

LAMPIRAN

Lampiran 1. Surat Kesepakatan Bimbingan Skripsi

SURAT KESEPAKATAN BIMBINGAN TUGAS AKHIR

Kami yang bertanda tangan di bawah ini :

Pihak Pertama

Nama : Berliani Risqi Dwi Saputri
NIM : 22090124
Program Studi : DIV Teknik Informatika

Pihak Kedua

Nama : Dwi Intan Af'idah, S.T., M.Kom.
Status : Dosen
NIDN : 0620089203
Jabatan Fungsional : Lektor
Pangkat/Golongan : Penata Muda Tk.I – III/b

Pada hari ini Senin tanggal 6 April 2026 telah terjadi sebuah kesepakatan bahwa Pihak Kedua bersedia menjadi Pembimbing I Skripsi Pihak Pertama dengan syarat adanya peningkatan progress dari proyek skripsi yang dilaporkan setiap satu pekan. Jika Pihak Pertama tidak memenuhi syarat tersebut, maka Pihak Kedua berhak memutuskan kesepakatan ini. Adapun waktu dan tempat pelaksanaan disepakati antar pihak.

Demikian kesepakatan ini dibuat dengan penuh kesadaran guna kelancaran penyelesaian Skripsi.

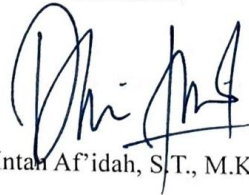
Tegal, 6 April 2026

Pihak Pertama



Berliani Risqi Dwi Saputri

Pihak Kedua



Dwi Intan Af'idah, S.T., M.Kom.

Mengetahui
Ketua Program Studi D IV Teknik Informatika



Dyah Apriyanti, S.T., M.Kom

NIPY. 09.015.225

SURAT KESEPAKATAN BIMBINGAN TUGAS AKHIR

Kami yang bertanda tangan di bawah ini :

Pihak Pertama

Nama : Berliani Risqi Dwi Saputri
NIM : 22090124
Program Studi : DIV Teknik Informatika

Pihak Kedua

Nama : Arif Hidayah, S.Tr.Kom., M.Tr.Kom
Status : Dosen / Praktisi
NUPTK : 2154773674130252
Jabatan Fungsional : -
Pangkat/Golongan : -

Pada hari ini Kamis tanggal 12 Maret 2026 telah dibuat kesepakatan antara **Pihak Pertama (Mahasiswa)** dan **Pihak Kedua (Dosen Pembimbing)** bahwa Pihak Kedua bersedia menjadi Pembimbing II Tugas Akhir bagi Pihak Pertama.

Sebagai bagian dari proses penyelesaian Tugas Akhir, Pihak **Pertama wajib melakukan bimbingan secara rutin sebanyak 1 (satu) kali dalam setiap minggu** kepada Pihak Kedua. Bimbingan tersebut **tidak dapat dirapel** atau **digabungkan** pada minggu berikutnya apabila pada minggu berjalan tidak dilaksanakan.

Setiap kegiatan bimbingan wajib disertai dengan **presentasi perkembangan Tugas Akhir** yang telah dipersiapkan oleh Pihak Pertama sesuai dengan ketentuan yang telah ditetapkan.

Selain itu, selama proses penyusunan Tugas Akhir, **Pihak Pertama wajib melaksanakan bimbingan dengan jumlah minimal 10 (sepuluh) kali pertemuan** yang tercatat sebagai bukti proses pembimbingan.

Adapun waktu dan tempat pelaksanaan bimbingan disepakati bersama oleh kedua belah pihak.

Demikian kesepakatan ini dibuat dengan penuh kesadaran dan tanggung jawab oleh kedua belah pihak guna mendukung kelancaran proses penyelesaian Tugas Akhir.

Tegal, 12 Maret 2026

Pihak Pertama



Berliani Risqi Dwi Saputri

Pihak Kedua



Arif Hidayah, S.Tr.Kom., M.Tr.Kom

Mengetahui

Ketua Program Studi D IV Teknik Informatika



Dyah Apriliani, S.T., M.Kom

NIPY. 09.015.225

Lampiran 2 Surat Pernyataan Pengajuan HKI

SURAT PERNYATAAN

Yang bertanda tangan di bawah ini, pemegang hak cipta:

- 1 Nama : Berliani Risqi Dwi Saputri
Kewarganegaraan : Indonesia
Alamat : Jalan Sumodirjo, RT 002/RW 003, Kecamatan Tegal Barat,
Kelurahan Kemandungan, Provinsi Jawa Tengah, 52114.
- 2 Nama : Dwi Intan Af'idah, S.T., M.Kom
Kewarganegaraan : Indonesia
Alamat : Grinting RT03/RW01 No.45, Kecamatan Bulakamba,
Kabupaten Brebes, Provinsi Jawa Tengah, 52253.
- 3 Nama : Arif Hidayah
Kewarganegaraan : Indonesia
Alamat : Jl. Bulu Emas No. 8, Desa Lawatan, Kecamatan Dukuhturi,
Kabupaten Tegal, Provinsi Jawa Tengah, 52192.

Dengan ini menyatakan bahwa:

1. Karya Cipta yang saya mohonkan:
Berupa : Program Komputer
Berjudul : Aplikasi Mobile Cerdas untuk Asistensi Visual Tunanetra Berbasis Computer Vision
 - Tidak meniru dan tidak sama secara esensial dengan Karya Cipta milik pihak lain atau obyek kekayaan intelektual lainnya sebagaimana dimaksud dalam Pasal 68 ayat (2);
 - Bukan merupakan Ekspresi Budaya Tradisional sebagaimana dimaksud dalam Pasal 38;
 - Bukan merupakan Ciptaan yang tidak diketahui penciptanya sebagaimana dimaksud dalam Pasal 39;
 - Bukan merupakan hasil karya yang tidak dilindungi Hak Cipta sebagaimana dimaksud dalam Pasal 41 dan 42;
 - Bukan merupakan Ciptaan seni lukis yang berupa logo atau tanda pembeda yang digunakan sebagai merek dalam perdagangan barang/jasa atau digunakan sebagai lambang organisasi, badan usaha, atau badan hukum sebagaimana dimaksud dalam Pasal 65 dan;
 - Bukan merupakan Ciptaan yang melanggar norma agama, norma susila, ketertiban umum, pertahanan dan keamanan negara atau melanggar peraturan perundang-undangan sebagaimana dimaksud dalam Pasal 74 ayat (1) huruf d Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.
2. Sebagai pemohon mempunyai kewajiban untuk menyimpan asli contoh ciptaan yang dimohonkan dan harus memberikan apabila dibutuhkan untuk kepentingan penyelesaian sengketa perdata maupun pidana sesuai dengan ketentuan perundang-undangan.
3. Karya Cipta yang saya mohonkan pada Angka 1 tersebut di atas tidak pernah dan tidak sedang dalam sengketa pidana dan/atau perdata di Pengadilan.

4. Dalam hal ketentuan sebagaimana dimaksud dalam Angka 1 dan Angka 3 tersebut di atas saya / kami langgar, maka saya / kami bersedia secara sukarela bahwa:
- permohonan karya cipta yang saya ajukan dianggap ditarik kembali; atau
 - Karya Cipta yang telah terdaftar dalam Daftar Umum Ciptaan Direktorat Hak Cipta, Direktorat Jenderal Hak Kekayaan Intelektual, Kementerian Hukum Dan Hak Asasi Manusia R.I dihapuskan sesuai dengan ketentuan perundang-undangan yang berlaku.
 - Dalam hal kepemilikan Hak Cipta yang dimohonkan secara elektronik sedang dalam berperkara dan/atau sedang dalam gugatan di Pengadilan maka status kepemilikan surat pencatatan elektronik tersebut ditangguhkan menunggu putusan Pengadilan yang berkekuatan hukum tetap.

Demikian Surat pernyataan ini saya/kami buat dengan sebenarnya dan untuk dipergunakan sebagaimana mestinya.

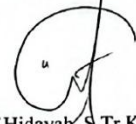
Tegal, 3 Juni 2026



Berliani Risqi Dwi Saputri
Pemegang Hak Cipta*



Dwi Intan Af'idah, S.T., M.Kom.
Pemegang Hak Cipta*



Arif Hidayah, S.Tr.Kom, M.Tr
Pemegang Hak Cipta*

* Semua pemegang hak cipta agar menandatangani di atas materai (untuk penulis pertama)

Lampiran 3 Surat Pengalihan Hak Cipta

SURAT PENGALIHAN HAK CIPTA

Yang bertanda tangan di bawah ini :

Adalah **Pihak I** selaku pencipta, dengan ini menyerahkan karya ciptaan saya kepada :

- 1 Nama : Berliani Risqi Dwi Saputri
Kewarganegaraan : Indonesia
Alamat : Jalan Sumodirjo, RT 002/RW 003, Kelurahan Kemandungan,
Kecamatan Tegal Barat, Kota Tegal, 52114.
- 2 Nama : Dwi Intan Afidah, S.T., M.Kom
Kewarganegaraan : Indonesia
Alamat : Grinting RT03/RW01 No.45, Kecamatan Bulakamba,
Kabupaten Brebes, Provinsi Jawa Tengah, 52253.
- 3 Nama : Arif Hidayah, S.Tr.Kom., M.Tr.Kom
Kewarganegaraan : Indonesia
Alamat : Jl. Bulu Emas No. 8, Desa Lawatan, Kecamatan Dukuhturi,
Kabupaten Tegal, Provinsi Jawa Tengah, 52192.

Adalah **Pihak II** selaku penerima, dengan ini menyerahkan karya ciptaan saya kepada :

- Nama : Pusat Penelitian dan Pengabdian Masyarakat (P3M) Universitas Harkat
Negeri
Alamat : Jl. Mataram No. 9 Pesurungan Lor Kota Tegal

Adalah **Pihak II** selaku Pemegang Hak Cipta berupa Program Komputer dengan judul "Aplikasi Mobile Cerdas untuk Asistensi Tunanetra Berbasis Computer Vision". untuk didaftarkan di Direktorat Hak Cipta dan Desain Industri, Direktorat Jenderal Kekayaan Intelektual, Kementerian Hukum dan Hak Asasi Manusia Republik Indonesia.

Demikianlah surat pengalihan hak ini kami buat, agar dapat dipergunakan sebagaimana mestinya.

Pemegang Hak Cipta

Kepala P3M

 Universitas
Harkat Negeri

(Sharfina Febbi Handayani, M.Kom)

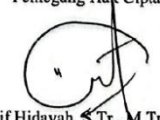
Tegal, 3 Juni 2026

Pencipta


Berliani Risqi Dwi Saputri
Pencipta

Pemegang Hak Cipta*


(Dwi Intan Afidah, S.T., M.Kom)
Pemegang Hak Cipta*


(Arif Hidayah, S.Tr., M.Tr.Kom)
Pemegang Hak Cipta*

Lampiran 4 Manual Book



MANUAL BOOK APLIKASI TUNTUN

Aplikasi Mobile Cerdas untuk Asistensi Visual
Tunanetra Berbasis Computer Vision

Oleh:
Berliani Risqi Dwi Saputri - 22090124

DAFTAR ISI

1. PENDAHULUAN	3
1.1 Tujuan Dokumen.....	3
1.2 Deskripsi Umum Sistem	3
1.3 Deskripsi Dokumen.....	4
2. KEBUTUHAN PERANGKAT	4
2.1 Perangkat Keras	4
2.2 Perangkat Lunak.....	5
3. PENGENALAN KONSEP DASAR.....	5
3.1 Cara Aplikasi "Melihat"	5
3.2 Cara Berinteraksi dengan Aplikasi.....	6
3.3 Sistem Suara Dua Arah	6
3.4 Asisten Suara Global.....	6
4. CARA MENGGUNAKAN APLIKASI	7
4.1 Instalasi Aplikasi.....	7
4.2 Kalibrasi Kamera (Wajib Dilakukan Pertama Kali)	7
4.3 Tampilan Utama dan Navigasi Antar Fitur.....	9
4.4 Mode Sekitar (Deteksi Lingkungan).....	10
4.5 Mode Pencarian Objek.....	11
4.6 Mode Navigasi (Berjalan & Kendaraan).....	14
4.8 Mode Deteksi Uang.....	17
4.7 Mode Baca Teks.....	19
4.9 Mode Analisis Pakaian.....	22
RINGKASAN SEMUA FITUR	24

1. PENDAHULUAN

1.1 Tujuan Dokumen

Dokumen ini merupakan panduan penggunaan aplikasi TunTun.

Tujuan utama dokumen ini adalah:

- Menggambarkan dan menjelaskan penggunaan aplikasi TunTun.
- Memberikan panduan langkah demi langkah yang jelas, mulai dari instalasi hingga penggunaan seluruh fitur.

1.2 Deskripsi Umum Sistem

TunTun merupakan aplikasi smartphone berbasis kecerdasan buatan (*Artificial Intelligence / AI*) yang dirancang khusus sebagai asisten navigasi dan pengenalan lingkungan bagi penyandang tunanetra. Aplikasi TunTun di desain dengan UI yang mudah dipahami pengguna dan fitur-fitur di dalamnya dibuat untuk menyesuaikan kebutuhan sehari-hari berdasarkan hasil observasi yang telah dilakukan. Fitur pada aplikasi TunTun dapat dilihat pada tabel dibawah ini.

No	Fitur	Fungsi Singkat
1	Kalibrasi	Mengatur kamera ponsel agar mendapatkan nilai <i>focal length</i> sehingga dapat mengukur jarak objek dari kamera/pengguna berada.
2	Mode Sekitar	Mendeteksi dan mendeskripsikan benda-benda di sekitar pengguna secara otomatis, termasuk deskripsi cerdas yang menggunakan bantuan AI.
3	Mode Pencarian Objek	Membantu menemukan benda spesifik (contoh: kursi, pintu, botol) dengan panduan suara arah dan jarak
4	Mode Navigasi	Memandu perjalanan ke suatu tujuan dengan dua pilihan: jalan kaki atau naik kendaraan .
5	Mode Deteksi Uang	Mengenali dan menyebutkan nominal uang kertas Rupiah secara otomatis.
6	Mode Baca Teks	Membacakan teks dari media yang pengguna inginkan ataupun dari foto galeri.
7	Mode Analisis Pakaian	Mendeskripsikan pakaian yang sedang dikenakan atau akan dipakai, dan menjawab pertanyaan tentang pakaian tersebut.

Frequently Ask Questions

- TunTun dirancang untuk siapa?

