

Pengembangan Aplikasi Teknologi Bantu bagi Penyandang Gangguan Penglihatan Berbasis Deteksi Objek dan *Text-to-Speech*

PROYEK AKHIR

Disusun Oleh:

Willy Fernando

3312301084

Disusun untuk memenuhi salah satu syarat memperoleh gelar Ahli Madya
Teknik Informatika



**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
POLITEKNIK NEGERI BATAM**

2026

KATA PENGANTAR

Puji dan syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa, karena atas berkat dan rahmat-Nya penulis dapat menyelesaikan laporan Proyek Akhir yang berjudul “Pengembangan Aplikasi Teknologi Bantu bagi Penyandang Gangguan Penglihatan Berbasis Deteksi Objek dan *Text-to-Speech*” dengan baik dan tepat pada waktunya.

Penyusunan laporan ini bertujuan untuk memenuhi salah satu syarat dalam menyelesaikan pendidikan pada Program Studi Teknik Informatika Politeknik Negeri Batam. Selain itu, laporan ini juga diharapkan dapat memberikan kontribusi dalam pengembangan teknologi bantu berbasis mobile yang dapat membantu penyandang gangguan penglihatan dalam menjalankan aktivitas sehari-hari.

Dalam proses penyusunan laporan ini, penulis menyadari bahwa banyak pihak yang telah memberikan bantuan, bimbingan, serta dukungan. Oleh karena itu, penulis ingin menyampaikan ucapan terima kasih kepada:

1. Ibu Hilda Widyastuti, S.T., M.T. selaku dosen pembimbing yang telah memberikan arahan, bimbingan, serta motivasi selama proses pengerjaan Proyek Akhir ini.
2. Ibu Ir. Anis Rahmi, S.Tr. Kom., M.Sn. selaku dosen penguji sidang proposal yang telah memberikan masukan dan saran yang membangun.
3. Ibu Dwi Amalia Purnamasari, S.T., M.Cs. selaku dosen penguji sidang proposal yang telah memberikan kritik dan saran demi penyempurnaan laporan ini.
4. Orang tua dan keluarga yang selalu memberikan doa, dukungan, dan semangat kepada penulis.
5. Rekan-rekan dan semua pihak yang telah membantu, baik secara langsung maupun tidak langsung, dalam penyusunan laporan ini.

Penulis menyadari bahwa laporan ini masih jauh dari sempurna. Oleh karena itu, penulis sangat mengharapkan kritik dan saran yang membangun dari berbagai pihak guna perbaikan di masa yang akan datang.

Batam, - April 2026

Penulis,

Willy Fernando 3312301084

DAFTAR ISI

KATA PENGANTAR.....	1
DAFTAR ISI	2
DAFTAR TABEL	4
DAFTAR GAMBAR.....	5
DAFTAR LAMPIRAN	6
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	3
1.3. Tujuan.....	3
1.4. Batasan Masalah	4
1.5. Manfaat.....	4
BAB II TINJAUAN PUSTAKA.....	6
2.1. Penelitian Terkait.....	6
2.2. Landasan Teori.....	7
2.3. Metode Pengembangan Produk	19
BAB III ANALISIS DAN PERANCANGAN.....	23
3.1. Analisis Kebutuhan.....	23
3.2. Perancangan.....	27
BAB IV IMPLEMENTASI DAN PENGUJIAN.....	43
4.1. Implementasi Metode Agile.....	43
4.1.1 Hasil Evaluasi Setiap Iterasi	43
4.2. Hasil Implementasi	44
BAB V KESIMPULAN	55

5.1. Kesimpulan.....	55
5.2. Saran.....	56
5.3. Keterbatasan Penelitian.....	57
DAFTAR PUSTAKA.....	58
DAFTAR LAMPIRAN	60

