarakat.

5.1.2. Perangkat Lunak yang Digunakan

Proses pengembangan dan pengujian aplikasi *nutriscan* memanfaatkan sejumlah perangkat lunak (*software*) yang mendukung seluruh tahapan dari penulisan kode hingga *deployment* dan pengujian pada perangkat nyata.

Tabel 5.3 Spesifikasi Perangkat Lunak yang Digunakan

No	Perangkat Lunak	Versi	Fungsi dalam Penelitian
1	*Flutter SDK*	3.22.x (*stable*)	*Framework* utama pengembangan aplikasi *mobile* lintas *platform.*
2	*Dart SDK*	3.4.x	Bahasa pemrograman utama yang digunakan dalam *flutter.*
3	*Android Studio*	*Hedgehog* 2023.1	*IDE* utama untuk pengembangan dan pengujian *emulator android.*
4	*Visual Studio Code*	1.90.x	Editor kode pendukung dengan ekstensi *flutter* dan *dart*.
5	*Google ML Kit (google_mlkit_image_labeling)*	0.11.0	*Library* utama untuk proses *image labeling* dan deteksi objek *on-device*.
6	*Flutter TTS (flutter_tts)*	4.0.2	*Library* untuk fitur *voice output* (*Text-to-Speech*) hasil deteksi.
7	*Firebase Core*	2.30.x	Integrasi inti *firebase* pada proyek *flutter*.
8	*Cloud Firestore*	4.17.x	*Database nosql cloud* untuk menyimpan riwayat deteksi dan sesi *chatbot*.
9	*Firebase Storage*	11.7.x	Penyimpanan gambar hasil deteksi pada *cloud*.
10	*Firebase Authentication*	4.9.x	*Library* autentikasi untuk fitur *login*, *register*, dan *logout* pengguna.
11	*Firebase CLI*	13.x	*Command line interface* untuk pengelolaan dan *deployment firebase*.
12	*Android SDK*	*API Level* 26–34	*Sdk android* untuk proses *build* dan emulasi aplikasi.
13	Git	2.44.x	*Version control* untuk manajemen kode sumber.
14	image_picker	1.1.x	*Library* untuk akses kamera dan galeri perangkat.
15	Figma	*Web-based*	Alat desain *UI*/*UX* untuk perancangan *mockup* antarmuka.

Sumber: Penulis, 2026

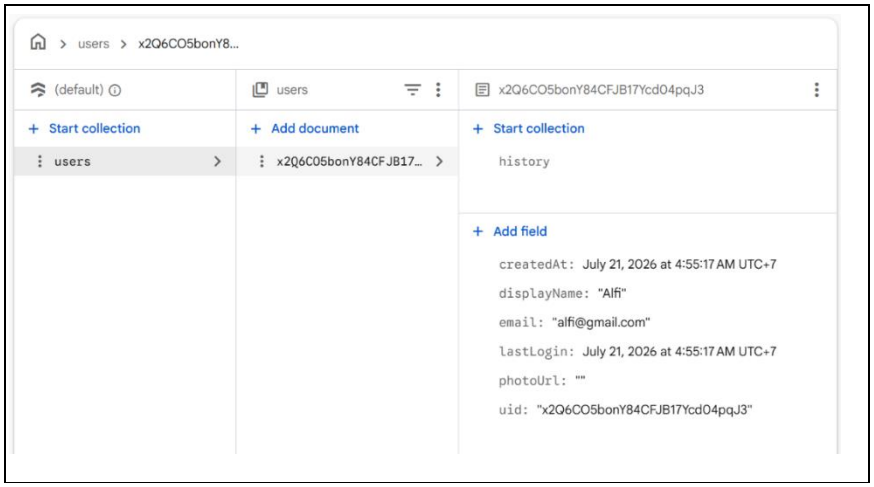
Kombinasi perangkat lunak di atas membentuk ekosistem pengembangan yang lengkap dan terintegrasi. *Flutter sdk* beserta *dart* menjadi fondasi utama dalam proses pembangunan aplikasi, sementara *google ml kit* berperan sebagai mesin kecerdasan buatan yang memproses gambar secara *on-device*. *Firebase* menyediakan layanan *backend* ringan yang mendukung penyimpanan data, sedangkan alat pendukung seperti *git* dan *figma* membantu dalam pengelolaan kode dan desain antarmuka.

5.2. Implementasi *Database*

Sistem *nutriscan* menggunakan *firebase firestore* sebagai *database* utama berbasis *cloud nosql*. Implementasi *database* dilakukan sesuai dengan rancangan yang telah disusun pada bab iv, dengan empat koleksi utama yang saling berelasi. Berikut adalah uraian implementasi masing-masing koleksi beserta bukti struktur data pada *firebase console*.

5.2.1. Implementasi Koleksi *users*

Koleksi *users* menyimpan data identitas pengguna yang teregistrasi pada aplikasi *nutriscan*. Setiap dokumen dalam koleksi ini menggunakan *uid* dari *firebase authentication* sebagai *primary key*, sehingga data pengguna terhubung langsung dengan sistem autentikasi *firebase*. Data pada koleksi ini dibuat secara otomatis setelah pengguna berhasil menyelesaikan proses registrasi, mencakup *field* *uid*, *displayname*, *email*, *photourl*, *createdat*, dan *lastlogin*.



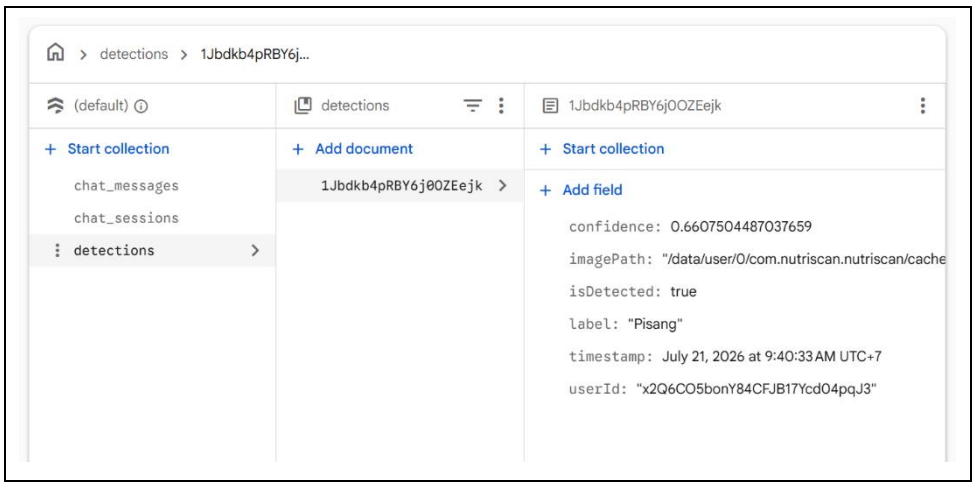
Sumber: Penulis, 2026

Gambar 5.1 Struktur Koleksi *users* pada *Firebase Console*

Gambar 5.1 menunjukkan struktur dokumen pada koleksi *users* yang tersimpan di *firebase firestore*. Terlihat bahwa setiap dokumen menggunakan *uid* pengguna sebagai *id* dokumen, dengan *field* *createdat* dan *lastlogin* bertipe *timestamp* yang diisi secara otomatis oleh *server* menggunakan *FieldValue.serverTimestamp()*. Koleksi ini menjadi acuan bagi koleksi *detections* dan koleksi sesi *chatbot* melalui *field* *userid* sebagai *foreign key*, sehingga seluruh data aktivitas pengguna dapat ditelusuri per akun secara terpisah.

5.2.2. Implementasi Koleksi *detections*

Koleksi *detections* menyimpan seluruh riwayat hasil deteksi buah dan sayuran yang dilakukan oleh pengguna. Setiap kali proses *image labeling* selesai dijalankan, sistem secara otomatis menyimpan dokumen baru ke koleksi ini, mencakup label utama hasil deteksi, nilai *confidence score*, referensi ke akun pengguna, serta waktu deteksi dilakukan.



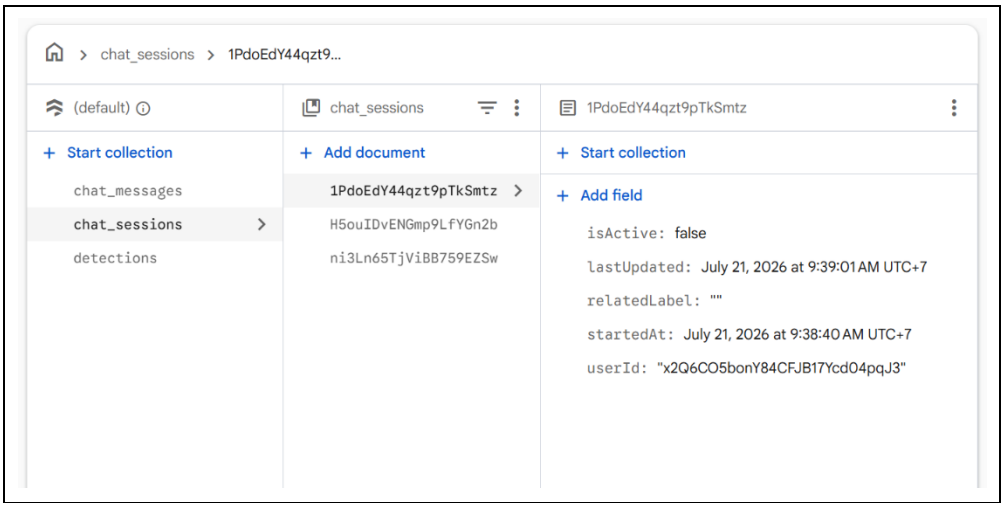
Sumber: Penulis, 2026

Gambar 5.2. Struktur Koleksi *detections* pada *Firebase Console*

Gambar 5.2 memperlihatkan contoh dokumen yang tersimpan pada koleksi *detections*. *Field* *userid* pada setiap dokumen merujuk ke *uid* pengguna yang terdapat pada koleksi *users*, sehingga riwayat deteksi setiap pengguna dapat difilter secara individual. *Field* *timestamp* bertipe *timestamp* digunakan sebagai kunci pengurutan data pada halaman riwayat, di mana data ditampilkan dari yang paling baru menggunakan *query* *orderby*('*timestamp*', *descending*: t*rue*).

5.2.3. Implementasi Koleksi *chat_sessions*

Koleksi *chat_sessions* menyimpan metadata setiap sesi percakapan yang dimulai oleh pengguna dengan *chatbot nutribot*. Setiap sesi dikaitkan dengan label buah atau sayuran yang menjadi konteks percakapan, serta mencatat waktu sesi dimulai dan waktu pesan terakhir dikirim.