Tunanetra total maupun low vision yang membutuhkan bantuan dalam mengenali lingkungan.

- Bagaimana TunTun Berbicara dengan Pengguna?

Seluruh umpan balik diberikan melalui suara (*Text-to-Speech*) dan bantuan *Talkback* yang ada pada sistem smartphone. Aplikasi akan berbicara secara otomatis tanpa pengguna perlu menyentuh atau melihat layar. Pengguna juga bisa memberikan perintah menggunakan suara pada fitur-fitur tertentu.

1.3 Deskripsi Dokumen

Dokumen ini terdiri dari 4 bagian utama:

Bagian	Isi
Bab 1 - Pendahuluan	Gambaran umum aplikasi dan tujuan penulisan dokumen
Bab 2 - Kebutuhan Perangkat	Spesifikasi yang dibutuhkan agar aplikasi berjalan dengan baik
Bab 3 - Pengenalan Konsep Dasar	Penjelasan cara kerja teknologi utama yang digunakan aplikasi
Bab 4 - Cara Menggunakan	Panduan langkah demi langkah penggunaan seluruh fitur

2. KEBUTUHAN PERANGKAT

2.1 Perangkat Keras

Untuk menjalankan aplikasi TunTun dengan optimal, ponsel yang digunakan harus memenuhi spesifikasi berikut:

Komponen	Persyaratan Minimum	Rekomendasi
Sistem Operasi	<i>Android 8.0 (Oreo)</i>	Android 11 ke atas
Prosesor	<i>Prosesor 8</i>	700-series ke atas
RAM	3 GB	4 GB atau lebih
Penyimpanan	500 MB tersedia	1 GB tersedia
Kamera Belakang	Resolusi 8 Megapiksel	13 Megapiksel atau lebih
GPS	Wajib ada	-
Sensor Kompas	Wajib ada	-
Koneksi Internet	Diperlukan untuk fitur AI	WiFi atau data seluler minimal 4G
Speaker / Earphone	Wajib ada	<i>Earphone</i> sangat disarankan untuk privasi

❖ **Catatan Penting:** Fitur **Deskripsi Cerdas AI** (pada Mode Sekitar), **Baca Teks** dan **Analisis Pakaian** memerlukan koneksi internet aktif karena menggunakan

layanan AI berbasis cloud. Fitur lain seperti deteksi objek, navigasi GPS, dan deteksi uang dapat berjalan tanpa internet setelah pertama kali diunduh.

2.2 Perangkat Lunak

Perangkat Lunak	Keterangan
Android 8.0 atau lebih baru	Sistem operasi ponsel
TalkBack (opsional)	Fitur bawaan Android untuk aksesibilitas. Jika diaktifkan, TunTun akan menyesuaikan cara interaksinya secara otomatis
Izin Aplikasi yang Diperlukan:	
– Kamera	Untuk semua fitur pendeteksian visual
– Mikrofon	Untuk fitur perintah suara
– Lokasi (GPS)	Untuk fitur navigasi
– Penyimpanan	Untuk menyimpan data kalibrasi dan riwayat
– Internet	Untuk fitur AI berbasis cloud

❖ Cara Mengaktifkan *TalkBack* (opsional):
 Buka Pengaturan → Aksesibilitas → *TalkBack* → Aktifkan.
 Dengan *TalkBack* aktif, pengguna cukup melakukan **ketuk dua kali** untuk menekan tombol.

3. PENGENALAN KONSEP DASAR

Sebelum menggunakan aplikasi, penting untuk memahami beberapa konsep dasar berikut ini untuk membantu pengguna menggunakan TunTun dengan lebih efektif.

3.1 Cara Aplikasi "Melihat"

TunTun menggunakan kamera ponsel sebagai "mata". Kamera merekam gambar secara terus-menerus, lalu gambar tersebut dianalisis oleh teknologi kecerdasan buatan (AI) yang berjalan di dalam ponsel. Pada aplikasi ini teknologi yang digunakan yaitu **YOLO** (*You Only Look Once*), sebuah sistem pengenalan gambar yang mampu mengidentifikasi puluhan jenis benda (seperti orang, kursi, pintu, botol, dll.) dalam hitungan milidetik.

Hasil dari deteksi tersebut akan langsung diubah menjadi **informasi suara** yang disampaikan kepada pengguna.

Kamera merekam → AI menganalisis → Informasi diubah jadi suara → Pengguna mendengar

3.2 Cara Berinteraksi dengan Aplikasi

TunTun dirancang untuk digunakan **tanpa harus melihat layar**. Ada dua cara utama berinteraksi:

A. Sentuhan Layar

Gerakan	Fungsi
Ketuk satu kali	Menekan tombol (tanpa <i>TalkBack</i>).
Ketuk dua kali	Menekan tombol (dengan <i>TalkBack</i> aktif).
Tahan lama (<i>Long Press</i>)	Fungsi khusus tergantung mode.
Geser ke bawah (<i>Swipe Down</i>)	Menutup panel hasil deskripsi.
Tahan dua jari	Mengaktifkan Asisten Suara Global.

B. Perintah Suara

Pada beberapa fitur, pengguna dapat mengucapkan perintah. Aplikasi akan mendengarkan, memproses, dan merespons secara lisan.

3.3 Sistem Suara Dua Arah

TunTun memiliki sistem suara dua arah:

1. **Suara Keluar (*Text-to-Speech* / TTS):** Aplikasi berbicara untuk memberi informasi kepada pengguna. Misalnya: "*Kursi terdeteksi di sebelah kiri, berjarak kurang lebih 1 meter.*"
2. **Suara Masuk (*Speech-to-Text* / STT):** Aplikasi mendengarkan perintah suara dari pengguna. Misalnya, pengguna mengucapkan "*kursi*" untuk mencari kursi.

3.4 Asisten Suara Global

TunTun memiliki fitur **Asisten Suara Global** yang bisa diakses dari fitur mana pun tanpa perlu menavigasi menu. Asisten ini memungkinkan pengguna berpindah antar fitur hanya dengan suara.

Cara Mengaktifkan:

- **Tanpa TalkBack:** Tahan layar dengan **dua jari** selama kurang lebih 1 detik
- **Dengan TalkBack aktif:** Tahan layar dengan **satu jari** (*long press*)

Setelah asisten aktif, ucapkan salah satu kata kunci berikut:

Kata Kunci yang Diucapkan	Fitur yang Dibuka
"Sekitar"	Mode Sekitar
"Cari"	Mode Pencarian Objek
"Jalan"	Mode Navigasi
"Uang"	Mode Deteksi Uang
"Baca"	Mode Membaca Teks
"Pakaian"	Mode Analisis Pakaian

4. CARA MENGGUNAKAN APLIKASI

4.1 Instalasi Aplikasi

Langkah-langkah:

1. Unduh file instalasi (.apk) dari sumber yang telah disediakan oleh pengembang.
2. Buka file .apk tersebut di ponsel. Jika ada peringatan "Sumber tidak dikenal", izinkan instalasi dari sumber tidak dikenal melalui pengaturan.
3. Ikuti instruksi instalasi yang muncul di layar.
4. Setelah terpasang, buka aplikasi TunTun.
5. Saat pertama kali dibuka, aplikasi akan meminta beberapa **izin** (kamera, mikrofon, lokasi, dan penyimpanan). **Pilih "Izinkan" untuk semua permintaan** agar aplikasi dapat berfungsi dengan baik.

4.2 Kalibrasi Kamera (Wajib Dilakukan Pertama Kali)

Apa itu Kalibrasi?

Kalibrasi adalah proses "mengajari" kamera ponsel tentang cara mengukur jarak. Setiap ponsel memiliki kamera dengan ukuran dan kemampuan yang berbeda. Proses kalibrasi memastikan TunTun dapat memberikan informasi jarak yang **akurat dan sesuai dengan kamera ponsel Anda**. Kalibrasi hanya perlu dilakukan **satu kali** saja. Setelah selesai, data kalibrasi tersimpan permanen di ponsel.

- ❖ **Apa yang Dibutuhkan untuk Kalibrasi?**
Sebuah kartu berukuran standar, seperti **KTP, SIM, kartu ATM, atau kartu pelajar**.

Langkah-langkah Kalibrasi:

Langkah 1 – Dengarkan Instruksi Saat layar kalibrasi pertama kali muncul, aplikasi akan berbicara:

"Kamera siap. Pegang KTP atau kartu ATM anda di depan kamera pada jarak 20 sentimeter, atau kira-kira sepanjang satu jengkal tangan. Tahan posisi tersebut sampai sistem selesai mengukur."

Langkah 2 – Siapkan Kartu Pegang KTP atau kartu ATM Anda dengan satu tangan.

Langkah 3 – Posisikan Kartu di Depan Kamera

- Arahkan sisi belakang ponsel (kamera) ke arah kartu.
- Posisikan kartu pada jarak ± 20 cm dari kamera — kira-kira sepanjang satu jengkal tangan.
- Usahakan kartu berada di **tengah-tengah pandangan kamera** dengan cahaya yang cukup terang.

Langkah 4 – Tahan Posisi dan Tunggu Tahan kartu di posisi tersebut. Aplikasi akan mulai mengukur dan memberikan umpan balik suara:

- "25 persen" → Sudah seperempat selesai
- "50 persen" → Sudah setengah selesai
- "75 persen" → Hampir selesai

Langkah 5 – Kalibrasi Selesai Saat aplikasi mengucapkan "*Kalibrasi selesai. Mengalihkan ke menu utama.*", kalibrasi berhasil dan aplikasi otomatis membuka halaman utama.

Pesan Suara Selama Kalibrasi dan Artinya:

Pesan Suara	Arti	Yang Harus Dilakukan
"Kartu belum terlihat jelas..."	Kartu tidak terdeteksi	Pastikan cahaya cukup dan kartu menghadap kamera
"Kartu terlihat, cari cahaya lebih terang."	Terdeteksi tapi kurang jelas	Cari tempat lebih terang
"Kartu bergerak. Tahan posisi di jarak satu jengkal."	Kartu bergerak terlalu banyak	Pegang lebih stabil
"Jarak berubah terlalu jauh. Perhitungan diulang dari nol."	Jarak berubah drastis	Mulai ulang dari jarak satu jengkal

Sebelum menggunakan aplikasi, Kamu bisa lakukan proses kalibrasi terlebih dahulu.



Mulai Kalibrasi



4.3 Tampilan Utama dan Navigasi Antar Fitur

Setelah kalibrasi selesai, pengguna akan masuk ke **halaman utama** aplikasi. Di bagian paling bawah layar terdapat **menu navigasi** yang berisi 6 fitur utama:

	Nama Fitur
1	Sekitar
2	Cari Objek
3	Navigasi Jalan
4	Deteksi Uang
5	Baca Teks
6	Analisis Pakaian



Sekitar



Cari



Jalan



Uang



Baca



Pakaian

Cara berpindah fitur:

- **Cara 1:** Sentuh ikon fitur yang diinginkan di bagian bawah layar.
- **Cara 2:** Gunakan Talkback lalu klik dua kali untuk berpindah fitur atau bisa juga

menggunakan swipe, atau

- Gunakan **Asisten Suara Global** (Bagian 3.4) untuk berpindah fitur dengan ucapan
- ❖ **Tips:** Setiap fitur memiliki **tombol bantuan (ikon tanda tanya ?)** di pojok kanan atas layar. Tekan tombol ini untuk mendengarkan panduan singkat cara menggunakan fitur tersebut.

4.4 Mode Sekitar (Deteksi Lingkungan)

Mode Sekitar menggunakan kamera secara terus-menerus untuk mendeteksi dan mengidentifikasi benda-benda di sekitar pengguna seperti orang, kursi, meja, pintu, kendaraan, dan lain-lain. Mode ini juga dilengkapi fitur **Deskripsi Cerdas AI** yang dapat mendeskripsikan seluruh ruangan atau lingkungan secara lengkap menggunakan kecerdasan buatan.

Cara Menggunakan:

Langkah 1 – Masuk ke Mode Sekitar

- Ketuk ikon pertama di menu bawah (ikon "Sekitar"), atau
- Ucapkan "*Sekitar*" melalui Asisten Suara Global
- ❖ Saat pertama kali masuk, aplikasi akan memberikan tutorial singkat: "*Mode Sekitar. Kamera akan otomatis membacakan nama dan jarak benda. Untuk mendeskripsikan ruangan secara detail, raba bagian bawah layar lalu ketuk dua kali. Untuk menutup hasil deskripsi, ketuk dua kali di bagian atas layar atau geser ke bawah.*"

Langkah 2 – Arahkan Kamera Pegang ponsel dengan kamera mengarah ke depan, lalu arahkan perlahan ke berbagai arah. Aplikasi secara otomatis mendeteksi benda yang masuk ke bidang pandang kamera.

Langkah 3 – Dengarkan Informasi Aplikasi akan secara otomatis menyebutkan nama benda-benda yang terdeteksi.

Langkah 4 – Gunakan Fitur Deskripsi Cerdas AI (Opsional)

Untuk mendapatkan deskripsi lengkap seluruh ruangan:

1. Tekan tombol "**Deskripsikan Ruangan**" di bagian bawah layar (atau ketuk dua kali jika TalkBack aktif)
2. Tunggu beberapa detik sementara AI menganalisis (akan terdengar suara tunggu dan tampilan layar akan menggelapkan sejenak)
3. Aplikasi akan membacakan deskripsi lengkap ruangan — termasuk benda-benda yang ada, tata letaknya, dan informasi lain yang terlihat

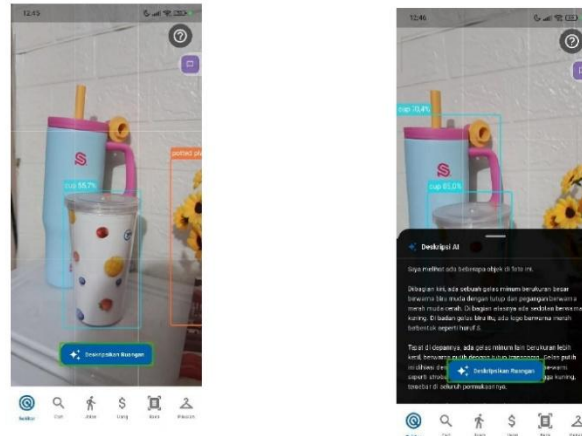
Memahami Panel Deskripsi AI:

- Hasil deskripsi muncul sebagai panel di bagian bawah layar
- Panel dapat digeser ke bawah (swipe down) untuk menutup, atau ketuk bagian mana pun di luar panel

- Aplikasi juga **otomatis membacakan** hasil deskripsi secara lisan

Tombol Bantuan:

- Tekan tombol ? di pojok kanan atas untuk mendengar ulang panduan fitur ini



4.5 Mode Pencarian Objek

Mode Pencarian Objek membantu pengguna menemukan satu benda spesifik di dalam ruangan atau lingkungan sekitar. Setelah benda dipilih, aplikasi akan memandu pengguna menuju benda tersebut menggunakan instruksi suara arah (kiri/depan/kanan) dan perkiraan jarak.