DAFTAR TABEL

Tabel 1 Perbandingan Penelitian Sebelumnya	6
Tabel 2 Perbandingan Model YOLO	11
Tabel 3 Perbandingan Teknologi OCR.....	12
Tabel 4 Perbandingan Teknologi TTS	15
Tabel 5 Prinsip Aksesibilitas Aplikasi Mobile	17
Tabel 6 Pengguna Low Vision.....	23
Tabel 7 Pengguna Tunanetra Total	24
Tabel 8 Kebutuhan Fungsional	27
Tabel 9 Non-Fungsional.....	27
Tabel 10 Scenario Usecase (Deteksi Objek via Kamera)	30
Tabel 11 Scenario Usecase (Pengenalan Teks (OCR) & Konversi Teks ke Suara (TTS))	31
Tabel 12 Scenario Usecase (Menyimpan & Mengakses Data).....	31
Tabel 13 Scenario User (Menggunakan TalkBack & Aksesibilitas)	32
Tabel 14 Implementasi Sprint Agile	43
Tabel 15 Black Box Testing	53

DAFTAR GAMBAR

Gambar 1 Ilustrasi Deteksi Objek.....	7
Gambar 2 Ilustrasi Algoritma Yolo	9
Gambar 3 Diagram ALgoritma Yolo.....	9
Gambar 4 Algoritma Piper TTS.....	14
Gambar 5 Metode Agile.....	19
Gambar 6 Gambaran Umum Sistem	26
Gambar 7 Usecase Diagram.....	29
Gambar 8 Activity Deteksi Objek.....	33
Gambar 9 Activity Pindai Teks (OCR).....	34
Gambar 10 Activity Perpustakaan	35
Gambar 11 Activity Pengaturan.....	36
Gambar 12 Activity Impor File.....	37
Gambar 13 ER-Diagram sistem	38
Gambar 14 Halaman Utama (Beranda).....	39
Gambar 15 Halaman Perpustakaan.	40
Gambar 16 Halaman Impor File	41
Gambar 17 Halaman Pengaturan	42
Gambar 18 Halaman Utama (Beranda).....	47
Gambar 19 Fitur Deteksi Objek.....	48
Gambar 20 Fitur Pindai Teks (OCR).....	48
Gambar 21 Fitur Perpustakaan.....	49
Gambar 22 Fitur Impor File.....	49
Gambar 23 Fitur Pengaturan.....	50

DAFTAR LAMPIRAN

Lampiran A. Link Produk	60
-------------------------------	----

ABSTRAK

Perkembangan teknologi informasi memberikan peluang besar dalam menciptakan solusi yang inklusif bagi penyandang gangguan penglihatan, terutama dalam membantu aktivitas sehari-hari yang berkaitan dengan pengenalan lingkungan dan akses informasi visual. Proyek akhir ini bertujuan untuk mengembangkan aplikasi teknologi bantu berbasis mobile yang mampu membantu penyandang gangguan penglihatan melalui fitur deteksi objek, pengenalan teks (*Optical Character Recognition/OCR*), dan konversi teks ke suara (*Text-to-Speech/TTS*). Metode yang digunakan dalam pengembangan sistem meliputi tahapan analisis kebutuhan, perancangan sistem, implementasi, dan pengujian. Teknologi yang diterapkan antara lain model deteksi objek berbasis YOLO, OCR menggunakan *Tesseract*, serta TTS menggunakan Piper yang diintegrasikan dalam aplikasi Android. Hasil dari proyek ini berupa aplikasi yang mampu mendeteksi objek secara *real-time*, membaca teks dari gambar, serta menyampaikan informasi dalam bentuk audio kepada pengguna. Berdasarkan hasil pengujian, sistem dapat berjalan dengan baik dan memberikan kemudahan bagi pengguna dalam mengenali lingkungan serta mengakses informasi visual. Dengan demikian, aplikasi yang dikembangkan diharapkan dapat meningkatkan kemandirian dan kualitas hidup penyandang gangguan penglihatan dalam menjalankan aktivitas sehari-hari.