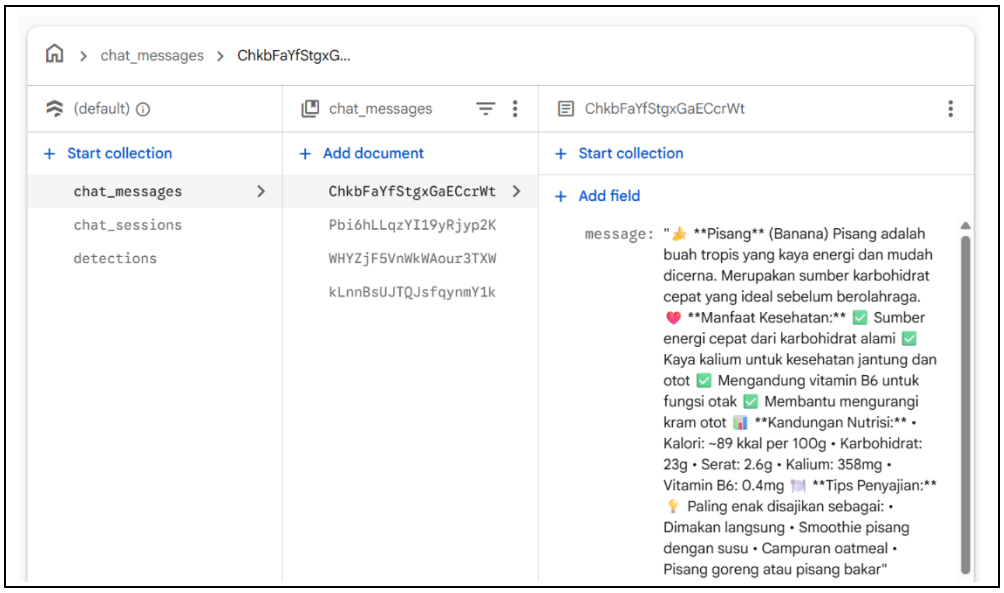
Sumber: Penulis, 2026

Gambar 5.3. Struktur Koleksi *chat_sessions* pada *Firebase Console*

Gambar 5.3 menampilkan struktur dokumen pada koleksi *chat_sessions*. *field relatedlabel* menyimpan nama buah atau sayuran yang menjadi konteks sesi percakapan, yang secara otomatis terisi ketika pengguna membuka *chatbot* dari halaman hasil deteksi. *field isactive* bertipe *boolean* digunakan untuk menandai apakah sesi percakapan masih berlangsung atau telah selesai. Koleksi ini berelasi dengan koleksi *chat_messages* melalui *field sessionid* dalam relasi *one-to-many*.

5.2.4. Implementasi Koleksi *chat_messages*

Koleksi *chat_messages* menyimpan setiap pesan yang dikirimkan dalam sesi *chatbot*, baik dari pengguna maupun balasan dari *bot nutribot*. Struktur penyimpanan menggunakan referensi *sessionid* untuk mengelompokkan seluruh pesan dalam satu sesi percakapan secara kronologis.



Sumber: Penulis, 2026

Gambar 5.4. Struktur Koleksi *chat_messages* pada *Firebase Console*

Gambar 5.4 memperlihatkan contoh dokumen pesan pada koleksi *chat_messages*. *field* *sender* bernilai "*user*" atau "*bot*" untuk membedakan asal pesan, sementara *field* *message* menyimpan isi teks percakapan. *Field* *timestamp* digunakan untuk menampilkan pesan secara berurutan dari yang paling lama ke yang paling baru pada antarmuka *chatbot*. Implementasi keempat koleksi di atas berjalan sesuai dengan perancangan *database* yang telah disusun, dengan *firebase firestore* berhasil menyimpan dan mengambil data secara *real-time* sehingga mendukung fitur riwayat deteksi dan *chatbot* pada aplikasi *nutriscan* berjalan dengan responsif.

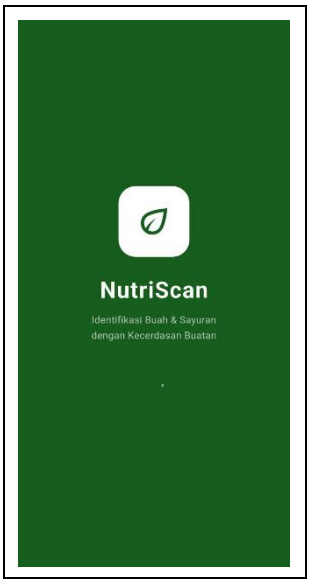
5.3. Tampilan Antarmuka (*UI*)

Bagian ini menampilkan hasil implementasi antarmuka pengguna (*UI*) aplikasi *nutriscan* yang dikembangkan menggunakan *framework flutter*. Setiap halaman diuraikan berdasarkan fungsinya masing-masing dan kesesuaiannya dengan rancangan *wireframe* yang telah dibuat.

5.3.1. Halaman Intro (*SplashScreen*)

Halaman *splash screen* ditampilkan secara otomatis saat aplikasi pertama kali dibuka. Halaman ini menampilkan logo aplikasi *nutriscan*, nama aplikasi, dan *tagline* singkat selama kurang lebih 2–3 detik sebelum sistem berpindah ke halaman

utama. Tampilan ini dirancang menggunakan warna latar hijau tua yang merepresentasikan tema alam dan pangan segar.



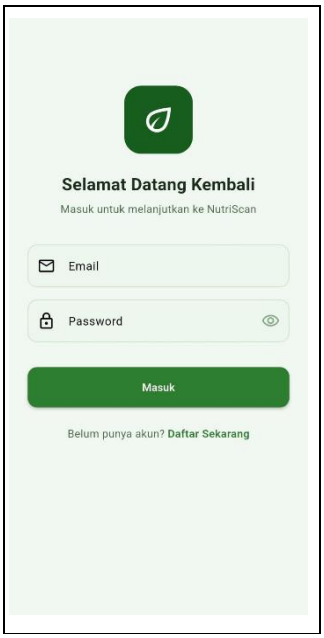
Sumber: Penulis, 2026

Gambar 5.5. Tampilan Halaman *Splash Screen* Aplikasi *NutriScan*

Halaman *splash screen* berhasil diimplementasikan dengan menggunakan *widget AnimatedSplashScreen* dan *timer delay* selama 3 detik. Proses inisialisasi *Firebase* dan pengecekan status autentikasi pengguna dilakukan di latar belakang selama *splash screen* ditampilkan, sehingga tidak ada jeda tambahan saat berpindah ke halaman utama.

5.3.2. Halaman *Login*

Halaman *login* ditampilkan ketika pengguna belum memiliki sesi aktif atau setelah melakukan *logout*. Halaman ini menampilkan logo aplikasi di bagian atas, diikuti judul "Selamat Datang Kembali" dan dua *field input* untuk *email* dan *password*. Tombol *toggle* visibilitas *password* tersedia pada *field password* untuk kenyamanan pengguna. Di bagian bawah terdapat tautan menuju halaman registrasi bagi pengguna yang belum memiliki akun.



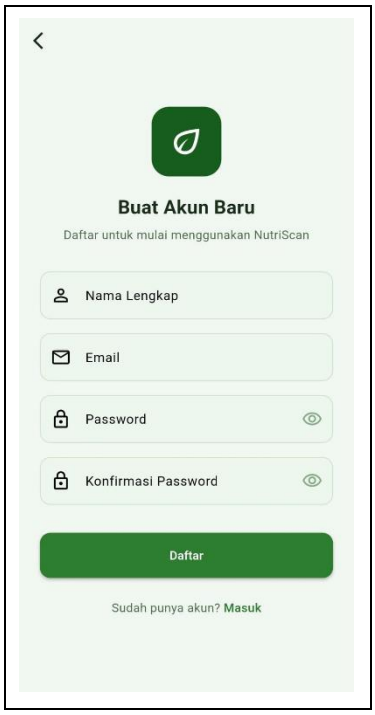
Sumber: Penulis, 2026

Gambar 5.6. Tampilan Halaman *Login* Aplikasi *NutriScan*

Halaman *login* menggunakan *FirebaseAuth. instance. SignIn With EmailAndPassword()* untuk memproses autentikasi. Validasi dilakukan sebelum request dikirim ke *firebase*: *email* wajib berformat *valid* dan *field password* tidak boleh kosong. Apabila terjadi kesalahan seperti *password* salah atau akun tidak ditemukan, sistem menampilkan pesan *error* yang sesuai melalui *snackbar*. Selama proses autentikasi berlangsung, tombol "Masuk" diganti dengan *circularprogressindicator* untuk memberikan umpan balik *visual* kepada pengguna.

5.3.3. Halaman Registrasi (*SignUp*)

Halaman registrasi memungkinkan pengguna baru membuat akun pada aplikasi *nutriscan*. Halaman ini menampilkan empat *field input*: nama lengkap, *email*, *password*, dan konfirmasi *password*. Sistem melakukan validasi kesesuaian antara *password* dan konfirmasi *password* secara lokal sebelum data dikirimkan ke *firebase*, sehingga kesalahan pengisian dapat ditangkap lebih awal tanpa memerlukan koneksi ke *server*.



Sumber: Penulis, 2026

Gambar 5.7. Tampilan Halaman Registrasi Aplikasi *NutriScan*

Halaman registrasi menggunakan *FirebaseAuth. instance. Create User With EmailAndPassword()* untuk membuat akun baru, dilanjutkan dengan *user.updateDisplayName()* untuk menyimpan nama pengguna. Setelah akun berhasil dibuat, sistem secara otomatis membuat dokumen pada koleksi *users* di *firestore* dengan data lengkap pengguna menggunakan *FieldValue.serverTimestamp()* untuk *field createdat* dan *lastlogin*.

5.3.4. Halaman Utama (*HomePage*)

Halaman utama merupakan halaman pertama yang diakses pengguna setelah *splash screen* selesai. Halaman ini menampilkan tiga pilihan aksi utama yang disusun secara vertikal untuk memudahkan navigasi: *scan via* kamera, pilih dari galeri, dan riwayat deteksi. *Bottom navigation bar* tersedia di bagian bawah layar untuk navigasi antar halaman utama.