- ❖ Saat pertama kali masuk, aplikasi akan memberikan tutorial singkat: *"Selamat datang di Mode Pencarian. Di sini Anda bisa mencari letak suatu benda. Raba bagian bawah layar dan ketuk dua kali tombol mikrofon untuk mencari dengan suara, atau pilih dari daftar yang tersedia."*

Cara Menggunakan:

Tahap 1: Memilih Benda yang Dicari

Ada dua cara memilih benda:

Cara A – Menggunakan Suara (Direkomendasikan):

1. Tekan tombol **mikrofon (ikon mic)** di bagian bawah layar
2. Tunggu aplikasi berkata *"Ucapkan benda yang dicari."*
3. Ucapkan nama benda dengan jelas, contoh: *"kursi"*, *"botol"*, *"pintu"*
4. Aplikasi akan mengonfirmasi dan mulai pencarian

Cara B – Memilih dari Daftar:

1. Di halaman Mode Pencarian, sudah tersedia daftar benda yang dapat dicari
2. Geser layar untuk menelusuri daftar
3. Ketuk (atau ketuk dua kali dengan TalkBack) nama benda yang ingin dicari
4. Tandai bintang/favorit ★ pada benda yang sering dicari agar muncul di urutan paling atas



Tahap 2: Proses Pencarian

Setelah benda dipilih, kamera akan aktif. Pengguna diminta menggerakkan diri berpindah posisi perlahan sambil mengarahkan kamera ke berbagai arah.

Selama pencarian, aplikasi akan memberikan umpan balik:

- *"Belum terlihat. Terus gerakkan kamera perlahan."* — Benda belum terdeteksi, terus berputar
- *"[Nama benda] ditemukan! Di sebelah kiri Anda, berjarak sekitar 2 meter. Gerak mendekat."* — Benda terdeteksi, mulai mendekat

Tahap 3: Panduan Menuju Benda

Saat benda sudah terdeteksi, sistem masuk ke fase panduan dengan tiga tingkat:

Fase	Kondisi	Frekuensi Pembaruan Suara
Jauh	Jarak > 1 meter	Setiap 5 detik
Dekat	Jarak $\leq$ 1 meter	Setiap 3 detik
Tiba	Sangat dekat / benda memenuhi layar	Satu kali + getaran ponsel

Informasi yang diberikan saat panduan aktif:

- **Posisi:** "di sebelah kiri", "di depan", atau "di sebelah kanan"
- **Jarak:** Perkiraan dalam meter atau sentimeter
- **Perubahan posisi:** "Objek bergeser ke kanan" jika benda berpindah

Tahap 4: Tiba di Tujuan

Saat benda sangat dekat, aplikasi akan:

1. Memberikan getaran panjang sebagai konfirmasi
2. Mengucapkan: "[Nama benda] ada tepat di depan Anda. Raih sekarang!"
3. Pencarian otomatis berhenti

Tombol Kontrol Pencarian:

- **Ganti Benda:** Tombol di bawah layar saat kamera aktif untuk kembali ke daftar dan memilih benda lain
- **Hentikan Pencarian:** Tombol untuk menghentikan pencarian yang sedang berjalan



4.6 Mode Navigasi (Berjalan & Kendaraan)

Mode Navigasi memandu pengguna dari lokasi saat ini menuju suatu tujuan yang ditentukan. Terdapat **dua mode perjalanan** yang bisa dipilih:

Mode	Keterangan
Jalan Kaki	Kamera aktif mendeteksi rintangan di depan, sambil GPS memandu arah. Cocok untuk berjalan di area pejalan kaki.
Kendaraan	Kamera dimatikan untuk menghemat baterai. Navigasi GPS tetap aktif. Cocok saat berada di dalam kendaraan.

Cara Menggunakan:

Langkah 1 – Masuk ke Mode Navigasi

- Ketuk ikon ketiga di menu bawah (ikon "Navigasi Jalan"), atau
- Ucapkan "*Jalan*" melalui Asisten Suara Global
- ❖ Saat pertama kali masuk, aplikasi akan memberikan tutorial singkat: "*Selamat datang di Mode Navigasi Jalan. Arahkan ponsel ke depan dada. Kamera akan memperingatkan rintangan dan jalur aman. Gunakan panel di bawah layar untuk info lokasi, cari tujuan dan mengganti mode berjalan atau menaiki kendaraan. Peringatan Penting: Aplikasi ini hanya alat bantu. Selalu gunakan tongkat putih Anda untuk keselamatan.*"

Langkah 2 – Pilih Tujuan

Tekan tombol "**CARI TUJUAN**" di bagian bawah layar. Sebuah panel akan muncul dengan beberapa pilihan:

A. Pilih Mode Perjalanan Terlebih Dahulu:

- Sentuh tombol "**Jalan Kaki**" atau "**Kendaraan**" sesuai situasi
- Aplikasi akan mengonfirmasi: "*Mode diubah ke Jalan Kaki*" atau "*Mode diubah ke Kendaraan*".

B. Tentukan Tujuan:

Cara 1 – Dengan Suara (Direkomendasikan):

1. Ketuk tombol **mikrofon bulat besar** di tengah panel
2. Tunggu hingga ikon berubah merah (sedang mendengarkan)
3. Ucapkan nama tempat tujuan, contoh: "*Rumah Sakit Hasan Sadikin*", "*Masjid Istiqlal*", "*Stasiun Gambir*"
4. Aplikasi akan mencari dan memulai navigasi

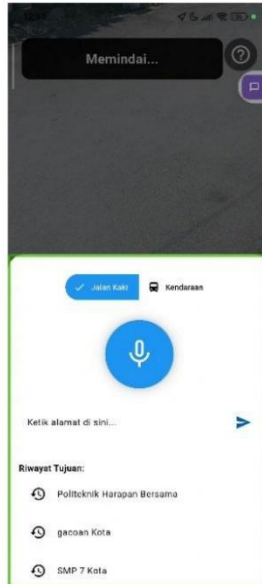
Cara 2 – Dengan Ketikan Manual:

1. Ketuk kolom teks "*Ketik alamat di sini...*"
2. Ketikkan tujuan menggunakan keyboard

3. Tekan tombol kirim (ikon panah) atau tekan Enter

Cara 3 – Dari Riwayat:

- Pilih tujuan yang pernah dikunjungi dari daftar **Riwayat Tujuan** di bawah panel



Langkah 3 – Selama Navigasi Berlangsung

Pada Mode Jalan Kaki:

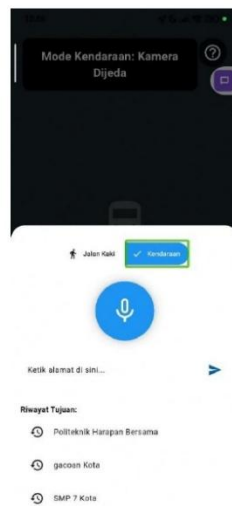
- Kamera aktif dan mendeteksi rintangan di depan secara real-time
- Warna latar layar berubah sebagai indikator:
 - **Hijau:** Jalur aman, bisa jalan terus
 - **Kuning:** Hati-hati, ada rintangan
 - **Merah:** Bahaya, berhenti atau hindari

- GPS memberikan panduan arah: "*Belok kiri setelah 50 meter*", "*Terus lurus 200 meter*", dll.
- Panel peta kecil muncul di pojok kanan bawah, menampilkan posisi pengguna dan sisa jarak



Pada Mode Kendaraan:

- Layar menampilkan ikon bus dan keterangan "*Mode Kendaraan: Kamera Dijeda*"
- Navigasi GPS tetap aktif dan memberikan panduan suara



Langkah 4 – Mengecek Status Navigasi

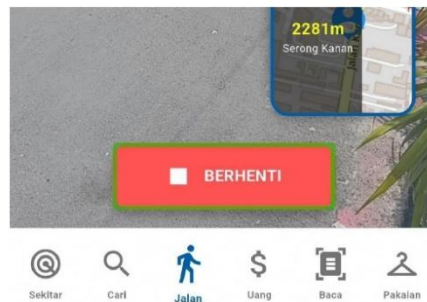
Kapan saja selama navigasi, **ketuk panel peta kecil** di pojok kanan bawah untuk mendengar:

- Lokasi Anda saat ini
- Sisa jarak ke tujuan
- Instruksi arah berikutnya



Langkah 5 – Menghentikan Navigasi

- Tekan tombol merah "**BERHENTI**" yang muncul di bawah layar, atau
- Tahan (long press) panel peta kecil di pojok kanan bawah



4.8 Mode Deteksi Uang

Mode Deteksi Uang secara otomatis mengenali dan menyebutkan nominal uang kertas Rupiah yang diarahkan ke kamera. Pengguna tidak perlu menekan tombol apapun — cukup arahkan uang ke kamera dan aplikasi langsung berbicara.

Cara Menggunakan:

Langkah 1 – Masuk ke Mode Deteksi Uang

- Ketuk ikon keempat di menu bawah (ikon "Uang"), atau
- Ucapkan "Uang" melalui Asisten Suara Global
- ❖ Saat pertama kali masuk, aplikasi akan memberikan tutorial singkat: *"Selamat datang di Mode Deteksi Uang. Arahkan kamera ke uang kertas atau koin. Aplikasi akan menjumlahkan nominalnya secara otomatis. Pastikan cahaya cukup terang dan uang tidak terlipat parah."*

Langkah 2 – Arahkan Uang ke Kamera

- Pegang uang kertas dengan sisi salah satu muka menghadap kamera belakang ponsel
- Usahakan seluruh permukaan uang terlihat oleh kamera
- Pastikan cahaya cukup terang

Langkah 3 – Dengarkan Hasil Aplikasi akan secara otomatis menyebutkan nominal uang yang terdeteksi, misalnya: *"Lima puluh ribu rupiah"* **Tips Penggunaan Mode Deteksi Uang:**

Kondisi	Yang Dilakukan
Uang tidak terdeteksi	Coba ubah sudut kemiringan uang atau perbaiki pencahayaan
Hasil kurang tepat	Pastikan seluruh permukaan uang (bukan hanya sebagian) terlihat kamera
Cahaya kurang	Pindah ke tempat yang lebih terang atau nyalakan lampu
Uang terlipat/kusut	Luruskan terlebih dahulu agar lebih mudah terdeteksi

Tombol Bantuan:

Tekan tombol ? di pojok kanan atas untuk mendengar panduan fitur ini



4.7 Mode Baca Teks

Mode Baca Teks membantu pengguna membaca tulisan yang ada di kertas, papan, layar, kemasan produk, atau dokumen apa pun — cukup dengan mengarahkan kamera. Aplikasi secara otomatis mendeteksi ketika tulisan sudah cukup jelas, lalu mengambil foto dan membacakannya dengan suara. Setelah teks dibacakan, pengguna dapat mengajukan pertanyaan lebih lanjut tentang isi teks tersebut.

Mode ini mendukung dua cara input:

- **Kamera langsung** - arahkan kamera ke dokumen/tulisan
- **Galeri foto** - pilih foto yang sudah tersimpan di ponsel

Cara Menggunakan:

Langkah 1 – Masuk ke Mode Baca Teks

- Ketuk ikon kelima di menu bawah (ikon "Baca"), atau
- Ucapkan "Baca" melalui Asisten Suara Global
- ❖ Saat pertama kali masuk, aplikasi akan memberikan tutorial singkat: *"Selamat datang di Mode Baca Teks. Arahkan kamera ke kertas atau layar. Ponsel akan bergetar pendek jika teks terdeteksi, dan bergetar panjang jika teks berhasil difoto. Anda juga bisa menekan tombol Buka Galeri di kiri bawah untuk membaca teks dari foto tersimpan."*

Cara A: Membaca dengan Kamera

Langkah 2 – Siapkan Dokumen atau Tulisan Letakkan atau pegang dokumen (buku, surat, kemasan, papan nama, dll.) di bawah cahaya yang cukup terang.

Langkah 3 – Arahkan Kamera ke Tulisan Pergang ponsel dengan kamera belakang

menghadap ke teks. Jaga agar kamera tegak lurus (tidak miring) terhadap tulisan.

Langkah 4 – Ikuti Panduan Getaran Aplikasi secara otomatis memindai gambar dari kamera setiap saat. Perhatikan petunjuk getaran berikut:

Sinyal Getaran	Arti	Yang Harus Dilakukan
Tidak bergetar + suara "Tidak ada teks"	Tulisan tidak terdeteksi	Dekatkan kamera ke teks atau perbaiki pencahayaan
Getaran pendek (sekali)	Teks mulai terdeteksi	Tahan posisi, jangan gerak
Getaran panjang	Foto berhasil diambil otomatis	Tunggu hasil pembacaan

Langkah 5 – Tunggu Hasil Dibacakan Setelah foto diambil, aplikasi akan memproses dan membacakan isi teks. Ini membutuhkan **koneksi internet** dan mungkin memakan waktu beberapa detik.

Contoh hasil yang dibacakan: *"Teks ditemukan. Dosis: dua tablet sehari setelah makan. Simpan di tempat sejuk dan kering. Dokumen ini adalah petunjuk penggunaan obat."*

Cara B: Membaca dari Foto di Galeri

Langkah 2 – Tekan Tombol "Buka Galeri" Tekan tombol **"Buka Galeri"** (ikon foto) di bagian kiri bawah layar.

Langkah 3 – Pilih Foto Galeri foto akan terbuka. Pilih foto yang berisi tulisan yang ingin dibaca.

Langkah 4 – Tunggu Hasil Dibacakan Aplikasi akan memproses foto tersebut dan membacakan teks yang ditemukan secara otomatis.