Kata Kunci: teknologi bantu, deteksi objek, OCR, *text-to-speech*, Android

BAB I PENDAHULUAN

1.1. Latar Belakang

Teknologi merupakan salah satu aspek penting dalam meningkatkan kualitas hidup manusia, termasuk bagi penyandang disabilitas. Perkembangan teknologi yang pesat di era digital saat ini tidak hanya membawa kemudahan bagi masyarakat umum, tetapi juga memberikan peluang besar dalam menciptakan solusi yang inklusif bagi kelompok dengan kebutuhan khusus. Salah satu kelompok yang sangat membutuhkan dukungan teknologi adalah penyandang gangguan penglihatan. Mereka sering kali menghadapi berbagai keterbatasan dalam menjalankan aktivitas sehari-hari, seperti membaca, mengenali lingkungan sekitar, berpindah tempat, dan berinteraksi sosial. Tercatat sekitar 43,3 juta orang di dunia mengalami kebutaan total dan sekitar 295 juta orang mengalami gangguan penglihatan sedang hingga berat (WHO, 2020).

Pada penelitian ini, fokus pengguna ditujukan kepada penyandang gangguan penglihatan kategori tunanetra dan *low vision* usia produktif (18–60 tahun). Kelompok ini dipilih karena memiliki tingkat mobilitas yang tinggi dalam aktivitas sehari-hari seperti bekerja, belajar, berbelanja, dan menggunakan fasilitas umum. Keterbatasan dalam mengenali objek di lingkungan sekitar maupun membaca informasi visual sering menjadi hambatan yang dapat mempengaruhi kemandirian dan keselamatan pengguna.

Kondisi ini berdampak signifikan terhadap kemandirian dan partisipasi mereka dalam berbagai aspek kehidupan, termasuk pendidikan, pekerjaan, serta akses terhadap informasi dan layanan publik. Di Indonesia sendiri, berdasarkan laporan Kementerian Koordinator Bidang Pembangunan Manusia dan Kebudayaan (Kemenko PMK, 2023), jumlah penyandang disabilitas penglihatan mencapai 22,97 juta jiwa atau sekitar 8,5% dari total populasi penduduk di Indonesia. Dari jumlah tersebut, kelompok usia lanjut mendominasi proporsi penyandang disabilitas penglihatan tertinggi.

Selama ini, penyandang gangguan penglihatan mengandalkan teknologi bantu sederhana seperti tongkat putih (*white cane*), perangkat *braille*, atau panduan dari orang lain dalam melakukan aktivitas sehari-hari. Meskipun alat bantu tersebut berfungsi cukup efektif, penggunaannya masih memiliki keterbatasan, terutama dalam mendeteksi

rintangan secara *real-time*, mengenali lingkungan yang dinamis, serta membaca informasi visual yang semakin banyak digunakan di era digital (Marito, Y, *et al*, 2024). Dengan perkembangan teknologi informasi dan komunikasi, muncul peluang besar untuk menghadirkan inovasi baru berupa *assistive technology* atau teknologi bantu yang lebih canggih, efektif, dan mudah diakses oleh masyarakat luas.

Assistive Technology atau teknologi bantu adalah teknologi yang dirancang untuk meningkatkan kemampuan fungsional individu yang memiliki keterbatasan fisik maupun sensorik. Menurut WHO (2022), *assistive technology* mencakup perangkat, perangkat lunak, maupun sistem yang membantu penyandang disabilitas menjalankan aktivitas sehari-hari secara lebih mandiri. Dalam konteks penyandang gangguan penglihatan, teknologi bantu dapat berupa pembaca layar, sistem navigasi berbasis suara, deteksi objek, hingga pembacaan teks otomatis.

Berdasarkan studi literatur dan analisis kebutuhan pengguna, terdapat tiga kebutuhan utama yang paling sering dihadapi penyandang gangguan penglihatan, yaitu mengenali objek di sekitar secara *real-time*, membaca informasi berbasis teks, serta memperoleh informasi dalam bentuk suara. Oleh karena itu, penelitian ini berfokus pada tiga fitur utama, yaitu deteksi objek menggunakan YOLO, *Optical Character Recognition* (OCR), dan *Text-to-Speech* (TTS).