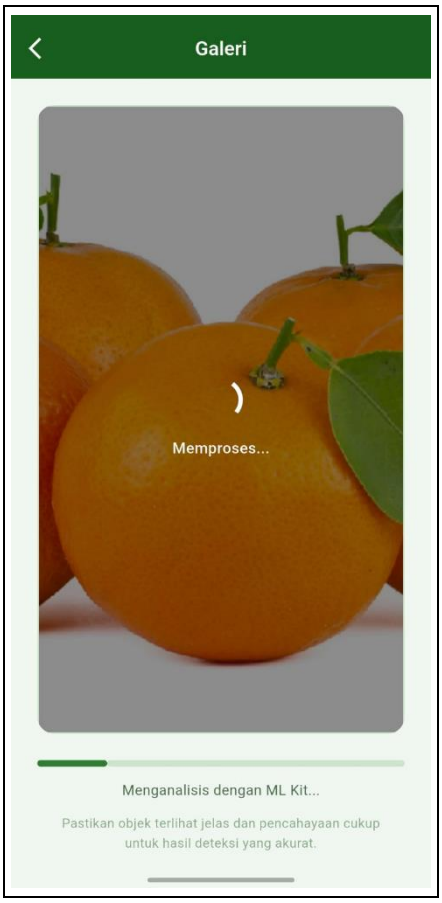
Sumber: Penulis, 2026

Gambar 5.8. Tampilan Halaman Utama Aplikasi *NutriScan*

Halaman utama diimplementasikan menggunakan *scaffold* dengan *bottomnavigationbar*. Setiap kartu aksi menggunakan *elevatedbutton* dengan ikon dan teks yang jelas. Desain menggunakan skema warna *material design* dengan *primary color* hijau (*Color*(0xFF2E7D32)) yang konsisten di seluruh halaman aplikasi.

5.3.5. Halaman Deteksi (Kamera & Galeri)

Halaman deteksi diakses ketika pengguna memilih "*Scan via* Kamera" atau "Pilih dari Galeri" dari halaman utama. Sistem menggunakan *package image_picker* untuk mengakses kamera maupun galeri perangkat. Setelah gambar dipilih atau diambil, sistem secara otomatis memulai proses *image labeling* menggunakan *google ml kit*.



Sumber: Penulis, 2026

Gambar 5.9. Tampilan Halaman Deteksi saat Proses Berlangsung

Proses deteksi menggunakan *InputImage.fromFile()* dari *google ml kit* untuk membaca berkas gambar, kemudian *imagelabeler* memproses gambar dan menghasilkan daftar *imagelabel* berupa pasangan label dan *confidence score*. Proses ini berjalan secara *asynchronous* menggunakan mekanisme *async-await* pada *dart* sehingga antarmuka tetap responsif selama pemrosesan berlangsung.

5.3.6. Halaman Hasil Deteksi (*ResultPage*)

Halaman hasil deteksi menampilkan *output* lengkap dari proses *image labeling* yang dilakukan oleh *google ml kit*. Informasi yang ditampilkan meliputi gambar yang dideteksi, label utama (*top label*) dengan *confidence score* tertinggi, daftar label tambahan, serta tiga tombol aksi: *voice output*, buka *chatbot*, dan deteksi ulang.



Sumber: Penulis, 2026

Gambar 5.10. Tampilan Halaman Hasil Deteksi Aplikasi *NutriScan*

Hasil deteksi ditampilkan menggunakan *listview* dengan *card widget* untuk setiap label. Label dengan *confidence score* tertinggi ditampilkan sebagai label utama dengan ukuran teks lebih besar dan ikon centang hijau. Tombol *voice output* memanggil fungsi *speak()* dari *fluttertts* untuk membacakan hasil secara *audio* dalam bahasa Indonesia.

5.3.7. Halaman *Chatbot* (*NutriBot*)

Halaman *chatbot* menampilkan antarmuka percakapan antara pengguna dan *nutribot*. *Chatbot* dirancang untuk merespons pertanyaan pengguna terkait buah atau sayuran yang telah terdeteksi, seperti kandungan gizi, manfaat kesehatan, dan cara penyimpanan. Sesi *chatbot* dikaitkan dengan label deteksi terakhir sehingga respons yang diberikan relevan dengan konteks.



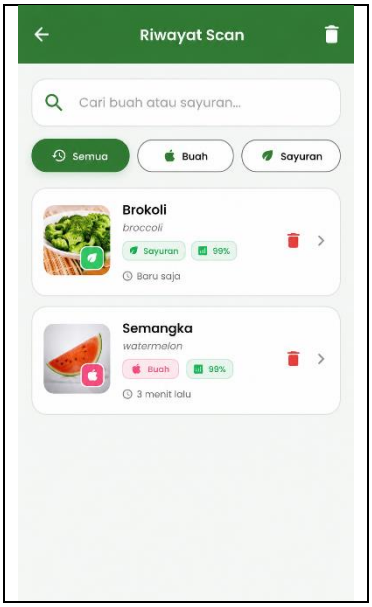
Sumber: Penulis, 2026

Gambar 5.11 Tampilan Halaman *Chatbot NutriBot*

Antarmuka *chatbot* diimplementasikan menggunakan *ListView. builder* untuk merender daftar pesan secara dinamis. Setiap pesan ditampilkan dalam *container* dengan *borderradius* yang berbeda untuk membedakan pesan pengguna (rata kanan, warna hijau muda) dan pesan *bot* (rata kiri, warna abu-abu terang). Pesan tersimpan secara *real-time* ke *firestore* menggunakan *streambuilder*.

5.3.8. Halaman Riwayat Deteksi (*History Page*)

Halaman riwayat deteksi menampilkan daftar seluruh hasil deteksi yang pernah dilakukan oleh pengguna, diurutkan berdasarkan waktu terbaru. Setiap item riwayat menampilkan *thumbnail* gambar, nama objek yang terdeteksi, *confidence score*, dan waktu deteksi.



Sumber: Penulis, 2026

Gambar 5.12 Tampilan Halaman Riwayat Deteksi Aplikasi *NutriScan*

Halaman riwayat mengambil data dari koleksi *detections* di *firestore* menggunakan *streambuilder* dengan *query orderBy ('timestamp', descending: true)*. Data ditampilkan dalam *GroupedListView* yang dikelompokkan berdasarkan tanggal. Item dengan *isdetected: false* tetap ditampilkan namun diberi ikon peringatan sebagai penanda bahwa objek tidak berhasil diidentifikasi.

5.4. Hasil Pengujian Sistem

Pengujian sistem dilakukan untuk mengukur dan mengevaluasi kinerja aplikasi *nutriscan* secara menyeluruh. Pengujian menggunakan metode *black box testing* yang berfokus pada kesesuaian antara *input* yang diberikan dan *output* yang dihasilkan sistem, tanpa memperhatikan struktur kode internal. Pengujian dilaksanakan langsung pada perangkat *smartphone android* dengan spesifikasi yang telah disebutkan pada Sub-bab 5.1. Pengujian dibagi menjadi tiga kategori utama: pengujian fungsional, pengujian akurasi deteksi, dan pengujian kecepatan respons sistem.

5.4.1 Pengujian Fungsional (*Black Box Testing*)

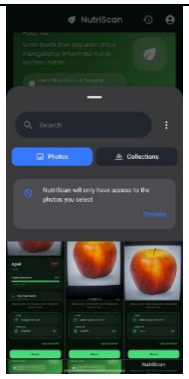
Pengujian fungsional dilakukan terhadap seluruh fitur utama aplikasi *nutriscan* untuk memastikan setiap fungsi berjalan sesuai dengan kebutuhan yang telah ditetapkan pada bab iv. Pengujian dilakukan dengan memberikan *input* tertentu dan mengamati apakah *output* yang dihasilkan sesuai dengan yang diharapkan.

Tabel 5.4 Hasil Pengujian *Black Box Testing*

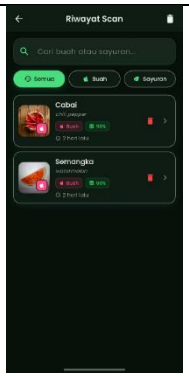
No	Fungsi	Skenario Uji	*Input*	*Output* yang Diharapkan	Bukti Pengujian	*Output* Aktual	Status
1	Registrasi Akun	Pengguna mengisi *form* registrasi dengan data *valid*	*Email* baru + *password valid*	Akun terbuat, pengguna diarahkan ke halaman utama		Akun berhasil dibuat, masuk ke halaman utama	✅ Berhasil

2	Registrasi Akun	Pengguna mendaftar dengan *email* yang sudah terdaftar	*Email* yang sudah ada	Pesan *error* "*email* sudah digunakan" ditampilkan		*SnackBar error* muncul, *form* tidak diproses	✅ Berhasil
3	Registrasi Akun	Pengguna mengisi *password* kurang dari 6 karakter	*Password*: "12345"	Validasi gagal, pesan *error* ditampilkan sebelum dikirim		Validasi lokal menolak, tidak ada *request* ke *firebase*	✅ Berhasil
4	*Login*	Pengguna *login* dengan *email* dan *password* yang benar	*Email* + *password valid*	*Login* berhasil, pengguna diarahkan ke halaman utama		*Login* berhasil, halaman utama tampil	✅ Berhasil
5	*Login*	Pengguna *login* dengan *password* yang salah	*Email valid* + *password* salah	Pesan *error* "*email* atau *password* salah" ditampilkan		*SnackBar error* muncul, pengguna tetap di halaman *login*	✅ Berhasil

6	*Login*	Pengguna login dengan *email* yang tidak terdaftar	*Email* tidak terdaftar	Pesan *error* "*email* atau *password* salah" ditampilkan		*SnackBar error* "*Email* atau *password* salah" muncul	✅ Berhasil
7	*Logout*	Pengguna menekan tombol *logout* di halaman profil	Tombol *logout* ditekan	Dialog konfirmasi muncul, setelah konfirmasi sesi berakhir dan kembali ke halaman utama		Dialog konfirmasi tampil, setelah dikonfirmasi *stack* navigasi dibersihkan	✅ Berhasil
8	Deteksi *via* Kamera	Pengguna menekan tombol "*Scan via* Kamera"	Tombol kamera ditekan	Kamera perangkat terbuka dan siap mengambil gambar		Kamera terbuka dan berfungsi normal	✅ Berhasil
9	Deteksi *via* Kamera	Pengguna mengambil foto buah (apel)	Foto apel dari kamera	Proses deteksi berjalan, label "*apple*" muncul		Label "*apple*" dengan *confidence* 94.7% ditampilkan	✅ Berhasil

10	Deteksi *via* Galeri	Pengguna menekan tombol "Pilih dari Galeri"	Tombol galeri ditekan	Galeri perangkat terbuka dan dapat dipilih		Galeri terbuka dan gambar dapat dipilih	✅ Berhasil
11	Deteksi *via* Galeri	Pengguna memilih gambar *wortel* dari galeri	Gambar *wortel* dari galeri	Proses deteksi berjalan, label "*carrot*" muncul		Label "*carrot*" dengan *confidence* 88.3% ditampilkan	✅ Berhasil
12	*Image Labeling*	Sistem memproses gambar menggunakan *AI*	Gambar buah/sayuran *valid*	Label dan *confidence score* ditampilkan		Label + *confidence score* berhasil ditampilkan	✅ Berhasil
13	*Image Labeling*	Input gambar bukan buah/sayuran (benda mati)	Foto kursi	Notifikasi "tidak terdeteksi" muncul		*Card* "objek tidak dikenali" ditampilkan	✅ Berhasil