Mengajukan Pertanyaan Tentang Teks (Fitur Tanya Jawab)

Setelah teks selesai dibacakan, pengguna dapat mengajukan pertanyaan tambahan tentang isi teks tersebut:

1. Di bagian bawah layar, tekan tombol mikrofon **"Tanya Asisten"**
2. Tunggu hingga tombol berubah merah (sedang mendengarkan)
3. Ucapkan pertanyaan Anda dengan jelas, contoh:
 - *"Apa nama obatnya?"*
 - *"Berapa harganya?"*
 - *"Kapan tanggal kedaluwarsanya?"*
 - *"Apa artinya kalimat kedua?"*
4. Tunggu 3 detik atau ketuk tombol mikrofon lagi untuk mengirim pertanyaan
5. Dengarkan jawaban dari AI

Pengguna dapat terus mengajukan pertanyaan berulang kali selama masih dalam sesi yang sama.



Ringkasan Tombol di Mode Baca Teks:

Tombol	Posisi	Fungsi
Buka Galeri (ikon foto)	Kiri bawah layar	Memilih foto dari galeri ponsel
Pindai Ulang (ikon refresh)	Kanan bawah layar / tengah	Kembali ke mode kamera untuk scan dokumen baru
Tanya Asisten (ikon mikrofon)	Bawah layar (setelah ada hasil)	Mengajukan pertanyaan tentang teks yang sudah dibaca
? (tanda tanya)	Pojok kanan atas	Mendengar ulang panduan penggunaan

Tips Penggunaan Mode Baca Teks:

Situasi	Saran
Teks tidak terdeteksi	Dekatkan kamera, perbaiki pencahayaan, pastikan tulisan tidak blur
Hasil pembacaan tidak lengkap	Pastikan seluruh teks masuk ke bidang kamera
Koneksi lambat	Proses mungkin lebih lama, tetap tunggu hingga selesai
Tulisan terlalu kecil	Dekatkan ponsel lebih ke dokumen
Huruf miring/kursif	Pastikan cahaya cukup terang agar hasilnya lebih akurat

4.9 Mode Analisis Pakaian

Mode Analisis Pakaian menggunakan kamera dan kecerdasan buatan (AI) untuk **mendeskrripsikan pakaian** yang sedang dikenakan atau akan dipakai. Pengguna juga dapat **mengajukan pertanyaan** tentang pakaian tersebut — misalnya menanyakan warna, jenis, atau apakah cocok untuk acara tertentu.

Cara Menggunakan:

Langkah 1 – Masuk ke Mode Analisis Pakaian

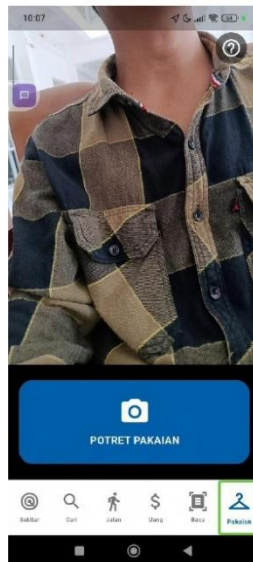
- Ketuk ikon keenam di menu bawah (ikon "Pakaian"), atau
- Ucapkan "*Pakaian*" melalui Asisten Suara Global
- ❖ Saat pertama kali masuk, aplikasi akan memberikan tutorial singkat: "*Selamat datang di Mode Pakaian. Arahkan kamera ke pakaian Anda, lalu raba bagian bawah layar dan ketuk dua kali tombol Potret. Setelah foto diambil, Anda bisa bertanya tentang pakaian tersebut.*"

Langkah 2 – Siapkan Pakaian yang Ingin Dianalisis

Ada dua cara:

- **Kenakan pakaian** dan arahkan kamera ke badan Anda, atau
- **Bentangkan pakaian** di atas permukaan datar lalu arahkan kamera ke pakaian tersebut

Langkah 3 – Potret Pakaian Tekan tombol besar "**POTRET PAKAIAN**" (dengan ikon kamera) di bagian bawah layar. Tunggu beberapa saat (mungkin 5–15 detik tergantung



kecepatan internet). Saat proses berlangsung, aplikasi menampilkan layar tunggu dan memberitahu: *"Sedang memproses, mohon tunggu."*

Langkah 4 – Dengarkan Hasil Deskripsi Setelah selesai, aplikasi akan secara otomatis membacakan deskripsi pakaian, misalnya: *"Kemeja lengan panjang berwarna biru dongker dengan motif kotak-kotak putih tipis. Berbahan katun dengan kancing di bagian depan. Cocok digunakan untuk acara formal maupun semi-formal."*

Hasil deskripsi juga ditampilkan sebagai teks dalam panel chat di layar.

Langkah 5 – Ajukan Pertanyaan Tambahan

Setelah hasil muncul, pengguna dapat mengajukan pertanyaan lanjutan:

1. Tekan tombol **"Tanya Asisten"** (tombol mikrofon) di bawah layar
2. Tunggu hingga tombol berubah merah (aplikasi mendengarkan)
3. Ucapkan pertanyaan Anda, misalnya:
 - *"Apakah cocok untuk kondangan?"*
 - *"Warna apa ini?"*
 - *"Bagaimana dengan motifnya?"*
4. Tekan tombol mikrofon sekali lagi untuk mengirim pertanyaan
5. Tunggu dan dengarkan jawaban AI



Langkah 6 – Potret Ulang Jika ingin menganalisis pakaian yang berbeda, tekan tombol "**Scan Ulang**" (ikon refresh) di bagian bawah layar.

Contoh Pertanyaan yang Bisa Diajukan:

Pertanyaan	Contoh Ucapan
Warna pakaian	"Apa warna baju ini?"
Jenis/model pakaian	"Ini baju apa?"
Kecocokan acara	"Cocok tidak untuk ke kantor?"
Motif/corak	"Ada motifnya tidak?"
Kondisi pakaian	"Apakah ada noda atau kotoran?"

Tombol Bantuan:

- Tekan tombol ? di pojok kanan atas untuk mendengar panduan fitur ini

RINGKASAN SEMUA FITUR

Fitur	Cara Masuk	Cara Kerja Utama	Butuh Internet?
Kalibrasi	Otomatis saat pertama kali	Arahkan kartu ATM/KTP pada jarak 1 jengkal	Tidak
Mode Sekitar	Menu ke-1 / ucap "Sekitar"	Arahkan kamera, dengarkan deteksi otomatis	Hanya untuk Deskripsi Cerdas
Mode Pencarian	Menu ke-2 / ucap "Cari"	Pilih benda → arahkan kamera → ikuti panduan suara	Tidak
Mode Navigasi	Menu ke-3 / ucap "Jalan"	Tentukan tujuan → ikuti panduan suara	Ya (GPS + Peta)
Mode Baca Teks	Menu ke-5 / ucap "Baca"	Arahkan kamera ke teks → dengar pembacaan → tanya lebih lanjut	Ya
Mode Uang	Menu ke-4 / ucap "Uang"	Arahkan uang ke kamera → dengar nominal	Tidak
Mode Pakaian	Menu ke-6 / ucap "Pakaian"	Potret pakaian → dengar deskripsi → tanya lebih lanjut	Ya

Lampiran 5 Dokumen Teknis



DOKUMEN TEKNIS APLIKASI TUNTUN

Aplikasi Mobile Cerdas untuk Asistensi Visual
Tunanetra Berbasis Computer Vision

Oleh:
Berliani Risqi Dwi Saputri - 22090124

1. Gambaran Umum Proyek

TunTun merupakan aplikasi mobile berbasis Flutter yang dirancang sebagai asisten visual bagi penyandang tunanetra. Aplikasi ini mengintegrasikan berbagai teknologi AI diantaranya deteksi objek real-time (YOLO), pengenalan teks (OCR), navigasi GPS berbasis suara, dan model bahasa besar (Google Gemini) dalam satu platform mobile yang dapat dioperasikan sepenuhnya tanpa melihat layar.

1.1 Stack Teknologi

Komponen	Teknologi
Framework	Flutter (Dart SDK ^3.8.0)
State Management	GetX ^4.7.3
Deteksi Objek (On-Device)	ultralytics_yolo ^0.1.19 + TFLite
OCR	google_mlkit_text_recognition ^0.11.0
AI Cloud	Google Gemini (via Cloudflare Worker Proxy)
Speech-to-Text	speech_to_text ^7.3.0
Text-to-Speech	flutter_tts ^3.8.3
GPS & Navigasi	geolocator ^14.0.2, geocoding ^4.0.0
Background Service	flutter_foreground_task ^9.2.2
Kompas	flutter_compass ^0.8.0
Peta	flutter_map ^7.0.0

1.2 Arsitektur Modul

Setiap fitur diorganisir dalam pola MVC yang difasilitasi GetX, dengan struktur direktori konsisten:

```
lib/app/modules/[nama_modul]/
├── bindings/      → Mendaftarkan controller ke DI container
├── controllers/   → Business logic (GetXController)
└── views/        → UI layer (GetView<ControllerType>)
```

1.3 Alur Navigasi Aplikasi

```
SPLASH → INTRO → CALIBRATION → MAIN
├── Tab 0: Sekitar
├── Tab 1: Cari Objek
├── Tab 2: Jalan (Walking + GPS)
├── Tab 3: Uang
├── Tab 4: Baca
└── Tab 5: Pakaian (OOTD)
```

2. Inisialisasi Aplikasi

2.1 Entry Point — main.dart

```
// lib/main.dart
void main() async {
  WidgetsFlutterBinding.ensureInitialized();

  FlutterForegroundTask.initCommunicationPort();
}
```

```

await GetStorage.init();

await Get.putAsync(() => TtsService()).init();

runApp(
  GetMaterialApp(
    title: "TunTun",
    initialRoute: AppPages.INITIAL,
    getPages: AppPages.routes,
    theme: ThemeData(
      primaryColor: AppColors.primary,
      fontFamily: 'Poppins',
    ),
  ),
);
}

```

Tiga langkah inisialisasi dilakukan secara berurutan sebelum `runApp()`. Pertama, port komunikasi untuk background isolate GPS diinisialisasi agar navigasi latar belakang dapat mengirim data ke UI thread. Kedua, `GetStorage` dibuka sebagai persistent storage lokal untuk menyimpan hasil kalibrasi dan status tutorial. Ketiga, `TtsService` didaftarkan sebagai singleton global menggunakan `Get.putAsync()` sehingga tersedia di seluruh aplikasi melalui `Get.find<TtsService>()` tanpa perlu inisialisasi ulang.

2.2 Dependency Injection — MainBinding

```

// lib/app/modules/main/bindings/main_binding.dart
class MainBinding extends Bindings {
  @override
  void dependencies() {
    Get.lazyPut<MainController>(() => MainController());
    Get.lazyPut<SurroundingController>(() => SurroundingController(), fenix:
true);
    Get.lazyPut<SearchingController>(() => SearchingController(), fenix:
true);
    Get.lazyPut<ReadingController>(() => ReadingController(), fenix:
true);
    Get.lazyPut<WalkingController>(() => WalkingController(), fenix:
true);
    Get.lazyPut<OotdController>(() => OotdController(), fenix:
true);
    Get.lazyPut<MoneyController>(() => MoneyController(), fenix:
true);
    Get.lazyPut<GpsController>(() => GpsController(), fenix:
true);
  }
}

```

`MainBinding` mendaftarkan seluruh controller yang dibutuhkan layar utama sebelum `MainView` ditampilkan. Parameter `fenix: true` memastikan controller dibuat ulang secara otomatis jika sebelumnya telah dihapus dari memori, mekanisme penting untuk tab yang jarang diakses. Dengan `lazyPut`, controller baru dibuat saat pertama kali dicari (`Get.find()`), bukan langsung saat binding dijalankan, sehingga hemat memori di awal.

3. Layanan Global: TtsService

3.1 Tiga Mode Bicara

```
// lib/app/services/tts_service.dart

// Mode 1: Non-blocking - langsung bicara, tidak menunggu selesai
Future<void> speak(String text) async {
  if (text.isEmpty) return;
  if (_speakCompleter != null && !_speakCompleter!.isCompleted)
    _speakCompleter!.complete();
  await _flutterTts.speak(text);
}

// Mode 2: Blocking - menunggu audio benar-benar selesai diucapkan
Future<void> speakAndWait(String text) async {
  if (text.isEmpty) return;
  _speakCompleter = Completer<void>();
  await _flutterTts.speak(text);

  // Timeout adaptif: cegah deadlock jika TTS engine hang
  int timeoutSeconds = (text.length / 5).ceil() + 5;
  await _speakCompleter!.future.timeout(Duration(seconds: timeoutSeconds));
}

// Mode 3: Navigasi - selalu stop TTS lama sebelum bicara
Future<void> speakNavigation(String text) async {
  if (text.isEmpty) return;
  await _flutterTts.stop();
  if (_speakCompleter != null && !_speakCompleter!.isCompleted)
    _speakCompleter!.complete();
  await Future.delayed(const Duration(milliseconds: 80));
  await _flutterTts.speak(text);
}
```

TtsService menyediakan tiga mode bicara dengan perilaku berbeda sesuai kebutuhan konteks. `speak()` digunakan untuk notifikasi instan seperti peringatan bahaya yang tidak perlu ditunggu selesai. `speakAndWait()` menggunakan pola `Completer<void>` untuk memblokir eksekusi sampai audio benar-benar selesai diucapkan, penting untuk urutan instruksi seperti onboarding. `speakNavigation()` selalu menghentikan TTS yang sedang berjalan sebelum bicara agar instruksi navigasi baru tidak mengantri di belakang instruksi lama yang sudah tidak relevan.

3.2 Kompatibilitas TalkBack

```
Future<void> waitForTalkbackAnnouncement(String announcementText) async {
  final isAccessibilityOn =
    SemanticsBinding.instance.accessibilityFeatures.accessibleNavigation;
  if (!isAccessibilityOn) return;

  // Estimasi waktu TalkBack membaca berdasarkan jumlah kata
  final wordCount = announcementText.trim().split(RegExp(r'\s+')).length;
  final msPerWord = (60000 / (150.0 * _speechRate)).round();
  final estimatedMs = (wordCount * msPerWord).clamp(400, 1500);

  await Future.delayed(Duration(milliseconds: estimatedMs));
}
```

Saat TalkBack aktif, Android screen reader mengumumkan fokus elemen UI secara otomatis setiap kali halaman dimuat. Jika TTS aplikasi langsung berbicara bersamaan, keduanya saling bertabrakan dan menghasilkan audio yang tidak dapat dipahami. Fungsi ini mendeteksi apakah TalkBack aktif

melalui `SemanticsBinding`, lalu menghitung estimasi durasi `TalkBack` membacakan label berdasarkan jumlah kata dan kecepatan bicara. Delay proporsional ini memastikan TTS aplikasi baru berbicara setelah `TalkBack` selesai.