Kebutuhan untuk mengenali objek di sekitar dapat dipenuhi melalui teknologi deteksi objek berbasis kecerdasan buatan yang mampu memberikan informasi mengenai keberadaan benda atau rintangan di lingkungan pengguna. Sementara itu, kebutuhan membaca informasi visual dapat didukung oleh teknologi *Optical Character Recognition* (OCR) yang mampu mengubah teks pada gambar menjadi teks digital. Agar informasi tersebut dapat diakses oleh penyandang gangguan penglihatan, hasil deteksi objek maupun OCR selanjutnya disampaikan dalam bentuk suara menggunakan teknologi *Text-to-Speech* (TTS).

Oleh karena itu, penelitian ini bertujuan mengembangkan aplikasi teknologi bantu berbasis Android yang mengintegrasikan deteksi objek, *Optical Character Recognition* (OCR), dan *Text-to-Speech* (TTS) untuk membantu penyandang tunanetra dan *low vision* memperoleh informasi mengenai objek di sekitar dan informasi berbasis teks secara lebih mandiri.

1.2. Rumusan Masalah

Berdasarkan latar belakang di atas, maka rumusan masalah yang dapat disusun adalah sebagai berikut:

- 1 Bagaimana mengembangkan aplikasi mobile berbasis Android yang dapat membantu penyandang gangguan penglihatan melalui fitur deteksi objek menggunakan model YOLO?
- 2 Bagaimana mengimplementasikan *Optical Character Recognition* (OCR) untuk mengenali teks dari gambar yang ditangkap oleh kamera atau dipilih dari penyimpanan perangkat?
- 3 Bagaimana mengintegrasikan teknologi *Text-to-Speech* (TTS) sehingga hasil deteksi objek dan OCR dapat disampaikan dalam bentuk suara kepada pengguna?
- 4 Bagaimana menguji dan mengevaluasi fungsionalitas aplikasi yang dikembangkan agar sesuai dengan kebutuhan pengguna?

1.3. Tujuan

Tujuan dari penyusunan proyek ini adalah:

- 1 Mengembangkan aplikasi teknologi bantu berbasis Android yang dapat membantu penyandang gangguan penglihatan dalam mengenali objek di sekitar menggunakan model YOLO.
- 2 Mengimplementasikan fitur *Optical Character Recognition* (OCR) untuk mengenali teks dari gambar dan mengubahnya menjadi informasi yang dapat diakses pengguna.
- 3 Mengintegrasikan teknologi *Text-to-Speech* (TTS) sehingga hasil deteksi objek dan OCR dapat disampaikan dalam bentuk suara.
- 4 Melakukan pengujian dan evaluasi terhadap fungsionalitas aplikasi menggunakan metode Black Box Testing.

1.4. Batasan Masalah

Agar penelitian lebih terarah, maka batasan masalah pada proyek ini adalah:

- 1 Teknologi bantu yang dikembangkan difokuskan pada penyandang gangguan penglihatan (*low vision* maupun tunanetra).
- 2 Lingkup penggunaan teknologi bantu dibatasi pada aktivitas sehari-hari tertentu, seperti navigasi sederhana, membaca informasi, atau interaksi lingkungan dengan fitur yang dikembangkan meliputi: deteksi objek *real-time*, pengenalan teks (OCR), dan *text-to-speech*
- 3 Teknologi yang dirancang bersifat prototipe sehingga masih memerlukan pengembangan lebih lanjut untuk implementasi luas.
- 4 Evaluasi efektivitas dilakukan secara terbatas pada kelompok kecil pengguna uji coba.
- 5 Hanya untuk jenis *smartphone android*

1.5. Manfaat

Penelitian ini diharapkan memberikan berbagai manfaat bagi berbagai pihak. Bagi penyandang gangguan penglihatan, penelitian ini dapat membantu meningkatkan kemandirian dalam menjalankan aktivitas sehari-hari melalui pemanfaatan teknologi bantu yang inovatif. Dengan adanya teknologi yang dirancang khusus untuk kebutuhan mereka, penyandang disabilitas netra dapat memperoleh akses yang lebih mudah terhadap informasi, lingkungan sekitar, serta peluang untuk berpartisipasi aktif dalam kehidupan sosial dan ekonomi.