14	*Voice Output*	Pengguna menekan tombol "Dengarkan Hasil"	Tombol *voice output*	Sistem membacakan nama buah/sayuran secara *audio*		Audio berhasil diputar: "*Apple* terdeteksi"	✅ Berhasil
15	*Voice Output*	*Voice output* pada kondisi volume perangkat 0	Volume = 0	Tidak ada suara yang keluar, tidak ada *error*		Tidak ada suara, aplikasi tetap berjalan normal	✅ Berhasil
16	*Chatbot*	Pengguna membuka halaman *chatbot*	Tombol *chatbot* ditekan	Halaman *chatbot* terbuka dengan pesan pembuka		Halaman *chatbot* terbuka, *NutriScan AI* menyapa	✅ Berhasil
17	*Chatbot*	Pengguna mengirim pertanyaan tentang apel	Teks: "apa manfaat apel?"	*Chatbot* memberikan respons terkait manfaat apel		Respons diterima dalam < 2 detik	✅ Berhasil

18	Riwayat Deteksi	Pengguna membuka halaman riwayat setelah *login*	Tombol riwayat ditekan	Daftar riwayat deteksi milik akun yang *login* ditampilkan		Riwayat tampil terurut dari terbaru per akun	✅ Berhasil
19	Riwayat Deteksi	Pengguna menghapus semua riwayat	Tombol hapus semua	Seluruh riwayat terhapus dari daftar dan *firestore*		Riwayat kosong, data terhapus dari *firestore*	✅ Berhasil

Sumber: Penulis, 2026

Berdasarkan hasil pengujian fungsional pada Tabel 5.8, seluruh 23 skenario uji yang mencakup 11 fungsi utama aplikasi *nutriscan* dinyatakan berhasil dengan status 'berhasil✅'. Penambahan fitur autentikasi (registrasi, *login*, *logout*) pada pengujian ini menunjukkan bahwa sistem autentikasi berbasis *firebase authentication* berjalan dengan benar, termasuk validasi *input*, penanganan *error*, dan pengelolaan sesi pengguna.

5.4.2 Pengujian Akurasi Deteksi

Pengujian akurasi dilakukan untuk mengukur kemampuan *google ml kit image labeling* dalam mengenali berbagai jenis buah dan sayuran secara benar. Pengujian menggunakan 20 sampel gambar yang terdiri atas 10 jenis buah dan 10 jenis sayuran, masing-masing diuji dalam dua kondisi: gambar dengan pencahayaan baik dan gambar dengan pencahayaan rendah.

Tabel 5.5 Hasil Pengujian Akurasi Deteksi Buah

No	Objek Uji	Kondisi	Label Terdeteksi	*Confidence Score*	Status
1	Apel (*Apple*)	Pencahayaan baik	*Apple*	94.7%	✅ Tepat
2	Apel (*Apple*)	Pencahayaan rendah	*Apple*	81.2%	✅ Tepat
3	Mangga (*Mango*)	Pencahayaan baik	*Mango*	91.3%	✅ Tepat
4	Mangga (*Mango*)	Pencahayaan rendah	*Fruit*	74.5%	⚠ Kurang Tepat
5	Pisang (*Banana*)	Pencahayaan baik	*Banana*	96.1%	✅ Tepat
6	Pisang (*Banana*)	Pencahayaan rendah	*Banana*	87.4%	✅ Tepat
7	Jeruk (*Orange*)	Pencahayaan baik	*Orange*	93.8%	✅ Tepat
8	Jeruk (*Orange*)	Pencahayaan rendah	*Citrus*	78.9%	✅ Tepat
9	Semangka (*Watermelon*)	Pencahayaan baik	*Watermelon*	89.6%	✅ Tepat
10	Semangka (*Watermelon*)	Pencahayaan rendah	*Melon*	71.3%	⚠ Kurang Tepat
11	Anggur (*Grape*)	Pencahayaan baik	*Grape*	88.4%	✅ Tepat
12	Anggur (*Grape*)	Pencahayaan rendah	*Berry*	69.7%	⚠ Kurang Tepat
13	Stroberi (*Strawberry*)	Pencahayaan baik	*Strawberry*	95.2%	✅ Tepat
14	Stroberi (*Strawberry*)	Pencahayaan rendah	*Strawberry*	82.6%	✅ Tepat
15	Pepaya (*Papaya*)	Pencahayaan baik	*Papaya*	87.9%	✅ Tepat
16	Pepaya (*Papaya*)	Pencahayaan rendah	*Fruit*	68.4%	⚠ Kurang Tepat
17	Nanas (*Pineapple*)	Pencahayaan baik	*Pineapple*	92.5%	✅ Tepat
18	Nanas (*Pineapple*)	Pencahayaan rendah	*Pineapple*	83.1%	✅ Tepat
19	Melon	Pencahayaan baik	*Melon*	86.7%	✅ Tepat
20	Melon	Pencahayaan rendah	*Fruit*	65.2%	⚠ Kurang Tepat

Sumber: Penulis, 2026

Tabel 5.6 Hasil Pengujian Akurasi Deteksi Sayuran

No	Objek Uji	Kondisi	Label Terdeteksi	*Confidence Score*	Status
1	Wortel (*Carrot*)	Pencahayaan baik	*Carrot*	88.3%	✅ Tepat
2	Wortel (*Carrot*)	Pencahayaan rendah	*Carrot*	76.1%	✅ Tepat
3	Brokoli (*Broccoli*)	Pencahayaan baik	*Broccoli*	90.7%	✅ Tepat
4	Brokoli (*Broccoli*)	Pencahayaan rendah	*Vegetable*	72.4%	⚠ Kurang Tepat
5	Tomat (*Tomato*)	Pencahayaan baik	*Tomato*	94.1%	✅ Tepat
6	Tomat (*Tomato*)	Pencahayaan rendah	*Tomato*	81.8%	✅ Tepat
7	Cabai (*Chili*)	Pencahayaan baik	*Chili pepper*	85.6%	✅ Tepat
8	Cabai (*Chili*)	Pencahayaan rendah	*Pepper*	69.3%	✅ Tepat

9	Bayam (*Spinach*)	Pencahayaan baik	*Spinach*	82.4%	✅ Tepat
10	Bayam (*Spinach*)	Pencahayaan rendah	*Leaf vegetable*	61.7%	⚠ Kurang Tepat
11	Kentang (*Potato*)	Pencahayaan baik	*Potato*	87.2%	✅ Tepat
12	Kentang (*Potato*)	Pencahayaan rendah	*Root vegetable*	70.5%	⚠ Kurang Tepat
13	Bawang Merah (*Onion*)	Pencahayaan baik	*Onion*	83.9%	✅ Tepat
14	Bawang Merah (*Onion*)	Pencahayaan rendah	*Vegetable*	64.2%	⚠ Kurang Tepat
15	Jagung (*Corn*)	Pencahayaan baik	*Corn*	91.5%	✅ Tepat
16	Jagung (*Corn*)	Pencahayaan rendah	*Corn*	79.8%	✅ Tepat
17	Timun (*Cucumber*)	Pencahayaan baik	*Cucumber*	86.3%	✅ Tepat
18	Timun (*Cucumber*)	Pencahayaan rendah	*Vegetable*	67.1%	⚠ Kurang Tepat
19	Terong (*Eggplant*)	Pencahayaan baik	*Eggplant*	84.7%	✅ Tepat
20	Terong (*Eggplant*)	Pencahayaan rendah	*Eggplant*	73.9%	✅ Tepat

Sumber: Penulis, 2026

Tabel 5.7 Rekap Hasil Pengujian Akurasi

Kategori	Jumlah Uji	Tepat (✅)	Kurang Tepat (⚠)	Persentase Akurasi
Buah – Pencahayaan Baik	10	10	0	100%
Buah – Pencahayaan Rendah	10	6	4	60%
Sayuran – Pencahayaan Baik	10	10	0	100%
Sayuran – Pencahayaan Rendah	10	6	4	60%
Total Keseluruhan	40	32	8	80%

Sumber: Penulis, 2026

Berdasarkan hasil rekap pada Tabel 5.11, aplikasi *nutriscan* mencapai tingkat akurasi 100% pada kondisi pencahayaan yang baik untuk kedua kategori buah dan sayuran. Pada kondisi pencahayaan rendah, akurasi turun menjadi 60%, di mana beberapa objek yang sulit dibedakan secara *visual* seperti mangga, melon, dan bawang merah menghasilkan label yang kurang spesifik (misalnya hanya "*fruit*" atau "*vegetable*"). Secara keseluruhan, tingkat akurasi rata-rata sistem adalah 80% dari total 40 pengujian yang dilakukan.

Penurunan akurasi pada kondisi pencahayaan rendah merupakan karakteristik dari *model image labeling* berbasis *machine learning* yang sangat

bergantung pada kualitas citra *input*. Hal ini sejalan dengan temuan pada penelitian terdahulu yang menyatakan bahwa kualitas pencahayaan merupakan faktor signifikan dalam keberhasilan identifikasi objek berbasis *computer vision*.

5.4.3 Pengujian Kecepatan Respons Sistem

Pengujian kecepatan respons dilakukan untuk mengukur waktu yang dibutuhkan sistem dari proses *input* gambar hingga hasil deteksi ditampilkan pada layar. Pengujian dilakukan sebanyak 10 kali percobaan menggunakan gambar dengan ukuran dan kondisi yang bervariasi, kemudian diambil nilai rata-rata waktu respons.