4. Layar Utama: Navigasi Suara Global

4.1 Two-Finger Long Press Detector

```
// lib/app/modules/main/views/main_view.dart
class _TwoFingerLongPressDetectorState extends
State<_TwoFingerLongPressDetector> {
  int _activePointers = 0;
  Timer? _longPressTimer;
  bool _triggered = false;
  static const Duration _kLongPressDuration = Duration(milliseconds: 750);

  void _onPointerDown(PointerDownEvent event) {
    _activePointers++;
    if (_activePointers == 2 && !_triggered) {
      _longPressTimer?.cancel();
      _longPressTimer = Timer(_kLongPressDuration, () {
        if (_activePointers >= 2 && !_triggered) {
          _triggered = true;
          widget.onTrigger();
        }
      });
    } else if (_activePointers > 2) {
      _longPressTimer?.cancel(); // >2 jari → batalkan
    }
  }

  void _onPointerUp(PointerUpEvent event) {
    _activePointers = (_activePointers - 1).clamp(0, 10);
    if (_activePointers < 2) {
      _longPressTimer?.cancel();
      Future.delayed(const Duration(milliseconds: 300), () => _triggered =
false);
    }
  }

  @override
  Widget build(BuildContext context) {
    return Listener(
      behavior: HitTestBehavior.translucent,
      onPointerDown: _onPointerDown,
      onPointerUp: _onPointerUp,
      onPointerCancel: _onPointerCancel,
      child: widget.child,
    );
  }
}
```

Gesture dua jari tahan digunakan sebagai pemicu navigasi suara global karena gesture ini tidak bentrok dengan gesture satu jari yang sudah digunakan oleh fitur lain (scroll, tap, swipe). Implementasi menggunakan `Listener` level rendah, bukan `GestureDetector` karena `Listener` dapat menghitung jumlah pointer secara tepat tanpa mengganggu gesture recognizer widget di dalamnya. Timer 750ms dimulai saat tepat dua jari menyentuh layar dan dibatalkan jika jari diangkat atau jari ketiga ditambahkan.

4.2 Global Voice Router — MainController

```
// lib/app/modules/main/controllers/main_controller.dart
static const Map<String, int> _tabKeywords = {
  'sekitar': 0, 'lingkungan': 0, 'sekeliling': 0,
  'cari': 1, 'pencarian': 1, 'objek': 1, 'benda': 1,
  'jalan': 2, 'navigasi': 2, 'peta': 2, 'perjalanan': 2,
  'uang': 3, 'duit': 3, 'nominal': 3, 'rupiah': 3,
  'baca': 4, 'teks': 4, 'tulisan': 4, 'bacaan': 4,
  'pakaian': 5, 'baju': 5, 'outfit': 5, 'busana': 5,
};

Future<void> startGlobalVoiceNav() async {
  if (_isGlobalListening) return;
  _isGlobalListening = true;
  isGlobalListening.value = true;

  await _leaveTab(tabIndex.value); // Hentikan semua aktivitas tab aktif
  await _ttsService.stop();

  await _ttsService.speakAndWait(
    'Sebarkan nama fitur: Sekitar, Cari Objek, Jalan, Uang, Baca, atau Pakaian.',
  );

  String heard = '';
  final completer = Completer<void>();

  await _globalSpeech.listen(
    localeId: 'id_ID',
    listenFor: const Duration(seconds: 8),
    pauseFor: const Duration(seconds: 2),
    onResult: (val) {
      if (val.recognizedWords.trim().isNotEmpty)
        heard = val.recognizedWords.toLowerCase().trim();
      if (val.finalResult && !completer.isCompleted)
        completer.complete();
    },
  );

  await completer.future.timeout(const Duration(seconds: 10), onTimeout: () {});
  await _globalSpeech.stop();

  _isGlobalListening = false;
  isGlobalListening.value = false;

  int? targetTab = _matchTabKeyword(heard);
  if (targetTab != null && targetTab != tabIndex.value) {
    await changeTabIndex(targetTab);
  } else {
    await _enterTab(tabIndex.value);
  }
}
```

Fungsi ini menggabungkan seluruh alur navigasi antar-fitur menggunakan suara. Alurnya dimulai dengan menghentikan semua aktivitas tab yang sedang berjalan agar tidak ada konflik sumber daya, kemudian memutar instruksi singkat, mendengarkan input suara selama maksimum 8 detik menggunakan pola Completer yang bersifat blocking, dan mencocokkan kata yang terdengar dengan peta kata kunci. Jika kata cocok ditemukan, fungsi `changeTabIndex()` dipanggil yang secara berurutan menjalankan `_leaveTab()` untuk tab lama dan `_enterTab()` untuk tab baru.

5. Modul Kalibrasi

5.1 Sistem Zona Posisi Kartu

```
// lib/app/modules/calibration/controllers/calibration_controller.dart

// Threshold zona berdasarkan rasio tinggi bbox kartu / dimensi terpendek layar
const double _kZoneTooFar = 0.10;
const double _kZoneGuideFar = 0.17;
const double _kZoneGuideClose = 0.40;
const double _kZoneTooClose = 0.55;

enum CardZone { tooFar, guideFar, ideal, guideClose, tooClose }

CardZone _classifyZone(double ratio) {
  if (ratio < _kZoneTooFar) return CardZone.tooFar;
  if (ratio < _kZoneGuideFar) return CardZone.guideFar;
  if (ratio <= _kZoneGuideClose) return CardZone.ideal;
  if (ratio <= _kZoneTooClose) return CardZone.guideClose;
  return CardZone.tooClose;
}
```

Kalibrasi menggunakan kartu KTP/ATM (ukuran standar ISO 7810, tinggi 5.398 cm) sebagai objek referensi pada jarak 20 cm. Posisi kartu diklasifikasikan ke dalam lima zona berdasarkan rasio antara tinggi bounding box kartu (dalam piksel) terhadap dimensi terpendek layar. Zona ideal (rasio 0.17–0.40) adalah satu-satunya zona yang data sampelnya dikumpulkan secara penuh. Zona panduan masih menerima data tetapi disertai instruksi audio agar pengguna menyesuaikan jarak. Zona terlalu jauh dan terlalu dekat langsung menolak data.

5.2 Pengumpulan Sampel dan Outlier Rejection

```
void onObjectDetected(List<YOLOResult> results, double screenW, double screenH)
{
  if (isCalibrating.value || _isInstructionPlaying) return;

  final target = results.firstWhereOrNull(
    (r) => r.className.toLowerCase().contains('card'),
  );
  if (target == null || target.confidence < 0.5) return;

  final Rect bb = target.boundingBox;
  double hBox = bb.height < bb.width ? bb.height : bb.width;
  double refDim = screenH < screenW ? screenH : screenW;
  double ratio = hBox / refDim;

  CardZone zone = _classifyZone(ratio);

  // Hitung focal length dari frame ini:  $f = (H_{\text{pixel}} \times d) / H_{\text{real}}$ 
  double currentFocal = (hBox * knownDistanceCm) / knownHeightCm;

  if (_validFocalLengths.isNotEmpty) {
    double avg = _validFocalLengths.reduce((a, b) => a + b) /
      _validFocalLengths.length;
    if ((currentFocal - avg).abs() > (avg * 0.15)) {
      _unstableFrameCount++;
      if (_unstableFrameCount >= _maxUnstableBeforeReset) {
        _validFocalLengths.clear();
        _unstableFrameCount = 0;
        _speakDebounce("Jarak berubah terlalu jauh. Perhitungan diulang dari nol.");
      }
    }
  }
}
```

```

        return;
    }
}

_unstableFrameCount = 0;
_validFocalLengths.add(currentFocal);
calibProgress.value = _validFocalLengths.length / _requiredSamples;

if (_validFocalLengths.length >= _requiredSamples) _finishCalibration();
}

```

Setiap frame yang terdeteksi oleh model `yolov11_card.tflite` dihitung nilai focal length-nya menggunakan rumus $f = (H_{\text{pixel}} \times d) / H_{\text{real}}$. Sistem outlier rejection menolak data yang menyimpang lebih dari 15% dari rata-rata sementara untuk menyaring getaran tangan. Jika ketidakstabilan berlangsung lebih dari 20 frame berturut-turut, seluruh data direset dan proses dimulai dari awal. Sistem mengumpulkan tepat 30 sampel yang valid sebelum melanjutkan ke finalisasi.

5.3 Finalisasi dengan Nilai Median

```

void _finishCalibration() {
    isCalibrating.value = true;

    _validFocalLengths.sort();
    double finalFocalLength = _validFocalLengths[_validFocalLengths.length ~/ 2];

    box.write('focalLength', finalFocalLength);
    box.write('isCalibrated', true);

    _ttsService.speakAndWait("Kalibrasi selesai. Mengalihkan ke menu utama.");
    Get.offAllNamed(Routes.MAIN);
}

```

Nilai median dipilih sebagai hasil akhir kalibrasi bukan rata-rata karena median lebih tahan terhadap outlier yang mungkin lolos dari filter 15%. Hasil disimpan ke `GetStorage` dengan kunci 'focalLength' dan digunakan oleh modul Sekitar setiap kali menghitung jarak objek. Setelah kalibrasi selesai, rute dialihkan ke layar utama menggunakan `Get.offAllNamed()` yang menghapus seluruh history navigasi sehingga pengguna tidak dapat kembali ke halaman kalibrasi dengan tombol back.

6. Modul Sekitar

6.1 Kalkulasi Posisi Spasial (Grid 3×3)

```

// lib/app/modules/surrounding/controllers/surrounding_controller.dart
String _calculateGridPosition(Rect box, double screenW, double screenH) {
    double cX = box.center.dx / screenW;
    double cY = box.center.dy / screenH;

    String hTag = (cX < 0.35) ? "sebelah kiri"
        : (cX > 0.65) ? "sebelah kanan"
        : "Tengah";
    String vTag = (cY < 0.35) ? "atas"
        : (cY > 0.65) ? "bawah"
        : "Tengah";

    if (hTag == "Tengah" && vTag == "Tengah") return "Tepat di Depan";
    if (hTag == "Tengah") return "depan bagian $vTag";
    if (vTag == "Tengah") return hTag;
}

```

```

    return "$hTag bagian $vTag";
}

```

Koordinat pusat bounding box setiap objek yang terdeteksi dinormalisasi menjadi nilai 0.0–1.0 relatif terhadap dimensi layar. Nilai tersebut kemudian diklasifikasikan ke dalam sel grid 3×3 yang menghasilkan deskripsi posisi verbal seperti "sebelah kiri bagian atas" atau "Tepat di Depan". Output fungsi ini langsung dimasukkan ke dalam kalimat TTS sehingga pengguna tunanetra dapat memahami lokasi relatif objek tanpa perlu melihat layar.

6.2 Kalkulasi Jarak Berbasis Focal Length

```

// lib/app/Utils/distance_estimator.dart
static String calculateDistanceText({
  required Rect boundingBox,
  required double screenW, required double screenH,
  required double focalLength,
  required double realHeightCm,
}) {
  if (focalLength <= 0) return '';

  // Jika bbox menyentuh tepi layar, objek terlalu dekat untuk diukur
  bool isCutOff =
    (boundingBox.left < screenW * 0.05) ||
    (boundingBox.right > screenW * 0.95) ||
    (boundingBox.top < screenH * 0.05) ||
    (boundingBox.bottom > screenH * 0.95);
  if (isCutOff) return 'sangat dekat';

  // Rumus: d = (H_real * f) / H_pixel
  double distCm = (realHeightCm * focalLength) / boundingBox.height;
  double distMeter = distCm / 100;

  if (distMeter < 1.0) return '${distCm.round()} sentimeter';

  String meterStr = distMeter.toStringAsFixed(1).replaceAll('.', ' koma ');
  if (meterStr.endsWith(' koma 0')) meterStr = distMeter.toStringAsFixed(0);
  return '$meterStr meter';
}

```

Kalkulasi jarak menggunakan rumus *similar triangles* dari optik kamera: jarak objek berbanding terbalik dengan tinggi pikselnya di gambar. Nilai focal length yang diperoleh dari kalibrasi digunakan bersama database tinggi fisik tiap objek (tersimpan di `object_height.dart`) untuk menghasilkan estimasi jarak dalam satuan meter atau sentimeter. Deteksi *cutoff* (objek yang terpotong tepi layar) digunakan sebagai indikator bahwa objek sudah terlalu dekat untuk diukur secara akurat. Format output dirancang ramah TTS: menggunakan kata "koma" sebagai pengganti titik desimal.

6.3 Pipeline Deteksi dan TTS Multi-Objek

```

void onObjectDetected(List<YOLOResult> results, double screenH, double screenW)
{
  if (!_isYoloActive || _isCurrentlySpeaking || _isGeminiSpeaking) return;

  if (_cameraStartTime != null &&
    DateTime.now().difference(_cameraStartTime!).inSeconds < 3) return;

  final validObjects = results.where((r) => r.confidence > 0.5).toList();

  validObjects.sort((a, b) =>
    (b.boundingBox.height * b.boundingBox.width)
    .compareTo(a.boundingBox.height * a.boundingBox.width));
}

```

```

        if (validObjects.isNotEmpty)
            _processAndSpeakMultipleObjects(validObjects, screenH, screenW);
    }

    Future<void> _processAndSpeakMultipleObjects(
        List<YOLOResult> objects, double screenH, double screenW) async {
        int maxObjects = objects.length > 3 ? 3 : objects.length;
        List<String> kalimatBagian = [];

        for (int i = 0; i < maxObjects; i++) {
            YOLOResult obj = objects[i];
            String indoLabel = indonesianLabels[obj.className.toLowerCase()] ??
            obj.className;
            String position = _calculateGridPosition(obj.boundingBox, screenW,
            screenH);
            double realH = objectHeights[obj.className.toLowerCase()] ??
            fallbackHeightCm;
            String jarak = DistanceEstimator.calculateDistanceText(
                boundingBox: obj.boundingBox,
                screenW: screenW, screenH: screenH,
                focalLength: focalLength, realHeightCm: realH,
            );

            kalimatBagian.add(i == 0
                ? (position == "Tepat di Depan"
                    ? "Di depan anda ada $indoLabel berjarak $jarak"
                    : "Di $position ada $indoLabel berjarak $jarak")
                : "dan di $position ada $indoLabel berjarak $jarak");
        }

        String fullSentence = kalimatBagian.join(", ");

        if (_lastSpokenSentence == fullSentence && _lastSpeakTime != null &&
            DateTime.now().difference(_lastSpeakTime!) < _repetitionDelay) return;

        _isCurrentlySpeaking = true;
        try {
            await _ttsService.speakAndWait(fullSentence);
            _lastSpokenSentence = fullSentence;
            _lastSpeakTime = DateTime.now();
        } finally {
            _isCurrentlySpeaking = false;
        }
    }
}

```

Setiap frame yang mengandung deteksi valid (confidence > 0.5) diproses untuk menghasilkan satu kalimat deskripsi lengkap. Objek diurutkan berdasarkan luas bounding box sehingga objek yang paling dekat (paling besar di layar) selalu disebutkan pertama. Maksimum tiga objek diproses per kalimat untuk menjaga durasi audio tidak terlalu panjang. Mekanisme anti-spam memastikan kalimat identik tidak diulang dalam 10 detik, mencegah TTS berbunyi berulang saat objek tetap diam.