Bagi peneliti dan akademisi, hasil penelitian ini dapat menjadi wadah pengembangan inovasi dalam bidang *assistive technology*, khususnya dalam konteks penerapan teknologi informasi untuk mendukung penyandang disabilitas. Penelitian ini juga diharapkan memberikan kontribusi ilmiah yang signifikan bagi pengembangan literatur dan referensi akademik, serta mendorong penelitian lanjutan yang berfokus pada kesetaraan dan inklusivitas bagi semua kalangan.

Sementara itu, bagi masyarakat dan pemerintah, penelitian ini memiliki manfaat dalam meningkatkan kesadaran tentang pentingnya inklusi sosial bagi penyandang disabilitas. Selain itu, hasil penelitian ini dapat dijadikan sebagai bahan pertimbangan dan rekomendasi bagi pengambilan kebijakan, perancangan fasilitas publik, serta pengembangan lingkungan yang lebih ramah dan aksesibel bagi semua individu, termasuk penyandang disabilitas. Dengan demikian, penelitian ini tidak hanya memberikan dampak pada aspek teknologi, tetapi juga berkontribusi terhadap terwujudnya masyarakat yang lebih inklusif dan berkeadilan sosial.

Penelitian ini juga mendukung pencapaian Tujuan Pembangunan Berkelanjutan (*Sustainable Development Goals / SDGs*), khususnya SDG 3 (*Good Health and Well-Being*) melalui upaya peningkatan kualitas hidup dan kemandirian penyandang gangguan penglihatan dalam menjalankan aktivitas sehari-hari. Selain itu, penelitian ini berkontribusi terhadap SDG 10 (*Reduced Inequalities*) dengan mendorong pemanfaatan teknologi yang inklusif dan aksesibel bagi penyandang disabilitas sehingga dapat membantu mengurangi kesenjangan akses terhadap informasi, teknologi, dan layanan publik. Dengan demikian, penelitian ini tidak hanya memberikan manfaat dalam aspek teknologi, tetapi juga mendukung terciptanya masyarakat yang lebih inklusif, setara, dan berkelanjutan.

BAB II TINJAUAN PUSTAKA

2.1. Penelitian Terkait

Dalam sebuah penelitian, landasan teori atau penelitian terkait merupakan peran penting dalam memperkuat dasar penelitian. Untuk mendukung pembuatan proposal ini, maka perlu di kemukakan teori-teori yang berkaitan dengan permasalahan dan ruang lingkup pembahasan sebagai landasan dalam pembuatan proposal. Hal-hal yang mendasari dibuatnya teknologi bantu bagi penyandang gangguan penglihatan berbasis mobile ini adalah sebagai berikut.

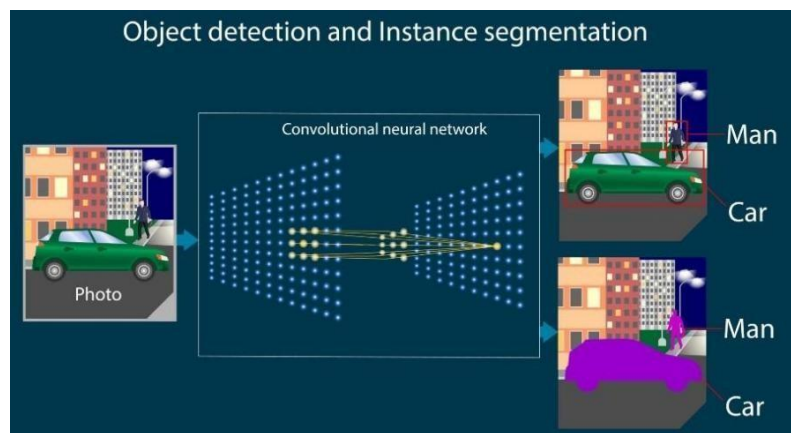
Tabel 1 Perbandingan Penelitian Sebelumnya