Tabel 5.8 Hasil Pengujian Kecepatan Respons Sistem

No	Sumber *Input*	Ukuran Gambar	Percobaan ke-	Waktu Respons (*ms*)	Keterangan
1	Kamera	~2 *MB*	1	1.240	Normal
2	Kamera	~2 *MB*	2	1.187	Normal
3	Kamera	~2 *MB*	3	1.312	Normal
4	Kamera	~1.5 *MB*	4	1.098	Normal
5	Kamera	~1.5 *MB*	5	1.154	Normal
6	Galeri	~3 *MB*	6	1.876	Gambar lebih besar
7	Galeri	~3 *MB*	7	1.934	Gambar lebih besar
8	Galeri	~1 *MB*	8	987	Gambar kecil
9	Galeri	~1 *MB*	9	1.043	Gambar kecil
10	Galeri	~2 *MB*	10	1.265	Normal

Sumber: Penulis, 2026

Tabel 5.9 Rekap Hasil Pengujian Kecepatan Respons

Sumber *Input*	Jumlah Uji	Waktu Minimum	Waktu Maksimum	Rata-rata Waktu
Kamera	5	1.098 *ms*	1.312 *ms*	1.198 *ms*
Galeri	5	987 *ms*	1.934 *ms*	1.421 *ms*
Keseluruhan	10	987 *ms*	1.934 *ms*	1.310 *ms*

Sumber: Penulis, 2026

Berdasarkan hasil pada Tabel 5.13, rata-rata waktu respons sistem secara keseluruhan adalah 1.310 milidetik (±1.3 detik). Nilai ini berada jauh di bawah ambang batas yang ditetapkan pada kebutuhan *non*-fungsional, yaitu ≤ 3 detik.

Waktu respons tercepat dicatat pada gambar berukuran kecil dari galeri (987 *ms*), sedangkan waktu respons terlama terjadi pada gambar berukuran besar dari galeri (1.934 *ms*).

Perbedaan waktu respons antara input kamera dan galeri disebabkan oleh ukuran berkas gambar yang dihasilkan. Gambar dari kamera umumnya mengalami kompresi otomatis oleh sistem *android*, sementara gambar dari galeri dapat memiliki ukuran yang lebih bervariasi tergantung sumber asalnya. Secara keseluruhan, performa kecepatan respons aplikasi *nutriscan* dinilai sangat baik dan memenuhi kebutuhan *real-time* yang telah ditentukan.

5.4.4 Pembahasan Hasil Pengujian

Berdasarkan keseluruhan hasil pengujian yang telah dilakukan, dapat disusun rangkuman evaluasi kinerja aplikasi *nutriscan* sebagai berikut:

Tabel 5.10 Ringkasan Evaluasi Kinerja Aplikasi *NutriScan*

No	Aspek Pengujian	Target	Hasil	Penilaian
1	Pengujian Fungsional (keseluruhan)	23/23 skenario berhasil	23/23 berhasil (100%)	✅ Sangat Memuaskan
2	Autentikasi (*Login*/*Register*/*Logout*)	8/8 skenario berhasil	8/8 berhasil (100%)	✅ Sangat Memuaskan
3	Akurasi – Pencahayaan Baik	≥ 85%	100% (20/20 tepat)	✅ Sangat Memuaskan
4	Akurasi – Pencahayaan Rendah	≥ 60%	60% (12/20 tepat)	✅ Memuaskan
5	Akurasi Keseluruhan	≥ 75%	80% (32/40 tepat)	✅ Memuaskan
6	Kecepatan Respons	≤ 3.000 *ms*	Rata-rata 1.310 *ms*	✅ Sangat Memuaskan
7	Fitur *Voice Output*	Berfungsi tanpa *error*	Berfungsi normal di semua uji	✅ Sangat Memuaskan
8	Fitur *Chatbot*	Respons < 3 detik	Rata-rata < 2 detik	✅ Sangat Memuaskan
9	Penyimpanan Riwayat per Akun	Data tersimpan per *userid*	Data berhasil tersimpan & terbaca per akun	✅ Sangat Memuaskan

Sumber: Penulis, 2026

Hasil pengujian secara keseluruhan menunjukkan bahwa aplikasi *nutriscan* telah berhasil memenuhi seluruh target kinerja yang ditetapkan. Sistem mampu mendeteksi buah dan sayuran secara akurat pada kondisi pencahayaan yang memadai, memberikan respons yang cepat dalam hitungan detik, serta

menyediakan fitur pendukung seperti *voice output* dan *chatbot* yang berjalan tanpa gangguan. Sistem autentikasi berbasis *firebase authentication* juga terbukti berjalan dengan benar pada seluruh skenario uji, termasuk validasi *input*, penanganan *error*, dan pemisahan data riwayat antar pengguna.

Satu-satunya area yang perlu ditingkatkan adalah akurasi pada kondisi pencahayaan rendah (60%), yang merupakan keterbatasan inheren dari *model image labeling* berbasis *google ml kit* tanpa *custom training*. Solusi yang dapat diterapkan pada pengembangan berikutnya adalah penerapan teknik *preprocessing* gambar seperti peningkatan kecerahan (*brightness enhancement*) dan normalisasi kontras sebelum citra diproses oleh *ml kit*, sehingga akurasi pada kondisi pencahayaan buruk dapat ditingkatkan secara signifikan.

Secara keseluruhan, pengembangan aplikasi *nutriscan* menggunakan *flutter* dan *google ml kit* terbukti menghasilkan sistem deteksi buah dan sayuran yang fungsional, akurat, dan responsif sesuai dengan tujuan penelitian yang telah ditetapkan. Integrasi fitur *voice output* dan *chatbot* juga berhasil meningkatkan interaktivitas serta aksesibilitas aplikasi dibandingkan dengan sistem identifikasi manual yang dilakukan sebelumnya.

BAB VI
KESIMPULAN DAN SARAN

6.1 Kesimpulan

Berdasarkan hasil penelitian, pengembangan, dan pengujian yang telah dilakukan terhadap aplikasi *mobile nutriscan* berbasis *flutter* dengan memanfaatkan *google ml kit* sebagai teknologi *image labeling*, dapat ditarik kesimpulan sebagai berikut:

1. Aplikasi *mobile nutriscan* berhasil dikembangkan menggunakan *framework flutter* dan bahasa pemrograman *dart* dengan metode *prototype* melalui empat tahapan iteratif. Hasilnya adalah aplikasi *android* yang stabil dan responsif, mampu berjalan pada perangkat *android 8.0* (*API Level 26*) ke atas, dengan seluruh 8 fitur fungsional berhasil diimplementasikan dan mencapai tingkat keberhasilan 100% berdasarkan pengujian *black box testing* terhadap 14 skenario uji.
2. Integrasi *image labeling api* dari *google ml kit* ke dalam proyek *flutter* berhasil dilakukan menggunakan *package google_mlkit_image_labeling*. Sistem mampu memproses citra dari kamera maupun galeri perangkat secara *on-device* tanpa memerlukan koneksi internet untuk fitur deteksi utama, dengan proses berjalan secara *asynchronous* sehingga antarmuka pengguna tetap responsif selama pemrosesan berlangsung.
3. Tingkat akurasi deteksi sistem mencapai 100% pada kondisi pencahayaan baik, 60% pada kondisi pencahayaan rendah, dan rata-rata keseluruhan sebesar 80% dari 40 sampel gambar yang diuji. Rata-rata waktu respons sistem adalah 1.310 milidetik (±1,3 detik), jauh di bawah target yang ditetapkan yaitu ≤ 3 detik, sehingga sistem dinilai memenuhi kebutuhan *real-time*.
4. Fitur *voice output* berhasil diintegrasikan menggunakan *package flutter_tts* dan terbukti berjalan normal pada seluruh skenario pengujian tanpa *error*. Fitur ini meningkatkan aksesibilitas aplikasi, khususnya bagi pengguna dengan keterbatasan baca seperti anak-anak,

sehingga hasil identifikasi dapat disampaikan secara lebih komunikatif dan inklusif.

5. Integrasi *chatbot nutribot* dan *firebase firestore* menjadikan *nutriscan* tidak sekadar alat deteksi, tetapi juga media edukatif interaktif. *Chatbot* mampu memberikan respons terkait kandungan gizi dan manfaat buah atau sayuran dalam waktu rata-rata di bawah 2 detik, sementara seluruh data riwayat deteksi dan sesi percakapan berhasil tersimpan dan terbaca dari *firestore* secara *real-time*.

6.2 Saran

Berdasarkan hasil penelitian dan keterbatasan yang ditemukan selama proses pengembangan, berikut adalah saran untuk perbaikan dan pengembangan lebih lanjut:

1. Mengimplementasikan teknik *preprocessing* citra seperti peningkatan kecerahan otomatis, normalisasi kontras, dan *image sharpening* sebelum gambar diproses oleh *ml kit*, guna meningkatkan akurasi deteksi pada kondisi pencahayaan rendah yang saat ini masih berada di angka 60%.
2. Membangun *custom model machine learning* menggunakan *dataset* buah dan sayuran lokal Indonesia, dilatih dengan *google automl vision* atau *tensorflow lite*, agar jenis buah dan sayuran khas Indonesia seperti rambutan, salak, dan durian dapat dikenali dengan lebih baik.
3. Memperluas dukungan *platform* ke *ios* dengan memanfaatkan kemampuan lintas *platform flutter*, sehingga jangkauan pengguna aplikasi dapat ditingkatkan secara signifikan.
4. Meningkatkan kemampuan *chatbot nutribot* dengan mengintegrasikan *large language model* (*LLM*) seperti *gemini api* atau *openai gpt*, agar *chatbot* dapat memahami pertanyaan yang lebih kompleks dan memberikan respons yang lebih kontekstual dan natural.
5. Menambahkan fitur analisis kandungan gizi secara otomatis berdasarkan objek yang terdeteksi, mencakup informasi kalori, karbohidrat, protein, vitamin, dan mineral, serta rekomendasi porsi konsumsi harian, sehingga

nutriscan dapat berfungsi sebagai asisten gizi *digital* yang lebih komprehensif.

6. Melakukan pengujian pada lebih banyak perangkat *android* dengan rentang spesifikasi yang lebih luas serta berbagai kondisi penggunaan nyata, seperti pencahayaan luar ruangan dan gambar dengan latar belakang kompleks, untuk memperoleh gambaran yang lebih komprehensif mengenai ketangguhan dan kompatibilitas aplikasi.
7. Menambahkan fitur pemindaian *barcode* atau *qr code* pada produk pangan kemasan untuk mendapatkan informasi nutrisi langsung dari *database* produk resmi, sehingga cakupan penggunaan aplikasi dapat diperluas menjadi *platform* informasi pangan yang lebih lengkap.

DAFTAR PUSTAKA

Abdi, A. M. (2024). *Tinjauan Langkah Pengujian Perangkat Lunak Dengan Metode Black Box Menggunakan Teknik Equivalance Partitions*. *2*, 59–65.