6.4 Deskripsi Cerdas via Gemini API

```

Future<void> captureSmartDescription() async {
    if (isSmartProcessing.value) return;
    isSmartProcessing.value = true;

    // Jeda YOLO, ambil foto dengan kamera terpisah, lalu aktifkan YOLO kembali
    isCameraActive.value = false;
    CameraController? tempCam;
    final cameras = await availableCameras();
}

```

```

tempCam = CameraController(cameras.first, ResolutionPreset.medium,
enableAudio: false);
await tempCam.initialize();
final photo = await tempCam.takePicture();
await tempCam.dispose();
isCameraActive.value = true;
yoloSession.value++; // Paksa YOLOView rebuild

// Kompresi untuk mengurangi payload: 800*600, quality 70%
final compressedBytes = await FlutterImageCompress.compressWithFile(
  photo.path, minWidth: 800, minHeight: 600, quality: 70,
);
final base64Image = base64Encode(compressedBytes!);
File(photo.path).delete().catchError((_) {});

final response = await http.post(
  Uri.parse(ApiConstants.geminiProxyUrl),
  headers: {"Content-Type": "application/json"},
  body: jsonEncode({
    "imageBase64": base64Image,
    "isInteractiveMode": false,
    "userQuestion":
      "Deskripsikan gambar ini untuk pengguna tunanetra. "
      "Gunakan gaya narasi objektif – tanpa kata ganti orang pertama, tanpa
sapaan. "
      "Struktur: ringkasan umum, lalu objek dari kiri ke kanan, lalu konteks.
"
      "Panjang: 3 kalimat.",
  })),
);

if (response.statusCode == 200) {
  final data = jsonDecode(response.body);
  if (data['success'] == true) {
    smartResult.value = data['text'];
    isShowingResult.value = true;
    isSmartProcessing.value = false;
    await _ttsService.speakAndWait(data['text']);
    isShowingResult.value = false;
  }
}
}
}

```

Fitur deskripsi cerdas menggunakan arsitektur kamera sementara, kamera YOLO dijeda sementara, kamera terpisah dibuka hanya untuk mengambil satu foto, kemudian langsung di-dispose. Gambar dikompresi sebelum dikirim untuk mengurangi ukuran payload HTTP. Prompt ditulis secara spesifik untuk memastikan output Gemini mengikuti struktur naratif yang cocok dibacakan TTS: objektif, tanpa kata ganti orang pertama, dan terstruktur dari kiri ke kanan. Nilai `yoloSession` diinkrementasi untuk memaksa `YOLOView` merender ulang dengan `ValueKey` baru setelah kamera dikembalikan.

7. Modul Jalan (Walking & GPS)

7.1 Konfigurasi Terrain dan Pola Vibrasi

```

// lib/app/utils/terrain_config.dart
enum TerrainType { safe, caution, danger, guide }

class TerrainInfo {
  final String label;
  final TerrainType type;
}

```

```

final Color color;
final List<int> vibPattern; // Format: [delay_ms, on_ms, off_ms, on_ms, ...]
const TerrainInfo(this.label, this.type, this.color, this.vibPattern);
}

const Map<int, TerrainInfo> terrainMap = {
  0: TerrainInfo('Sepeda', TerrainType.caution, Colors.orange, [0, 150,
50, 150]),
  1: TerrainInfo('Bus', TerrainType.danger, Colors.red, [0, 400,
100, 400]),
  2: TerrainInfo('Mobil', TerrainType.danger, Colors.red, [0, 300,
100, 300]),
  3: TerrainInfo('Zebra Cross', TerrainType.guide, Colors.yellow, [0, 50,
100, 50]),
  4: TerrainInfo('Guiding Block', TerrainType.guide, Colors.yellow, [0, 50,
50, 50]),
  5: TerrainInfo('Motor', TerrainType.danger, Colors.red, [0, 200,
50, 200]),
  6: TerrainInfo('Orang', TerrainType.caution, Colors.orange, [0, 100,
50, 100]),
  7: TerrainInfo('Tiang', TerrainType.danger, Colors.red, [0, 250,
50, 250]),
  8: TerrainInfo('Lubang', TerrainType.danger, Colors.red, [0, 300,
50, 300]),
  9: TerrainInfo('Genangan Air', TerrainType.caution, Colors.orange, [0, 150,
50, 150]),
  10: TerrainInfo('Tangga', TerrainType.caution, Colors.orange, [0, 200,
100, 200]),
  11: TerrainInfo('Pohon', TerrainType.danger, Colors.red, [0, 200,
50, 200]),
  12: TerrainInfo('Truk', TerrainType.danger, Colors.red, [0, 500,
100, 500]),
};

```

terrainMap digunakan untuk konfigurasi terpusat yang memetakan setiap index kelas model ke properti lengkap termasuk label Bahasa Indonesia, tipe bahaya, warna UI, dan pola vibrasi. Pola vibrasi dirancang berbeda untuk setiap kelas sehingga pengguna dapat membedakan jenis ancaman melalui haptic feedback tanpa bergantung pada audio, misalnya truk menghasilkan vibrasi paling panjang (500ms) karena merupakan ancaman terbesar, sementara guiding block menghasilkan dua ketukan pendek yang bersifat informatif.

7.2 Logika Deteksi Bahaya dengan Prioritas

```

// lib/app/modules/walking/controllers/walking_controller.dart
void onObjectDetected(List<YOLOResult> results) {
  if (_disposed || _isSearching || !isCameraInitialized.value ||
    isYoloPausedByVehicleMode.value || _isTutorialPlaying) return;

  // Confidence threshold 0.55 untuk kurangi false positive
  final validDetections = results.where((r) => r.confidence > 0.55).toList();
  if (validDetections.isEmpty) { _setIdle(); return; }

  YOLOResult? dangerDet;
  YOLOResult? guideDet;

  for (final det in validDetections) {
    int classIndex = modelClassNames.indexOf(det.className);
    if (classIndex == -1) continue;

    // Abaikan objek di 30% atas layar (objek jauh di horizon)
    if (det.boundingBox.center.dy / Get.height < 0.3) continue;
  }
}

```

```

        if (isDanger(classIndex) && dangerDet == null) dangerDet = det;
        if ((isGuide(classIndex) || isSafe(classIndex)) && guideDet == null)
            guideDet = det;
    }

    // Prioritas: bahaya > panduan > idle
    if (dangerDet != null) _executeDangerAlert(dangerDet);
    else if (guideDet != null) _executeGuideAlert(guideDet);
    else _setIdle();
}

void _executeDangerAlert(YOLOResult det) async {
    int classIndex = modelClassNames.indexOf(det.className);
    final info = getTerrainInfo(classIndex);

    currentAction.value = info.label.toUpperCase();
    uiColor.value = info.color;

    final now = DateTime.now();
    if (_lastActionTime == null ||
        now.difference(_lastActionTime!).inSeconds >= 2 ||
        _lastMessage != info.label) {
        _lastActionTime = now;
        _lastMessage = info.label;

        Vibration.vibrate(pattern: info.vibPattern);

        // Estimasi jarak dari luas bbox (tanpa focal length)
        double bboxArea = (det.boundingBox.width / Get.width) *
            (det.boundingBox.height / Get.height);
        _ttsService.speak('Awat! ${info.label} ${_estimateDistance(bboxArea)}');
    }
}

String _estimateDistance(double bboxArea) {
    if (bboxArea > 0.35) return 'sangat dekat, kurang dari 1 meter';
    if (bboxArea > 0.15) return 'sekitar 1 hingga 2 meter';
    if (bboxArea > 0.06) return 'sekitar 2 hingga 5 meter';
    if (bboxArea > 0.02) return 'sekitar 5 hingga 10 meter';
    return 'lebih dari 10 meter';
}

```

Sistem deteksi menggunakan confidence threshold yang lebih tinggi (0.55 vs 0.5 di modul Sekitar) untuk mengurangi false positive yang dapat menyebabkan peringatan bahaya yang tidak perlu. Filter vertikal membuang objek yang bounding box-nya berada di 30% atas layar karena objek jauh di horizon tidak merupakan ancaman langsung. Prioritas diberikan kepada objek bahaya di atas objek panduan. Estimasi jarak modul Walking menggunakan pendekatan empiris berbasis luas bounding box relatif terhadap layar karena modul ini tidak bergantung pada kalibrasi focal length.

7.3 GPS Background Service

```

// lib/app/services/gps_background_service.dart
Future<bool> startWithDestination({
    required LatLng target,
    required String name,
    required bool isVehicle,
}) async {
    LocationPermission perm = await Geolocator.checkPermission();
    if (perm == LocationPermission.denied)
        perm = await Geolocator.requestPermission();
}

```

```

ServiceRequestResult result;
if (await FlutterForegroundTask.isRunningService) {
  result = await FlutterForegroundTask.restartService();
} else {
  result = await FlutterForegroundTask.startService(
    serviceId: 1001,
    notificationTitle: '👤 Menuju $name',
    notificationText: 'GPS aktif · Menghitung rute...',
    notificationButtons: [
      const NotificationButton(id: 'btn_stop', text: '■ Stop'),
      const NotificationButton(id: 'btn_status', text: '📍 Lokasi Saya'),
    ],
    serviceTypes: [ForegroundServiceTypes.location],
    callback: gpsTaskCallback,
  );
}

if (result is ServiceRequestSuccess) {
  await Future.delayed(const Duration(milliseconds: 800));
  sendDestinationToTask(target: target, name: name, isVehicle: isVehicle);
  return true;
}
return false;
}

void sendDestinationToTask({required LatLng target, required String name, bool
isVehicle = false}) {
  FlutterForegroundTask.sendDataToTask({
    'action': 'set_destination',
    'lat': target.latitude,
    'lng': target.longitude,
    'name': name,
    'isVehicle': isVehicle,
  });
}

```

Navigasi GPS berjalan sebagai *foreground service* di isolate terpisah agar tetap aktif saat aplikasi diminimalkan, ini adalah persyaratan Android untuk akses lokasi berkelanjutan. Service ditampilkan kepada pengguna melalui notifikasi persisten dengan tombol aksi yang dapat langsung menghentikan navigasi atau membacakan lokasi saat ini. Setelah service berhasil dimulai, destinasi dikirim ke background isolate via `sendDataToTask()` menggunakan protokol JSON berbasis aksi.

7.4 GpsTaskHandler — Kalkulasi Arah di Background

```

// lib/app/services/gps_task_handler.dart
void _recalculateDirection({double gpsHeading = -1}) {
  final bearingToTarget = const Distance().bearing(_currentLocation!,
_targetLocation!);

  double currentHeading;
  if (gpsHeading >= 0) {
    currentHeading = gpsHeading;
  } else if (_previousLocation != null) {
    currentHeading = const Distance().bearing(_previousLocation!,
_currentLocation!);
    if (currentHeading < 0) currentHeading += 360;
  } else {
    currentHeading = bearingToTarget;
  }

  double diff = bearingToTarget - currentHeading;
}

```

```

if (diff < 0) diff += 360;
if (diff > 360) diff -= 360;

final instruction = _diffToInstruction(diff);

// Kirim ke main isolate HANYA jika instruksi berubah
if (instruction != _lastInstruction) {
  _lastInstruction = instruction;
  FlutterForegroundTask.sendDataToMain({
    'action': 'direction_changed',
    'instruction': instruction,
    'distance': _distanceMeters,
    'name': _targetName,
  });
}
}

String _diffToInstruction(double diff) {
  if (diff >= 340 || diff <= 20) return 'Di depan';
  if (diff > 20 && diff <= 70) return 'Serong kanan';
  if (diff > 70 && diff <= 110) return 'Ke kanan';
  if (diff > 110 && diff <= 160) return 'Kanan belakang';
  if (diff > 160 && diff <= 200) return 'Balik arah';
  if (diff > 200 && diff <= 250) return 'Kiri belakang';
  if (diff > 250 && diff <= 290) return 'Ke kiri';
  return 'Serong kiri';
}

```

Kalkulasi arah dilakukan di background isolate menggunakan selisih antara bearing ke tujuan dan heading saat ini. Heading pengguna diperoleh dari tiga sumber secara hierarkis: GPS chipset saat tersedia, perhitungan dari pergerakan posisi sebelumnya, atau diasumsikan menghadap tujuan sebagai fallback terakhir. Selisih sudut 0–360 derajat kemudian dipetakan ke delapan instruksi arah berbasis bahasa natural. Instruksi hanya dikirimkan ke main isolate saat nilainya berubah, mengurangi frekuensi komunikasi antar-isolate yang mahal secara komputasi.