Peneliti /Tahun	Judul/Topik Penelitian	Metode/Teknologi	Kelebihan	Kekurangan/Celah
Luqman Hidayat (2022)	<i>Assistive Technology</i> pada Aplikasi Android untuk Tunanetra	TalkBack, TapTapSee, Eye-D, Mas Jawa T-Netra, dan Google Goggles	Relevan dan humanis, Memberikan kontribusi sosial, dan komprehensif	Kurang spesifik dalam metodologi teknis dan Belum membahas tantangan lokal
Amka, S.Pd., M.Pd. (2021)	Pengembangan Mobile Application Berbasis Android sebagai Alat Identifikasi Peserta Didik Berkebutuhan Khusus di Lingkungan Lahan Basah	ADDIE (Analysis, Design, Development, Implementation, Evaluation).	Inovatif, Berbasis model pengembangan sistematis (ADDIE) dan Memadukan unsur teknologi dan pendidikan khusus	Uji coba masih terbatas dan Belum ada evaluasi jangka panjang
Syah Putra (2024)	Pengembangan Aplikasi Android untuk Membantu Penyandang Disabilitas Netra dalam Beraktivitas Sehari-hari	mencakup fitur navigasi berbasis suara, pengenalan objek dengan kamera (AI & OpenCV/TensorFlow), serta pembacaan teks menggunakan OCR (Optical Character Recognition).	Menggunakan pendekatan Agile dan Berbasis teknologi terkini, seperti OCR, AI vision, dan voice navigation.	Belum dilakukan uji efektivitas jangka panjang dan Fokus masih pada prototipe awal

2.2. Landasan Teori

2.2.1 Deteksi Objek

Deteksi objek adalah tugas dalam bidang penglihatan komputer (*computer vision*) yang melibatkan identifikasi dan pelokasian instans dari objek-objek semantik tertentu (seperti manusia, mobil, atau kucing) dalam gambar atau video digital. Tujuan utamanya adalah untuk menentukan di mana objek-objek tersebut berada dalam citra dan kelas apa objek-objek tersebut (Marpaung et al., 2022)



Gambar 1 Ilustrasi Deteksi Objek

Sumber: Penulis

Berdasarkan gambar 1 proses *object detection* dan *instance segmentation* menggunakan convolutional neural network (CNN) dalam bidang *computer vision*. Foto input yang menampilkan sebuah mobil hijau dan seorang pria di jalanan kota dimasukkan ke dalam jaringan CNN, yang terdiri dari lapisan-lapisan konvolusi berbentuk corong untuk mengekstrak fitur-fitur visual secara hierarkis. Hasil keluaran dari CNN kemudian digunakan untuk dua tugas utama: *object detection* yang menghasilkan *bounding box* di sekitar objek dengan label kelas (seperti "Man" dan "Car" pada mobil dan pria), serta *instance segmentation* yang memberikan segmentasi piksel-per-piksel untuk setiap instance objek secara terpisah, sehingga mobil dan pria ditampilkan dalam mask berwarna ungu yang membedakan bentuk dan batas presisi masing-masing objek dari latar belakang.

Proses deteksi objek umumnya menghasilkan kotak pembatas (*bounding box*) di sekitar setiap objek yang terdeteksi, bersama dengan label kelas yang memprediksi jenis objek tersebut dan skor keyakinan (*confidence score*). Teknologi ini sangat penting

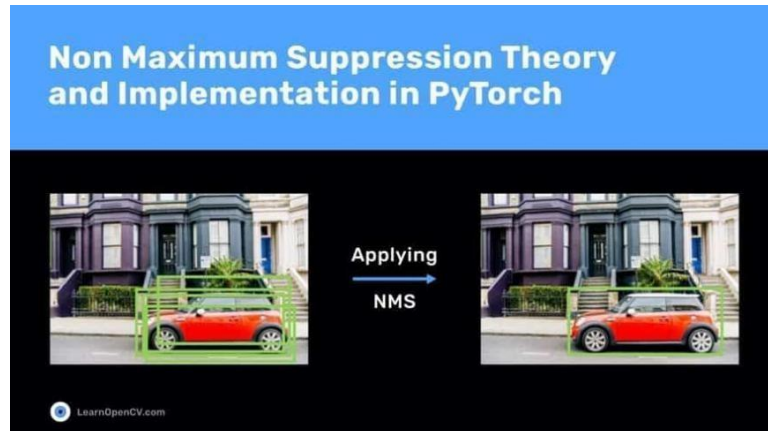
sebagai fondasi bagi sistem bantu penyandang gangguan penglihatan, memungkinkan aplikasi untuk "melihat" dan memahami lingkungan di sekitar pengguna (Marpaung et al., 2022).