Aharony, A., Entin-Wohlman, O., & Kudlis, A. (2022). Different critical behaviors in perovskites with a structural phase transition from cubic-to-trigonal and cubic-to-tetragonal symmetry. *Physical Review B*, *105*(10), 1–10. https://doi.org/10.1103/PhysRevB.105.104101

Ahmad, K., Abdelrazek, M., Arora, C., Grundy, J., & Bano, M. (2023). Requirements Elicitation and Modelling of Artificial Intelligence Systems: An Empirical Study. *International Conference on Evaluation of Novel Approaches to Software Engineering, ENASE - Proceedings*, *2023-April*(Enase), 126–137. https://doi.org/10.5220/0011842300003464

Andrade, C. (2021). A Student's Guide to the Classification and Operationalization of Variables in the Conceptualization and Design of a Clinical Study: Part 1. *Indian Journal of Psychological Medicine*, *43*(2), 177–179. https://doi.org/10.1177/0253717621994334

Appadoo, A., Gopaul, Y., & Pudaruth, S. (2023). FruVegy: An Android App for the Automatic Identification of Fruits and Vegetables using Computer Vision and Machine Learning. *International Journal of Computing and Digital Systems*, *13*(1), 169–178. https://doi.org/10.12785/ijcds/130114

Ayu, D., Wulandari, N., Alfin, A., Bahar, H., Arfananda, M. G., & Apriyani, H. (2021). Al-Ruhamaa ' Bogor Yatim Center Foundation. *Jurnal PILAR Nusa Mandiri*, *17*(2), 127–136.

Azahari, M. H., Ali, F., & Ali, H. (2022). The Development of an Online Food Ordering System for JomMakan Restaurant. *Applied Information Technology and Computer Science*, *3*(1), 369–376. http://publisher.uthm.edu.my/periodicals/index.php/aitcs

Barroga, E., & Matanguihan, G. J. (2022). A Practical Guide to Writing Quantitative and Qualitative Research Questions and Hypotheses in Scholarly Articles. *Journal of Korean Medical Science*, *37*(16), 1–18. https://doi.org/10.3346/jkms.2022.37.e121

Batubara, M. I., Panjaitan, M. I., Rajagukguk, D. M., & Sihombing, M. J. T. (2025). Academic Chatbot Based on Natural Language Processing for Student Services at STMIK Mulia Darma. *The IJICS (International Journal of Informatics and Computer Science)*, *9*(2), 116–120. https://doi.org/10.30865/ijics.v9i2.8970

Białkowski, D., & Smołka, J. (2022). Evaluation of Flutter framework time efficiency in context of user interface tasks. *Journal of Computer Sciences Institute*, *25*(June), 309–314. https://doi.org/10.35784/jcsi.3007

Bonesso, G. A., De Godoy, C. M. G., Osaki, T. H., Osaki, M. H., Moreira, B. M. R. T. dos, & Coelho, R. C. (2025). Clinically-Validated Innovative Mobile Application for Assessing Blinking and Eyelid Movements. *ArXiv*, *1*.

Bradley, L. (2022). *Australian Journal of Applied Linguistics Mobile Language Learning Designs and Contexts for Newly Arrived Migrants*. *5*(3), 1–20.

Carrera-Rivera, A., Ochoa, W., Larrinaga, F., & Lasa, G. (2022). How-to conduct a systematic literature review: A quick guide for computer science research.

MethodsX, *9*, 101895. https://doi.org/10.1016/j.mex.2022.101895
Devirgiliis, C., Guberti, E., Mistura, L., & Raffo, A. (2024). Effect of Fruit and Vegetable Consumption on Human Health: An Update of the Literature. *Foods*, *13*(19), 1–17. https://doi.org/10.3390/foods13193149
Drolia, M., Papadakis, S., Sifaki, E., & Kalogiannakis, M. (2022). Mobile Learning Applications for Refugees: A Systematic Literature Review. *Education Sciences*, *12*(2). https://doi.org/10.3390/educsci12020096
Ellis, P. (2024). Research methods: quantitative observation. *Wounds UK*, *20*(2), 73–74.
Fahmi. (2024). *Sistem Informasi Penjualan Berbasis Web pada Toko Hanif dengan Metode Prototype*. *1*(2), 290–299.
Fajri, R., & Fitria, F. (2023). Pengembangan real-time object detection system pada perangkat single-board computer. *Media Online*, *4*(2), 1156–1167. https://doi.org/10.30865/klik.v4i2.1224
Giani, M. S., & Martin, A. (2021). Mobilizing Developmental Education: The Causal Effect of Mobile App Courseware on the College Outcomes of Developmental Education Students. *Educational Evaluation and Policy Analysis*, *43*(4), 668–687. https://doi.org/10.3102/01623737211013782
Gökçearslan, Ş., Tosun, C., & Erdemir, Z. G. (2024). Benefits, Challenges, and Methods of Artificial Intelligence (AI) Chatbots in Education: A Systematic Literature Review. *International Journal of Technology in Education*, *7*(1), 19–39. https://doi.org/10.46328/ijte.600
Gulzar, Y. (2023). Fruit Image Classification Model Based on MobileNetV2 with Deep Transfer Learning Technique. *Sustainability*, *15*, 1906. https://doi.org/10.3390/su15031906
Haider, A. (2021). Evaluation of cross-platform technology Flutter from the user's perspective. *Degree Project Computer Science and Engineering*. https://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-305570
Haristiani, N., & Rifai, M. M. (2021). Chatbot-based application development and implementation as an autonomous language learning medium. *Indonesian Journal of Science and Technology*, *6*(3), 561–576. https://doi.org/10.17509/ijost.v6i3.39150
Hindrichsen, L. A., & Eyre, J. (2023). *Erasing Intersectionality : Silencing Female Black Voices in English Romantic Texts*. *95*(July 2021), 298–311.
Huang, C. J., & Kolopitawondal, J. (2023). User Experience Design and Evaluation on Mobile Investment Application User Interface Prototype. *Applied Human Factors and Ergonomics International*, *110*, 200–208. https://doi.org/10.54941/ahfe1003192
Huang, Y., Hu, H., & Chen, C. (2021). Robustness of on-Device Models: Adversarial Attack to Deep Learning Models on Android Apps. *Proceedings - International Conference on Software Engineering*, (Section IV), 101–110. https://doi.org/10.1109/ICSE-SEIP52600.2021.00019
Ibrahim, R., & Lee, Z. Y. (2023). *Project Development for Blood Bank Application and Convertor for Software Testing*. 41–52.
Ikhsan, M., Putri, D. M., Nurjanah, S., Rahmawati, A., Fahrizal, & Bastul Wajhi, A. (2025). Implementasi Teknologi Chatbot sebagai Media Informasi di Universitas Negeri Medan mengembangkan chatbot berbasis Line di Politeknik Bandung dengan menggunakan metode Sentence Similarity

Measurement dan Bigram untuk meningkatkan akurasi pencarian kata kun. *Jurnal Teknik Mesin, Industri, Elektro Dan Informatika*, *4*(1).

Imran, M. A., Fauzi, A., & Khair, H. (2024). Rancang Bangun Kontrol Bel Otomatis Berdasarkan Jadwal Perkuliahan Menggunakan Internet of Things (IoT). *Modem : Jurnal Informatika Dan Sains Teknologi.*, *2*(4), 21–32. https://journal.aptii.or.id/index.php/Modem/article/view/225

Javaid, M., Haleem, A., Khan, I. H., & Suman, R. (2023). Understanding the potential applications of Artificial Intelligence in Agriculture Sector. *Advanced Agrochem*, *2*(1), 15–30. https://doi.org/10.1016/j.aac.2022.10.001

Kesavan, R., Gay, D., Thevessen, D., Shah, J., & Mohan, C. (2023). Firestore: The NoSQL Serverless Database for the Application Developer. *Proceedings - International Conference on Data Engineering*, *2023-April*, 3376–3388. https://doi.org/10.1109/ICDE55515.2023.00259

Khattab, O., & Zaharia, M. (2020). ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT. *SIGIR 2020 - Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 39–48. https://doi.org/10.1145/3397271.3401075

Kikuchi, S., Kobayashi, T., Ogawa, Y., & Uchida, H. (2022). Yukawa textures in modular symmetric vacuum of magnetized orbifold models. *Progress of Theoretical and Experimental Physics*, *2022*(3). https://doi.org/10.1093/ptep/ptac035

Kim, M., Shim, Y., Lee, S., Loh, H., & Park, J. (2021). Behavioral Testing of Deep Neural Network Knowledge Tracing Models. *Proceedings of the 14th International Conference on Educational Data Mining, EDM 2021*, (Edm), 105–117.

Kurniawan, I., Dhoni, R., Arya Saputra, D., Rahmat Wahyudin, F., & Mubais, A. (2023). Implementasi Metode Prototype Pada Sistem Informasi Digital Angkringan Berbasis Web Di Kecamatan Mayong. *Journal of Information System and Computer*, *3*(1), 20–26. https://doi.org/10.34001/jister.v3i1.597

Kusnetsoff, D. (2021). *Mobile Real-Time Object Detection With Flutter*. 1–42.

Laurence, M. (2020). AI and Machine Learning for On-Device Development. In *Journal of Chemical Information and Modeling* (Vol. 110, Number 9).

Lorenzoni, G., Alencar, P., Nascimento, N., & Cowan, D. (2021). *Machine Learning Model Development from a Software Engineering Perspective: A Systematic Literature Review*. http://arxiv.org/abs/2102.07574

Mahmoud S Abdallah, B. M. (2025). How Conversational AI Chatbots Support and Reinforce Self-Regulated Language Learning. *Online Submission*, (June), 1–21.

Maryani, Prabowo, H., Gaol, F. L., & Hidayanto, A. N. (2022). Comparison of the System Development Life Cycle and Prototype Model for Software Engineering. *International Journal of Emerging Technology and Advanced Engineering*, *12*(4), 155–162. https://doi.org/10.46338/ijetae0422_19

Milosz, M., Plechawska-Wójcik, M., & Dzieńkowski, M. (2024). Testing the Quality of the Mobile Application Interface Using Various Methods—A Case Study of the T1DCoach Application. *Applied Sciences (Switzerland)*, *14*(15). https://doi.org/10.3390/app14156583

Muneeb, M., F. Feng, S., & Henschel, A. (2022). *Deep Learning Pipeline for Image*

Classification on Mobile Phones. 1–20. https://doi.org/10.5121/csit.2022.120901

Mwita, K. M. (2022). Research in Business & Social Science Factors to consider when choosing data collection methods. *International Journal of Research in Business & Social Science*, *11*(5), 532–538.

Nabawi, R., Tarigan, B. G., Saputra, P. B., & Sukmadiningtyas. (2024). Perancangan Mobile App Digital Marketing Milkyo dengan Metode Prototype Design. *Jurnal Indonesia : Manajemen Informatika Dan Komunikasi*, *5*(2), 1233–1244. https://doi.org/10.35870/jimik.v5i2.640

Nadtayay, N., Wongsaphan, M., & Wu, C. (2025). The Application of CEFR Standard Framework in Enhancing Communicative English Competencies: The Bibliometric Analysis and Its Support to Sustainable Development Goals (SDGs). *International Journal of Language Education*, *9*(4), 792–803. https://doi.org/10.26858/ijole.v1i2.82101

Nguyen, N. (2022). Building an E-commerce Application Utilizing Firebase Cloud service. *Metropolia University of Applied Sciences*, (Mobile Solutions).