8. Modul Uang

8.1 Sistem untuk Stabilitas Deteksi

```

// lib/app/modules/money/controllers/money_controller.dart
void processYoloResults(List<YOLOResult> results) {
  if (!isScanning.value || !isTutorialPlaying) return;
  if (_cameraStartTime != null &&
    DateTime.now().difference(_cameraStartTime!).inSeconds < 2) return;

  final validResults = results.where((r) => r.confidence >
    _currentThreshold).toList();
  if (validResults.isEmpty) { _handleNoMoneyDetected(); return; }

  Map<String, int> moneyCounts = {};
  int totalAmount = 0;

  for (var money in validResults) {
    String label = money.className.toLowerCase().trim();
    int nominalValue = _getNominalValue(label);
    if (nominalValue > 0) {
      moneyCounts[label] = (moneyCounts[label] ?? 0) + 1;
      totalAmount += nominalValue;
    }
  }
}

```

```

// Buat signature unik dari kombinasi uang yang terdeteksi
// Contoh: "10000x1|50000x2" → 1 lembar 10rb + 2 lembar 50rb
var sortedLabels = moneyCounts.keys.toList()..sort();
String currentFrameSignature = sortedLabels.map((k) =>
"$k)x${moneyCounts[k]}").join('|');

_processMultiNominal(currentFrameSignature, moneyCounts, totalAmount);
}

void _processMultiNominal(String signature, Map<String, int> counts, int total)
{
  if (signature == _currentSignature) {
    _hitStreak++;
  } else {
    _currentSignature = signature;
    _hitStreak = 1;
  }

  // Baru diucapkan setelah 3 frame berurutan dengan signature yang sama
  if (_hitStreak >= 3) {
    String spokenText = _buildSpokenText(counts, total);
    currentStatusMessage.value = spokenText;

    if (!_isSpeakingTTS) {
      if (signature != _lastSpokenSignature ||
        (_lastSpeakTime == null ||
          DateTime.now().difference(_lastSpeakTime!).inSeconds > 6)) {
        _lastSpokenSignature = signature;
        _lastSpeakTime = DateTime.now();
        Vibration.vibrate(duration: 150);
        _speakWithLock(spokenText);
      }
    }
    _hitStreak = 0;
  }
}
}

```

Setiap frame menghasilkan string unik yang merepresentasikan seluruh kombinasi uang yang terdeteksi, misalnya "10000x1|50000x2" untuk satu lembar 10.000 dan dua lembar 50.000. Hit streak counter memastikan kombinasi yang sama terdeteksi pada tiga frame berurutan sebelum diucapkan, menyaring fluktuasi deteksi sesaat. Mekanisme duplikat mencegah pengulangan dalam 6 detik jika kombinasi yang sama terus terdeteksi.

8.2 Pembangunan Kalimat dan Format Rupiah

```

String _buildSpokenText(Map<String, int> counts, int total) {
  // Kasus sederhana: 1 jenis, 1 lembar/keping
  if (counts.length == 1 && counts.values.first == 1) {
    String label = counts.keys.first;
    bool isCoin = label.contains('koin');
    int nominal = _getNominalValue(label);
    return "Uang ${isCoin ? 'koin' : 'kertas'} ${_formatRupiah(nominal)}
Rupiah";
  }

  // Kasus kompleks: beberapa jenis atau jumlah
  List<String> parts = [];
  counts.forEach((label, count) {
    bool isCoin = label.contains('koin');
    int nominal = _getNominalValue(label);
    String satuan = isCoin ? "keping" : "lembar";
    parts.add("$count $satuan ${_formatRupiah(nominal)}");
  });
}

```

```

});

return "Terdeteksi ${parts.join(', dan ')} Total: ${_formatRupiah(total)}
Rupiah.";
}

String _formatRupiah(int value) {
  String str = value.toString();
  String result = '';
  int count = 0;
  for (int i = str.length - 1; i >= 0; i--) {
    result = str[i] + result;
    count++;
    if (count % 3 == 0 && i > 0) result = '.' + result;
  }
  return result; // 100000 -> "100.000"
}

```

Pembangunan kalimat mempertimbangkan dua konteks berbeda: satu lembar tunggal menghasilkan kalimat sederhana seperti "Uang kertas 50.000 Rupiah", sedangkan kombinasi beberapa nominal menghasilkan kalimat rinci beserta total penjumlahan. Pemisahan antara uang kertas ("lembar") dan koin ("keping") membantu pengguna memahami jenis fisik uang yang terdeteksi. Fungsi `_formatRupiah()` mengimplementasikan format ribuan Indonesia (titik sebagai pemisah) tanpa bergantung pada library eksternal.

9. Modul Baca Teks

9.1 Auto-Capture Berbasis Coverage Area

```

// lib/app/modules/reading/controllers/reading_controller.dart
void _analyzeDocumentFraming(
  RecognizedText recognizedText, double imgWidth, double imgHeight) async {

  // Hitung bounding box gabungan dari seluruh blok teks yang terdeteksi
  double minX = double.infinity, minY = double.infinity, maxX = 0, maxY = 0;
  for (var block in recognizedText.blocks) {
    final rect = block.boundingBox;
    if (rect.left < minX) minX = rect.left;
    if (rect.top < minY) minY = rect.top;
    if (rect.right > maxX) maxX = rect.right;
    if (rect.bottom > maxY) maxY = rect.bottom;
  }

  textBoundingBox.value = Rect.fromLTRB(minX, minY, maxX, maxY);

  // Coverage: persentase area layar yang diisi oleh teks
  double coveragePercentage = ((maxX - minX) * (maxY - minY)) / (imgWidth *
imgHeight);

  if (coveragePercentage > 0.15) {
    _perfectFrameStreak++;
    Vibration.vibrate(duration: 30); // Getaran pendek: "teks terdeteksi"

    if (_perfectFrameStreak >= 3) {
      _perfectFrameStreak = 0;
      textBoundingBox.value = null;
      Vibration.vibrate(duration: 400); // Getaran panjang: "foto diambil"
      _triggerCloudAnalysis();
    }
  } else {
    _perfectFrameStreak = 0;
  }
}

```

```

        currentStatusMessage.value = "Dekatkan kamera ke teks.";
    }
}

```

Mode guiding menggunakan ML Kit Text Recognition secara lokal (on-device, tanpa API call) bukan untuk membaca teks, melainkan sebagai detektor keberadaan teks. Coverage threshold 15% memastikan bahwa teks sudah cukup besar dan jelas di frame sebelum foto diambil. Mekanisme tiga frame berurutan mencegah foto diambil saat kamera bergerak. Dua pola vibrasi yang berbeda memberikan umpan balik haptic yang jelas: getaran pendek menandakan teks terdeteksi, getaran panjang menandakan foto berhasil diambil dan sedang diproses.

9.2 Analisis Teks via Gemini dengan Prompt Terstruktur

```

Future<void> _triggerCloudAnalysis() async {
    final cam = cameraController.value;
    await cam!.stopImageStream();

    final XFile photo = await cam.takePicture();
    final bytes = await File(photo.path).readAsBytes();
    _lastBase64Image = base64Encode(bytes); // Disimpan untuk sesi Q&A
    chatHistory.clear();
    File(photo.path).delete().catchError((_) {});

    currentState.value = ReadingState.processingCloud;
    await _ttsService.speak("Teks ditemukan. Memproses gambar...");
    await _startLoadingSound();

    const String readingInstruction =
        "Ekstrak dan tulis ulang SEMUA TEKS yang ada di gambar secara lengkap dan akurat untuk tunanetra. "
        "ATURAN MUTLAK: "
        "1. DILARANG KERAS menggunakan kata ganti orang (saya, aku, kami, kita). "
        "2. DILARANG KERAS menggunakan kalimat pembuka atau basa-basi. "
        "3. LANGSUNG mulai output dengan isi teks dari gambar. "
        "4. Setelah seluruh teks, beri satu kalimat konteks dokumen. "
        "Gunakan Bahasa Indonesia.";

    final response = await http.post(
        Uri.parse(ApiConstants.geminiProxyUrl),
        headers: {"Content-Type": "application/json"},
        body: jsonEncode({
            "imageBase64": _lastBase64Image,
            "isInteractiveMode": false,
            "userQuestion": readingInstruction,
        })),
    );

    if (response.statusCode == 200) {
        final data = jsonDecode(response.body);
        final String finalResult = _cleanMarkdown(data['text']);
        chatHistory.add(ChatMessage(sender: "AI", text: finalResult));
        currentState.value = ReadingState.speaking;
        Vibration.vibrate(duration: 500);
        await _ttsService.speakAndWait(finalResult);
    }
}

```

Prompt yang dikirim ke Gemini dirancang untuk menghasilkan output yang optimal saat dibacakan TTS. Larangan penggunaan kata ganti orang pertama mencegah Gemini memulai respons dengan "Saya melihat..." yang tidak informatif. Instruksi untuk langsung memulai dengan isi teks

memastikan tidak ada kalimat pembuka yang membuang waktu pengguna. Gambar yang sama disimpan di `_lastBase64Image` agar sesi tanya-jawab berikutnya dapat merujuk ke dokumen yang sama tanpa perlu memotret ulang.

9.3 Voice Q&A dengan Chat History

```
Future<void> _sendInteractiveQuestion() async {
  currentState.value = ReadingState.processingCloud;
  await _startLoadingSound();

  // Bangun riwayat obrolan (maks. 6 pesan terakhir) untuk konteks Gemini
  List<String> rawHistory = chatHistory.map((msg) => "${msg.sender}:
  ${msg.text}").toList();
  if (rawHistory.length > 6) rawHistory = rawHistory.sublist(rawHistory.length -
  6);

  String finalPrompt =
    "Tugasmu: Menjawab pertanyaan tunanetra mengenai gambar dokumen
  sebelumnya.\n"
    "ATURAN MUTLAK:\n"
    "1. Jawab LANGSUNG ke intinya tanpa basa-basi.\n"
    "2. DILARANG menggunakan kata ganti (saya, aku, kami, kita) atau sapaan.\n";

  if (rawHistory.isNotEmpty)
    finalPrompt += "\n--- RIWAYAT OBROLAN ---\n${rawHistory.join('\n')}\n-----
  --\n\n";
  finalPrompt += "Pertanyaan pengguna: $_userQuestionText";

  final response = await http.post(
    Uri.parse(ApiConstants.geminiProxyUrl),
    headers: {"Content-Type": "application/json"},
    body: jsonEncode({
      "imageBase64": _lastBase64Image, // Gambar dokumen yang sama
      "isInteractiveMode": true,
      "userQuestion": finalPrompt,
    })),
  );

  if (response.statusCode == 200) {
    final String finalResult =
      _cleanMarkdown(jsonDecode(response.body)['text']);
    chatHistory.add(ChatMessage(sender: "User", text: _userQuestionText));
    chatHistory.add(ChatMessage(sender: "AI", text: finalResult));
    currentState.value = ReadingState.speaking;
    await _ttsService.speakAndWait(finalResult);
  }
}
```

Sistem Q&A mempertahankan konteks percakapan dengan menyertakan maksimum enam pesan terakhir dalam setiap permintaan ke Gemini. Batasan enam pesan mencegah prompt menjadi terlalu panjang yang dapat memperlambat respons API. Gambar dokumen original (`_lastBase64Image`) selalu disertakan sehingga Gemini dapat menjawab pertanyaan spesifik tentang konten visual yang tidak disebutkan dalam riwayat teks. Seluruh respons diproses oleh `_cleanMarkdown()` sebelum dibaca TTS untuk menghapus simbol Markdown seperti `**`, `##`, dan backtick.

10. Modul Pakaian (OOTD)

10.1 State Machine dan Alur Potret

```
// lib/app/modules/oofd/controllers/oofd_controller.dart
enum OofdState { initializing, ready, processingCloud, frozenCanvas, listening,
error }

Future<void> captureOutfitMaster() async {
  if (currentState.value != OofdState.ready || !_isTutorialPlaying) return;

  final cam = cameraController.value;
  await _ttsService.stop();
  Vibration.vibrate(duration: 150); // Feedback "shutter"

  currentState.value = OofdState.processingCloud;
  await Future.delayed(const Duration(milliseconds: 400));

  chatHistory.clear();

  final XFile photo = await cam!.takePicture();
  final bytes = await File(photo.path).readAsBytes();
  _frozenBase64Image = base64Encode(bytes);
  File(photo.path).delete().catchError((_) {});

  // Dispose kamera setelah foto, tidak perlu live feed lagi
  await _destroyCameraSafely();
  await _startLoadingSound();
  _ttsService.speak("Foto diambil. Sedang menganalisis warna dan bahan
pakaian...");

  final response = await http.post(
    Uri.parse(ApiConstants.geminiProxyUrl),
    headers: {"Content-Type": "application/json"},
    body: jsonEncode({
      "imageBase64": _frozenBase64Image,
      "isInteractiveMode": true,
      "userQuestion":
        "Anda adalah asisten fashion untuk tunanetra. Fokus pada pakaian. "
        "Jelaskan dengan singkat: 1. Jenis pakaian. 2. Warna dominan. 3.
Motifnya. "
        "Jika ada beberapa baju/celana, sebutkan lokasinya. "
        "Jangan gunakan kata ganti 'saya' atau 'kita'.",
    })),
  );

  if (response.statusCode == 200) {
    final data = jsonDecode(response.body);
    final String finalResult = _cleanMarkdown(data['text']);
    await _stopLoadingSound();

    chatHistory.add(OofdChatMessage(sender: "AI", text: finalResult));
    currentState.value = OofdState.frozenCanvas; // Beralih ke mode Q&A

    Vibration.vibrate(duration: 500);
    await _ttsService.speakAndWait(
      "$finalResult. Gunakan tombol Tanya Asisten di bawah jika ingin
bertanya.",
    );
  }
}
```

State machine lima-state mengelola alur OOTD. Setelah foto diambil, kamera langsung di-dispose

karena tidak dibutuhkan lagi, tampilan beralih ke gambar statis yang tersimpan di `_frozenBase64Image`. Pemilihan desain ini menghemat memori dan mencegah konflik kamera dengan modul lain. State `frozenCanvas` menandakan mode di mana pengguna dapat melakukan tanya-jawab berulang menggunakan gambar yang sama tanpa perlu memotret ulang. Prompt Gemini ditulis khusus untuk peran asisten fashion yang ringkas dan relevan bagi tunanetra.

10.2 Q&A Interaktif dengan Konteks Gambar Tetap

```
Future<void> _sendInteractiveQuestion() async {
  if (_isProcessingSTT) return;
  _isProcessingSTT = true;

  final questionSnapshot = _userQuestionText.trim();
  currentState.value = OotdState.processingCloud;
  await _startLoadingSound();

  final List<String> rawHistory = chatHistory.map((m) => "${m.sender}:
  ${m.text}").toList();
  if (rawHistory.length > 6) rawHistory.removeRange(0, rawHistory.length - 6);

  String finalPrompt = "Tugasmu asisten fashion tunanetra. Gambar pakaian
  sebelumnya masih berlaku.\n";
  if (rawHistory.isNotEmpty)
    finalPrompt += "\n--- RIWAYAT OBROLAN ---\n${rawHistory.join('\n')}\n-----
  ----\n\n";
  finalPrompt += "Pertanyaan pengguna: $questionSnapshot. Jawablah langsung
  tanpa basa-basi.";

  final response = await http.post(
    Uri.parse(ApiConstants.geminiProxyUrl),
    headers: {"Content-Type": "application/json"},
    body: jsonEncode({
      "imageBase64": _frozenBase64Image, // Gambar SAMA dari
      captureOutfitMaster()
      "isInteractiveMode": true,
      "userQuestion": finalPrompt,
    })),
  );

  if (response.statusCode == 200) {
    final String finalResult =
    _cleanMarkdown(jsonDecode(response.body)['text']);
    await _stopLoadingSound();

    chatHistory.add(OotdChatMessage(sender: "User", text: questionSnapshot));
    chatHistory.add(OotdChatMessage(sender: "AI", text: finalResult));

    currentState.value = OotdState.frozenCanvas;
    await _ttsService.speakAndWait(finalResult);
  }
  _isProcessingSTT = false;
}
```

Sistem Q&A OOTD menggunakan gambar yang sama (`_frozenBase64Image`) untuk seluruh sesi tanya-jawab, memungkinkan pengguna bertanya berulang kali tentang detail berbeda dari satu pakaian, misalnya menanyakan warna, bahan, merek, tanpa perlu mengambil foto baru. Flag `_isProcessingSTT` berfungsi sebagai mutex (mutual exclusion) untuk mencegah dua permintaan dikirim secara bersamaan jika pengguna mengetuk tombol mikrofon lebih dari sekali. Riwayat obrolan disertakan agar Gemini dapat memberikan jawaban yang mempertimbangkan konteks

pertanyaan sebelumnya.