2.2.2 Algoritma Yolo

YOLO (*You Only Look Once*) adalah salah satu algoritma deteksi objek *real-time* yang paling terkenal dan efisien. Algoritma ini merevolusi bidang deteksi objek karena kemampuannya untuk melakukan inferensi dengan kecepatan tinggi, menjadikannya ideal untuk aplikasi *mobile* atau sistem *live-feed*.

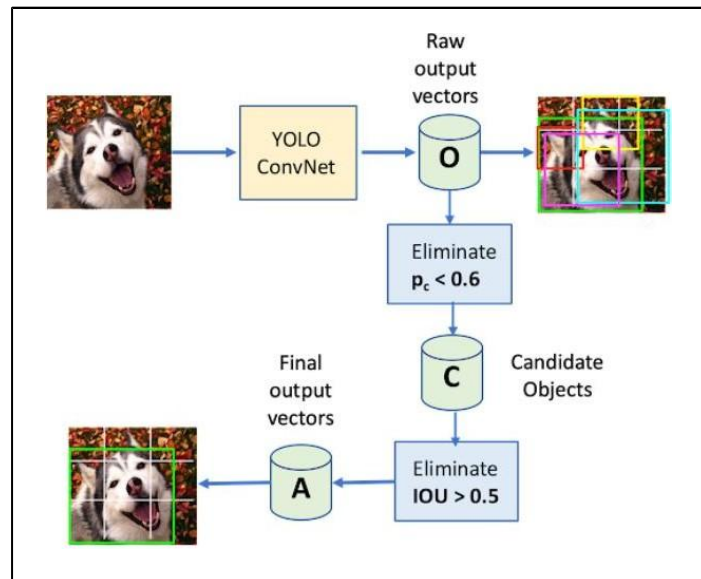
Tidak seperti metode deteksi objek tradisional yang memisahkan proses pelokasian (region proposal) dan klasifikasi, YOLO mendekati deteksi objek sebagai masalah regresi tunggal. Ini memungkinkan YOLO untuk memprediksi kotak pembatas (*bounding box*) dan probabilitas kelas secara langsung dari piksel gambar dalam satu run jaringan saraf, menjadikannya sangat cepat. Proses dimulai dengan membagi gambar masukan menjadi $S \times S$ *grid sel*. Setiap sel grid bertanggung jawab untuk memprediksi objek yang pusatnya jatuh ke dalam sel tersebut. Kemudian, setiap sel grid memprediksi B kotak pembatas, skor keyakinan (*confidence scores*), dan C probabilitas kelas kondisional (conditional class probabilities). Setiap kotak pembatas (B) mendefinisikan 5 prediksi: koordinat pusat relatif sel (x, y), tinggi (h), lebar (w), dan skor objek (*objectness score*). Skor keyakinan mencerminkan keyakinan model bahwa kotak tersebut berisi objek dan seberapa akurat kotak pembatas diprediksi, dihitung sebagai $P(\text{Object}) \times \text{IOU}$.

Setelah prediksi awal, tahap pasca-pemrosesan yang disebut *Non-Max Suppression* (NMS) dilakukan untuk menghilangkan duplikat kotak pembatas yang tumpang tindih untuk objek yang sama, hanya menyisakan kotak dengan skor kelas tertinggi (Jatin Prakash, 2021)



Gambar 2 Ilustrasi Algoritma Yolo

Sumber:LearnOpenCv.Com



Gambar 3 Diagram ALgoritma Yolo

Diagram alir yang disajikan merupakan representasi visual dari tahapan inti dalam proses deteksi objek menggunakan algoritma YOLO (You Only Look Once). Algoritma ini dikenal karena kecepatan dan akurasi yang tinggi dalam pemrosesan gambar secara *real-time*. Proses dimulai dengan masukan berupa gambar digital yang kemudian diproses melalui YOLO ConvNet (Convolutional Network).