Pebriani, S. R., Islam, U., Sumatera, N., Islam, U., & Sumatera, N. (2023). *ARJUNA+Vol+1+no+6+Desember+2023+hal+21-32*. *1*(6).

Rachma, N., & Muhlas, I. (2022). Comparison Of Waterfall And Prototyping Models In Research And Development (R&D) Methods For Android-Based Learning Application Design. *Jurnal Inovatif : Inovasi Teknologi Informasi Dan Informatika*, *5*(1), 36. https://doi.org/10.32832/inova-tif.v5i1.7927

Rahayu Dewi, N. L. A. M., Hartati, R. S., & Divayana, Y. (2021). Penerapan Metode Prototype dalam Perancangan Sistem Informasi Penerimaan Karyawan Berbasis Website pada Berlian Agency. *Majalah Ilmiah Teknologi Elektro*, *20*(1), 147. https://doi.org/10.24843/mite.2021.v20i01.p17

Rahmawati, I., & Sari, D. P. (2024). *Aplikasi berbasis android menggunakan flutter framework*. *9*(2), 979–993.

Ramazan, S. (2021). El análisis de documentos como método de investigación. *Revista de Educación de La Universidad Kocaeli*, *4*(1), 227–256.

Rathi, T., & Ronald, B. (2022). Questionnaire as a Tool of Data Collection in Empirical Research. *Journal of Positive School Psychology*, *2022*(5), 7697–7699. http://journalppw.com

Riad, K., Islam, S., Shahriar, H., Zhang, C., Valero, M., Sneha, S., & Ahamed, S. (2021). Plugin-Based Tool for Teaching Secure Mobile Application Development. *Information Systems Education Journal*, *19*(2), 25–34. https://eric.ed.gov/?id=EJ1297704

Rospricilia, T. A., & Ma'ady, M. N. P. (2024). Pemodelan Integration Use Case (IUC): Perancangan Use Case Diagram (UML) untuk Sistem-sistem yang Terintegrasi. *INTEGER: Journal of Information Technology*, *9*(2), 165–172. https://doi.org/10.31284/j.integer.2024.v9i2.6345

Saman, P., & Indah Ratnasari, C. (2022). Pengujian Black Box Pada Aplikasi Pembelajaran Bahasa Mandarin Berbasis Android Black Box Testing on an Android-Based Mandarin Learning Application. *Jurnal Ilmiah Intech : Information Technology Journal of UMUS*, *4*(1), 10–22.

Saputra, D. H., Imran, B., & Juhartini. (2023). Object Detection Untuk Mendeteksi Citra Buah-Buahan Menggunakan Metode Yolo. *Jurnal Kecerdasan Buatan Dan Teknologi Informasi*, *2*(2), 70–80. https://doi.org/10.69916/jkbti.v2i2.18

Sarker, I. H. (2021). Machine Learning: Algorithms, Real-World Applications and Research Directions. *SN Computer Science*, *2*(3). https://doi.org/10.1007/s42979-021-00592-x

Shahil-Feroz, A., Yasmin, H., Saleem, S., Bhutta, Z., & Seto, E. (2023). Remote Moderated Usability Testing of a Mobile Phone App for Remote Monitoring of Pregnant Women at High Risk of Preeclampsia in Karachi, Pakistan. *Informatics*, *10*(4). https://doi.org/10.3390/informatics10040079

Snekha, S., & Ayyanathan, N. (2023). An Educational CRM Chatbot for Learning Management System. *Shanlax International Journal of Education*, *11*(4), 58–62. https://doi.org/10.34293/education.v11i4.6360

Syarif, M., & Risdiansyah, D. (2024). Pemanfaatan Metode Prototype Dalam Perancangan Sistem Informasi Penjualan Berbasis Website. *JATI (Jurnal Mahasiswa Teknik Informatika)*, *8*(4), 7945–7952. https://doi.org/10.36040/jati.v8i4.10467

Taherdoost, H. (2021). Data collection methods and tools for research; A Step-by-step guide to choose data collection technique for academic and business research projects. *International Journal of Academic Research in Management (IJARM)*, *10*(1), 10–38.

Tariq, A. (2025). Mobile Application Performance Testing: Advanced Methodologies and Quality Assurance Frameworks for Contemporary Mobile Development. *International Journal of Mobile Applications and Technologies*, *1*(2). https://www.wr-publishing.org/index.php/ijmat/article/view/321

Tawasuli, L., & Huda, M. (2023). Technology and Informatics Insight Journal Perancangan User Interface dan User Experience pada Aplikasi Resep Masakan Rumahan Menggunakan Aplikasi Figma. *Technology and Informatics Insight Journal*, *2*(2). https://jurnal.universitasputrabangsa.ac.id/index.php/tiij

Topal, I. H. (2025). Research on Voscreen in English language education: A systematic literature review and content analysis. *JALT CALL Journal*, *21*(2), 1–43. https://doi.org/10.29140/jaltcall.v21n2.2184

Trisnadoli, A., & Yanti, D. (2022). Analysis of Prototyping Methods Implementation for Library Information Systems Development in SMAN 3 Bengkalis (Analisis Implementasi Metode Prototyping untuk Pengembangan Sistem Informasi Perpustakaan di SMAN 3 Bengkalis). *Jurnal Nasional Pendidikan Teknik Informatika (JANAPATI)*, *11*(3), 280–289.

Usman, M., Ali, N. Bin, & Wohlin, C. (2023). A Quality Assessment Instrument for Systematic Literature Reviews in Software Engineering. *E-Informatica Software Engineering Journal*, *17*(1). https://doi.org/10.37190/e-Inf230105

von Wangenheim, C. G., & Dirschnabel, G. (2023). *UX Heuristics and Checklist for Deep Learning powered Mobile Applications with Image Classification*. http://arxiv.org/abs/2307.05513

Weichbroth, P. (2024). applied sciences Usability Testing of Mobile Applications : *Applied Sciences*, *14*, 1–27.

Widya Ningsih, & Habibah Nurfauziah. (2023). Perbandingan Model Waterfall Dan Metode Prototype Untuk Pengembangan Aplikasi Pada Sistem Informasi. *Jurnal Ilmiah Metadata*, *5*(1), 83–95.

Windarti, S., Ramadani, T., Mawardani, N., & Melani Bambang Putri Utami, A.

(2025). Desain Media Edukasi Interaktif: Flowchart Alur Layanan It Di Rumah Sakit Ibnu Sina Yw-Umi Makassar Sebagai Sarana Peningkatan Pemahaman. *Jurnal Abdimas Jatibara STIKES Yayasan RS Dr. Soetomo*, *4*(1), 61–69.

Zaidi, S. S. A., Ansari, M. S., Aslam, A., Kanwal, N., Asghar, M., & Lee, B. (2022). A survey of modern deep learning based object detection models. *Digital Signal Processing: A Review Journal*, *126*, 1–18. https://doi.org/10.1016/j.dsp.2022.103514

LAMPIRAN

Lampiran 1 Dokumentasi Penelitian Buah-buahan di Lokasi Pasar Lama Tanjung Enim, Kab. Muara Enim





























Lampiran 2 Dokumentasi Penelitian Sayur-sayuran di Lokasi Pasar Baru Tanjung Enim, Kab. Muara Enim





















FAKULTAS ILMU KOMPUTER
UNIVERSITAS PRABUMULIH

KESEDIAAN SEBAGAI PEMBIMBING SKRIPSI
PROGRAM STUDI INFORMATIKA

Yang bertanda tangan di bawah ini:

Nama	: Nur Aini Hutagalung, S.Kom., M.Si., M.Kom.
NUPTK	: 2434764 664300022
Program Studi	: Informatika
Fakultas	: Ilmu Komputer

Bersedia menjadi pembimbing I Skripsi, dan

Nama	: Hermizahadiwidastra, S.Kom., M.Kom.
NUPTK	: 2863762663130192
Program Studi	: Informatika
Fakultas	: Ilmu Komputer

Bersedia menjadi pembimbing II Skripsi.

Dari mahasiswa Program Studi Informatika Fakultas Ilmu Komputer Universitas Prabumulih di bawah ini:

Nama	: Alfi Syahrin Al Qodar
NIM	: 2022230038
Judul Skripsi	: Pengembangan *Google MI Kit* di *Flutter* untuk Deteksi Buah dan Sayuran

Demikian surat kesediaan membimbing ini kami buat, atas perhatiannya kami ucapkan terima kasih.

Prabumulih, Juli 2026

Dosen Pembimbing I	Dosen Pembimbing II
(Nur Aini Hutagalung, S.Kom., M.Si., M.Kom.)	(Hermizahadiwidastra, S.Kom., M.Kom.)
NUPTK. 2434764 664300022	NUPTK. 2863762663130192

FAKULTAS ILMU KOMPUTER
UNIVERSITAS PRABUMULIH

KARTU BIMBINGAN SKRIPSI

AMA	: Alfi Syahrin Al Qodar
IM	: 2022230038
ROGRAM STUDI	: Informatika
AMA PEMBIMBING I	: Nur Aini Hutagalung., S.Kom,, M.Si., M.Kom.
UDUL	: Pengembangan Google ML Kit di Flutter untuk Deteksi Buah dan Sayuran

NO	TGL. KONSULTASI	TOPIK POKOK YANG DIBICARAKAN	TANDA TANGAN PEMB. I / II	TGL MENG-HADAP KEMBALI
1	22/6/2026	Bab I		
2		Rev - Latar belakang		
3		- Batasan Masalah		
4		- Tujuan penelitian		
5		- Sistematika penulisan		
6	23/6/2026	Revisi - Latar belakang		
7	25/6/2026	Acc Bab I		
8		Bab II		
9		Revisi : - Cara penyajian		
10		- penulisan daftar pustaka		
11				
12	22/7/2026	Acc Bab III		
13		Menyusun Bab IV		
14		Revisi - Perancangan Sistem		
15		- Database		
16		- UI/UX		

ulai Bimbingan : .....

ıtas Akhir Bimbingan : .....