11. Kode yang Digunakan Lintas Modul

11.1 Debouncing TTS

```
// Pola yang digunakan di CalibrationController, SurroundingController,
MoneyController
void _speakDebounced(String message) {
    final now = DateTime.now();
    if (_lastSpeakTime == null ||
        now.difference(_lastSpeakTime!).inSeconds >= 3) {
        _ttsService.speak(message);
        _lastSpeakTime = now;
    }
}
```

Pola debounce ini mencegah TTS diucapkan berulang kali dalam interval singkat saat kondisi yang sama terus terdeteksi. Threshold 3 detik dipilih sebagai keseimbangan antara memberikan umpan balik yang cukup sering dan tidak membanjiri pengguna dengan audio berulang.

11.2 Lifecycle Guard dengan Flag _disposed

```
// Pola yang digunakan di WalkingController, OotdController, ReadingController
bool _disposed = false;

@override
void onClose() {
    _disposed = true;
    // cleanup resources
    super.onClose();
}

Future<void> someAsyncOperation() async {
    if (_disposed) return;
    await Future.delayed(...);
    if (_disposed) return; // Cek lagi setelah setiap await
    await _ttsService.speak("...");
}
```

Karena banyak operasi bersifat asynchronous (API call, kamera, TTS), ada risiko operasi tersebut selesai setelah controller sudah dihapus dari memori. Flag _disposed digunakan sebagai penjaga yang dicek sebelum dan sesudah setiap await, memastikan tidak ada operasi yang berjalan pada controller yang sudah tidak aktif.

11.3 Tutorial First-Run dengan GetStorage

```
// Pola yang digunakan di semua modul fitur utama
Future<void> _checkAndPlayTutorial() async {
    bool hasHeardTutorial = box.read('tutorial_[nama_modul]') ?? false;

    if (!hasHeardTutorial) {
        _isTutorialPlaying = true; // Kunci semua deteksi selama tutorial
        await _ttsService.speakAndWait("Selamat datang di...");
        box.write('tutorial_[nama_modul]', true); // Tandai selesai
        _isTutorialPlaying = false;
    } else {
        await _ttsService.speakAndWait("Mode [nama] aktif.");
    }
}
```

Setiap modul menyimpan flag di `GetStorage` untuk memastikan tutorial onboarding hanya diputar satu kali, saat pertama kali fitur diakses. Flag `_isTutorialPlaying` mengunci seluruh pemrosesan deteksi (YOLO, OCR, kamera) selama tutorial berlangsung agar audio panduan tidak terganggu oleh notifikasi deteksi. Pengguna yang sudah pernah mengakses fitur hanya mendapat konfirmasi singkat "Mode aktif" tanpa tutorial lengkap.

Lampiran 6 Sertifikat HKI yang Terbit

 REPUBLIK INDONESIA KEMENTERIAN HUKUM	
SURAT PENCATATAN CIPTAAN	
Dalam rangka perlindungan ciptaan di bidang ilmu pengetahuan, seni dan sastra berdasarkan Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta, dengan ini menerangkan:	
Nomor dan tanggal permohonan	: EC002026081309, 8 Juni 2026
Pencipta	
Nama	: Berliani Risqi Dwi Saputri, Dwi Intan Af'idah, S.T., M.Kom dkk
Alamat	: Jalan Sumodirjo, RT 002/RW 003, No. 2, Kelurahan Kemandungan, Tegal Barat, Kota Tegal, Jawa Tengah, 52114.
Kewarganegaraan	: Indonesia
Pemegang Hak Cipta	
Nama	: Unit Penelitian Abdimas Universitas Harkat Negeri
Alamat	: Jalan Mataram No. 9, Pesturungan Lor, Kecamatan Margadana, Margadana, Kota Tegal, Jawa Tengah, 52142
Kewarganegaraan	: Indonesia
Jenis Ciptaan	: Program Komputer
Judul Ciptaan	: Aplikasi Mobile Cerdas untuk Asistensi Visual Tunanetra Berbasis Computer Vision
Tanggal dan tempat diumumkan untuk pertama kali di wilayah Indonesia atau di luar wilayah Indonesia	: 8 Juni 2026, di Kota Tegal
Jangka waktu perlindungan	: Berlaku selama 50 (lima puluh) tahun sejak Ciptaan tersebut pertama kali dilakukan Pengumuman.
Nomor Pencatatan	: 001270492
adalah benar berdasarkan keterangan yang diberikan oleh Pemohon. Surat Pencatatan Hak Cipta atau produk Hak terkait ini sesuai dengan Pasal 72 Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.	
	a.n. MENTERI HUKUM DIREKTUR JENDERAL KEKAYAAN INTELEKTUAL u.b Direktur Hak Cipta dan Desain Industri  Agung Damarsasongko,SH.,MH. NIP. 196912261994031001
	Disclaimer: 1. Dalam hal pemohon memberikan keterangan tidak sesuai dengan surat pernyataan, Menteri berwenang untuk mencabut surat pencatatan permohonan. 2. Surat Pencatatan ini telah disegel secara elektronik menggunakan segel elektronik yang diterbitkan oleh Balai Besar Sertifikasi Elektronik, Badan Siber dan Sandi Negara. 3. Surat Pencatatan ini dapat dibuktikan keasliannya dengan memindai kode QR pada dokumen ini dan informasi akan ditampilkan dalam browser.



LAMPIRAN PENCIPTA

No	Nama	Alamat
1	Berliani Risqi Dwi Saputri	Jalan Sumodirjo, RT 002/RW 003, No. 2, Kelurahan Kemandungan Tegal Barat, Kota Tegal
2	Dwi Intan Afidah, S.T., M.Kom	Grinting RT03/RW01 No.45, Kecamatan Bulakamba, Kabupaten Brebes, Provinsi Jawa Tengah, 52253. Bulakamba, Kab. Brebes
3	Arif Hidayah, S.Tr.Kom., M.Tr.Kom	JL. Bulu Emas No. 8, Desa Lawatan, Kecamatan Dukuhturi, Kabupaten Tegal, Provinsi Jawa Tengah, 52192. Dukuhturi, Kab. Tegal





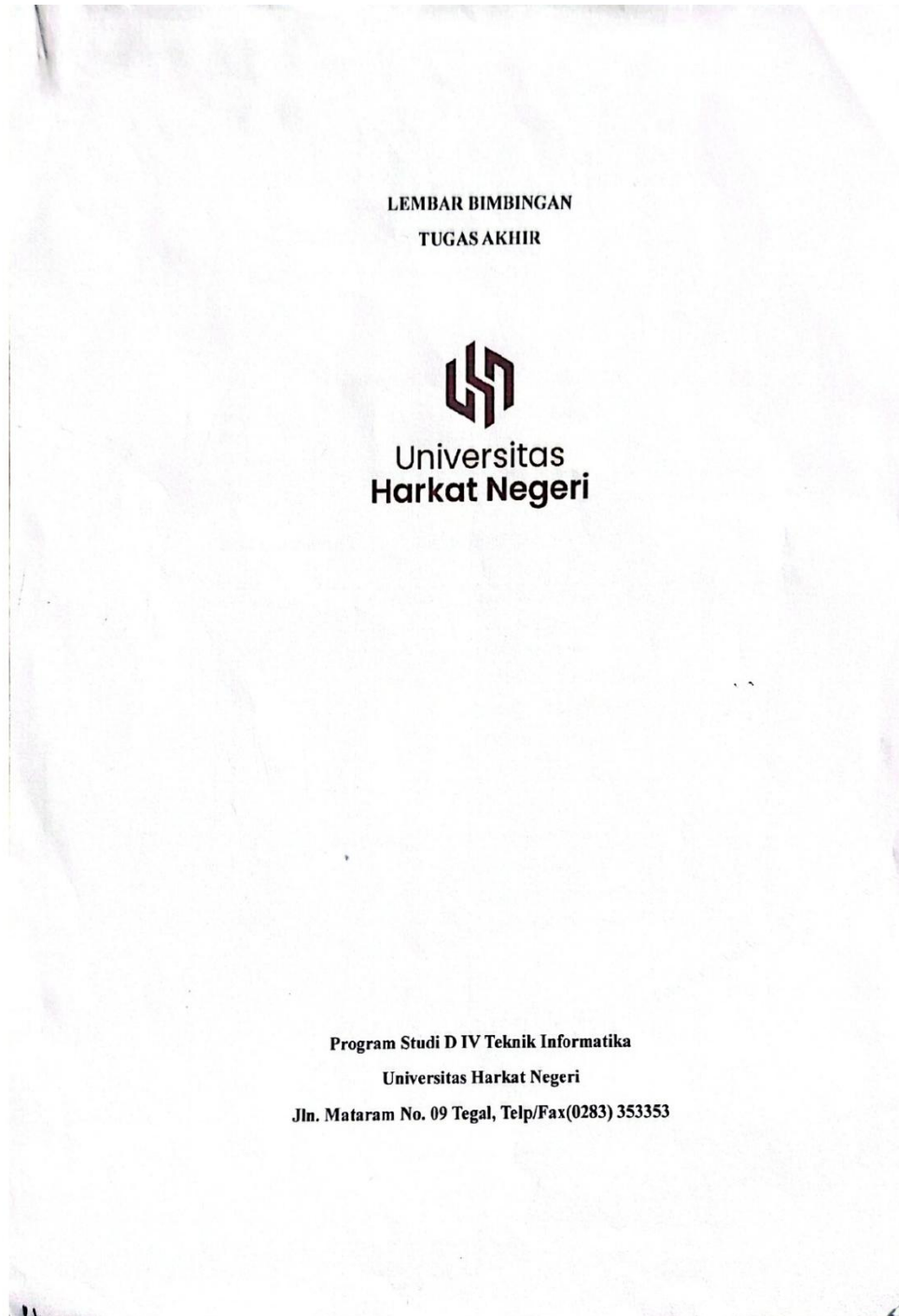


Balai Besar
Sertifikasi
Elektronik

Disclaimer:

1. Dalam hal pemohon memberikan keterangan tidak sesuai dengan surat pernyataan, Menteri berwenang untuk mencabut surat pencatatan permohonan.
2. Surat Pencatatan ini telah disegel secara elektronik menggunakan segel elektronik yang diterbitkan oleh Balai Besar Sertifikasi Elektronik, Badan Siber dan Sandi Negara.
3. Surat Pencatatan ini dapat dibuktikan keasliannya dengan memindai kode QR pada dokumen ini dan informasi akan ditampilkan dalam browser.

Lampiran 7 Lembar Bimbingan Skripsi





D IV TEKNIK INFORMATIKA
UNIVERSITAS HARKAT NEGERI

LEMBAR BIMBINGAN TUGAS AKHIR

Nama : Berliani Risqi Dwi Saputri
 NIM : 22090124
 No. Ponsel : 083116200598
 Judul TA : Aplikasi Mobile Cerdas untuk Asistensi Visual
 Tunanetra Menggunakan Arsitektur Computer
 Vision.
 Dosen Pembimbing 1 : Dwi Intan Af'idah, S.T., M.Kom.

No	Tanggal	Pemeriksaan	Perbaikan Yang Perlu Dilakukan	Paraf Pembimbing
1.	6/4-26	Aplikasi	Dilanjutkan untuk menyesuaikan fitur berdasar parameter	
2.	11/4-26	Aplikasi	Dilanjutkan untuk yang deteksi benda dan cari benda	
3.	21/4-26	Aplikasi	Coba jika memungkinkan menggunakan dataset penelitian lain y objek detection and/or	

4.	5/5-26	Aplikasi	- kalau sudah dibetulkan talkbag, bisa dilanjut pengujian ke tunanetra - dicari bakod paper	
5.	26/5-26	Aplikasi	- sudah sesuai perlu diuji ke anak yang tunanetra - pengujian u/ pengujian hak cipta	
6.	30/6-26	Laporan	dilanjutkan pengujian hak cipta dan pengujian laporan	
7.	22/6-26	Laporan	perbaiki sesuai catatan	
8.	29/6-26	Laporan	ACC	

Tegal, 6 April 2026

Dosen Pembimbing



Dwi Intan Afidha, S.T., M.Kom

NIPY. 11.020.470

LEMBAR BIMBINGAN TUGAS AKHIR

Nama : Berliani Risqi Dwi Saputri
 NIM : 22090124
 No. Ponsel : 083116200598
 Judul TA : Aplikasi Mobile Cerdas untuk Asistensi Visual
 Tunanetra Menggunakan Arsitektur Computer
 Vision.
 Dosen Pembimbing 1 : Arif Hidayah, S.Tr., M.Kom

No	Tanggal	Pemeriksaan	Perbaikan Yang Perlu Dilakukan	Paraf Pembimbing
1	9/4/26	APP	- lanjutkan sesuai catatan	
2.	16/4/26	APP	- fokus user experience	
3.	7/5/26	APP	- coba daemon	
4	25/5/26	APP	- lanjutkan	
5.	4/6/26	APP	- push ke google play	
6.	30/6/26	Laporan	- perbaikan	
		APP	- <u>OK.</u> - <u>HEI OK.</u>	

7.	3/7/26	Laporan APP	OK. - Playstore OK. Daftar sidang /	
----	--------	----------------	---	--

Tegal, 6 April 2026

Dosen Pembimbing



Arif Hidayah, S.Tr., M.Kom
NIPY. 10.024.548