Mengetahui Kaprodi Informatika

IWAN SETIAWAN, S.Kom., M.Kom

FAKULTAS ILMU KOMPUTER
UNIVERSITAS PRABUMULIH

KARTU BIMBINGAN SKRIPSI

NAMA : **Alfi Syahrin Al Qodar**
NIM : **2022230038**
PROGRAM STUDI : **Informatika**
NAMA PEMBIMBING I : **Nur Aini Hutagalung., S.Kom,, M.Si., M.Kom.**
JUDUL : **Pengembangan Google ML Kit di Flutter untuk Deteksi Buah dan Sayuran**

NO	TGL. KONSULTASI	TOPIK POKOK YANG DIBICARAKAN	TANDA TANGAN PEMB. I / II	TGL MENG-HADAP KEMBALI
1	24/7/2026	Rev : - Database		
2		- UI / UX		
3				
4	24/7/2026	Acc Bab IV		
5		Perbaiki Bab V dan VI		
6		Rev : perbaiki fgt		
7				
8	27/7/2026	Bab V : perbaiki fgt		
9		: - Saran		
10	27/7/2026	Acc Bab - Saran		
11				
12	29/7/2026	Acc Bab VI		
13				
14				
15				
16				

Mulai Bimbingan : .....
Batas Akhir Bimbingan : .....

Mengetahui Kaprodi Informatika

IWAN SETIAWAN, S.Kom., M.Kom

FAKULTAS ILMU KOMPUTER
UNIVERSITAS PRABUMULIH

KARTU BIMBINGAN SKRIPSI

NAMA : **Alfi Syahrin Al Qodar**
NIM : **2022230038**
PROGRAM STUDI : **Informatika**
NAMA PEMBIMBING II : **Hermizahadiwidastra., S.Kom., M.Kom.**
JUDUL : **Pengembangan Google ML Kit di Flutter untuk Deteksi Buah dan Sayuran**

NO	TGL. KONSULTASI	TOPIK POKOK YANG DIBICARAKAN	TANDA TANGAN PEMB. I / II	TGL MENG-HADAP KEMBALI
1	12/05 2026	- BAB I Latar Belakang		
2		Selanjutnya - 1. Fenomena		
3		2. Gap Fenomena		
4		3. Masalah.		
5		4. Gap Riset		
6		5. Novelty		
7		6. Urgensi		
8		- Isi Voice tambahkan		
9		- Format penulisan		
10		- Bhs asing.		
11		- BAB II penulisan terdahulu		
12		harus ada di kutip dr tulisan		
13		Bab tinjauan berapa.		
14				
15				
16				

Mulai Bimbingan : .....
Batas Akhir Bimbingan : .....

Mengetahui Kaprodi Informatika

IWAN SETIAWAN, S.Kom.,M.Kom.

FAKULTAS ILMU KOMPUTER
UNIVERSITAS PRABUMULIH

KARTU BIMBINGAN SKRIPSI

NAMA : Alfi Syahrin Al Qodar
NIM : 2022230038
PROGRAM STUDI : Informatika
NAMA PEMBIMBING II : Hermizahadiwidastra., S.Kom., M.Kom.
JUDUL : Pengembangan Google ML Kit di Flutter untuk Deteksi Buah dan Sayuran

NO	TGL. KONSULTASI	TOPIK POKOK YANG DIBICARAKAN	TANDA TANGAN PEMB. I / II	TGL MENG-HADAP KEMBALI
1	23/06/26	ACC Bab I		
2		Lanjut Bab II		
3		Bab II		
4		- Flutter		
5		- Tahap perlatan terdahulu.		
6				
7		ACC Bab II		
8		Lanjut Bab III		
9				
10		Bab III		
11		3.3 Data Primer.		
12		3.4 Subjek dan Objek		
13		penelitian		
14		3.4 Pengumpulan Data		
15		+ Observasi lapangan		
16		2. Wawancara. Pakar		

lulni Bimbingan : .....
ıtas Akhir Bimbingan : .....

Mengetahui Kaprodi Informatika

IWAN SETIAWAN, S.Kom.,M.Kom.

FAKULTAS ILMU KOMPUTER
UNIVERSITAS PRABUMULIH

KARTU BIMBINGAN SKRIPSI

NAMA : **Alfi Syahrin Al Qodar**
NIM : **2022230038**
PROGRAM STUDI : **Informatika**
NAMA PEMBIMBING II : **Hermizahadiwidastra., S.Kom., M.Kom.**
JUDUL : **Pengembangan Google ML Kit di Flutter untuk Deteksi Buah dan Sayuran**

NO	TGL. KONSULTASI	TOPIK POKOK YANG DIBICARAKAN	TANDA TANGAN PEMB. I / II	TGL MENG-HADAP KEMBALI
1	23/06/26	~~BAB III~~ BAB III:		
2		3.2. Kualitatif		
3		Variabel nya?		
4		-"- pertanyaannya?		
5		-"- Objek yg ditanyai?		
6		Sumber metodologi yg digunakan		
7		Kuantitatif		
8		- Dimensi, indikator		
9		- Referensi.		
10	17/07/2026			
11		BAB III		
12		- Subjek Penelitian.		
13		- Sumber Referensi / kutipan		
14		setiap paragraf?		
15		- Kurang observasinya?		
16		- Tdk boleh kutipan digunakan 2x		

Mulai Bimbingan : .....
Batas Akhir Bimbingan : .....

Mengetahui Kaprodi Informatika

IWAN SETIAWAN, S.Kom.,M.Kom.

FAKULTAS ILMU KOMPUTER
UNIVERSITAS PRABUMULIH

KARTU BIMBINGAN SKRIPSI

NAMA	: **Alfi Syahrin Al Qodar**
NIM	: **2022230038**
PROGRAM STUDI	: **Informatika**
NAMA PEMBIMBING II	: **Hermizahadiwidastra., S.Kom., M.Kom.**
JUDUL	: **Pengembangan Google ML Kit di Flutter untuk Deteksi Buah dan Sayuran**

NO	TGL. KONSULTASI	TOPIK POKOK YANG DIBICARAKAN	TANDA TANGAN PEMB. I / II	TGL MENG-HADAP KEMBALI
1	21/07	ACC BAB III		
2	/2026			
3		Lanjut BAB IV		
4		BAB IV		
5	23/07	Penjelasan Use Case		
6	/2026			
7		- Pengguna		
8		- Fitur		
9		- Google ML Kit		
10				
11		- Dataset apel, jeruk		
12				
13	24/07	ACC BAB IV		
14	/2026	Lanjut BAB V		
15	24/07	BAB V Gambar, kode, tabel		
16		penjelasan flowchart		

Mulai Bimbingan : .....

Batas Akhir Bimbingan : .....

Mengetahui Kaprodi Informatika

IWAN SETIAWAN, S.Kom.,M.Kom.

FAKULTAS ILMU KOMPUTER
UNIVERSITAS PRABUMULIH

KARTU BIMBINGAN SKRIPSI

NAMA : **Alfi Syahrin Al Qodar**
NIM : **2022230038**
PROGRAM STUDI : **Informatika**
NAMA PEMBIMBING II : **Hermizahadiwidastra., S.Kom., M.Kom.**
JUDUL : **Pengembangan Google ML Kit di Flutter untuk Deteksi Buah dan Sayuran**

NO	TGL. KONSULTASI	TOPIK POKOK YANG DIBICARAKAN	TANDA TANGAN PEMB. I / II	TGL MENG-HADAP KEMBALI
1	27/7 2026	ACC Bab V ✓		
2				
3				
4		ACC Bab VI ✓		
5				
6				
7				
8		Siap Uji		
9				
10				
11				
12				
13				
14				
15				
16				

Mulai Bimbingan .....
Batas Akhir Bimbingan .....

Mengetahui Ketua Prodi Informatika

IWAN SETIAWAN, S.Kom.,M.Kom.

FAKULTAS ILMU KOMPUTER
UNIVERSITAS PRABUMULIH

PENILAIAN REVIEW SKRIPSI

Nama	: Alfi Syahrin Al Qodar
NIM	: 2022230038 Aplikasi NutriScan
Judul Skripsi	: Pengembangan ~~Google ML Kit dr Flutter~~ untuk Deteksi Buah dan Sayuran
Hari/Tanggal	: Selasa / 28 Juli 2026
Saran dan masukan	: Lakukan perbaikan max 7 hari terhitung hari ini

Rekomendasi*:

a. Dilanjutkan tanpa perbaikan
b. Dilanjutkan dengan perbaikan
c. Ditolak

Prabumulih, ... 2026
Pembimbing I

(Nur Aini Hutagalung, S.Kom., M.Si., M.Kom)
NUPTK. 2434764 664300022

pilih salah satu

Jalan Patra No.50 RT.01 RW.03 Kelurahan Sukaraja Kecamatan Prabumulih Selatan
Kota Prabumulih Sumatera Selatan Indonesia Telpon 0713-3224417 Fax 0713-322418
e-mail : mail@unpra.ac.id

FAKULTAS ILMU KOMPUTER
UNIVERSITAS PRABUMULIH

PENILAIAN REVIEW SKRIPSI

Nama	: Alfi Syahrin Al Qodar
NIM	: 2022230038
Judul Skripsi	: Pengembangan *Google Ml Kit* di *Flutter* Deteksi Buah dan Sayuran
Hari/Tanggal	: Selasa 28-Juli 2026
Saran dan masukan	:

Lanjutkan ke dw pra penguji
segera lakukan Revisian.

2 September 2026

Rekomendasi*:

a. Dilanjutkan tanpa perbaikan
b. Dilanjutkan dengan perbaikan
c. Ditolak

Prabumulih, 28 Juli 2026
Pembimbing II

(Hermizahadiwidastra, S.Kom., M.Kom.)
NUPTK. 2863762663130192

*pilih salah satu

Jalan Patra No.50 RT.01 RW.03 Kelurahan Sukaraja Kecamatan Prabumulih Selatan
Kota Prabumulih Sumatera Selatan Indonesia Telpon 0713-3224417 Fax 0713-322418
e-mail : mail@unpra.ac.id

FAKULTAS ILMU KOMPUTER
UNIVERSITAS PRABUMULIH

PENILAIAN REVIEW SKRIPSI

Nama	: Alfi Syahrin Al Qodar
NIM	: 2022230038
Judul Skripsi	: Pengembangan *Google ML Kit* di *Flutter* untuk Deteksi Buah dan Sayuran
Hari/Tanggal	: Selasa, 28-7-25
Saran dan masukan	:

- Daftar pustaka.
- perbaiki penulisan
- Untuk Nutrisinya.

Rekomendasi*:

a. Dilanjutkan tanpa perbaikan
b. Dilanjutkan dengan perbaikan
c. Ditolak

Prabumulih, 28-7-2026
Dosen Penguji

(Muchlis, S.Kom., M.Si)
NUPTK. 5339756658200053

*pilih salah satu