Tahap Pemrosesan Jaringan dan *Raw Output* adalah Jaringan saraf konvolusional

(ConvNet) yang menjadi tulang punggung YOLO bertugas untuk memproses seluruh gambar secara simultan, bukan per bagian. Jaringan ini membagi gambar menjadi sebuah *grid* dan memprediksi kotak batas (*bounding box*), skor kepercayaan objek (*confidence score*), dan probabilitas kelas untuk setiap *grid cell*. Hasil dari pemrosesan ini adalah kumpulan Vektor *Output Mentah* (O) yang sangat banyak, mengandung prediksi untuk berbagai potensi objek di seluruh *grid*.

Tahap Eliminasi Awal: Penyaringan Berdasarkan *Confidence Score* adalah Tahap selanjutnya adalah proses penyaringan awal untuk mengurangi jumlah prediksi yang tidak relevan. Dalam diagram, langkah ini ditunjukkan dengan kotak eliminasi yang menghilangkan semua vektor prediksi di mana Probabilitas Kepercayaan Objek (p) kurang dari 0.6 merefleksikan keyakinan model bahwa kotak batas tersebut benar-benar mengandung objek. Setelah eliminasi ini, hanya prediksi dengan kepercayaan yang cukup tinggi yang dipertahankan, menghasilkan kumpulan Objek Kandidat (C).

Tahap Eliminasi Akhir: *Non-Maximum Suppression* (NMS) adalah Kumpulan Objek Kandidat (C) seringkali masih mengandung banyak *bounding box* yang tumpang tindih (*overlapping*) yang semuanya memprediksi objek yang sama. Untuk mengatasi redundansi ini, diterapkan mekanisme Eliminasi Berdasarkan IOU (Intersection Over Union) > 0.5 . Proses ini secara efektif mengimplementasikan Non-Maximum Suppression (NMS). NMS bekerja dengan memilih *bounding box* dengan skor kepercayaan tertinggi dan kemudian menghapus semua *box* lain yang memiliki IOU (rasio tumpang tindih) yang melebihi ambang batas (dalam kasus ini, 0.5) terhadap *box* yang terpilih. Tujuannya adalah memastikan bahwa setiap objek hanya diwakili oleh satu *bounding box* terbaik. Hasil akhir dari proses ini adalah kumpulan Vektor *Output Final* (A), yang merupakan prediksi deteksi objek yang telah disempurnakan dan siap untuk divisualisasikan (Prakash, 2021; Redmon et al., 2016).

Dalam penelitian ini digunakan YOLOv5n (Nano) sebagai model deteksi objek. Pemilihan model ini didasarkan pada pertimbangan keterbatasan sumber daya perangkat mobile yang digunakan sebagai media implementasi aplikasi. Dibandingkan dengan varian YOLO yang lebih besar seperti YOLOv5s maupun model terbaru seperti YOLOv8n, YOLOv5n memiliki ukuran model yang lebih kecil serta kebutuhan memori yang lebih rendah sehingga mampu memberikan kecepatan inferensi yang lebih baik pada perangkat Android. Walaupun akurasi YOLOv5n sedikit lebih rendah dibandingkan YOLOv5s maupun YOLOv8n, perbedaan tersebut masih dapat

diterima untuk kebutuhan aplikasi bantu real-time bagi penyandang gangguan penglihatan, yang lebih mengutamakan respons cepat dibanding peningkatan akurasi yang relatif kecil.

Tabel 2 Perbandingan Model YOLO

Aspek	YOLOv5n	YOLOv5s	YOLOv8n
Ukuran model	±3,9 MB	±14 MB	±6 MB
mAP COCO (50-95)	±28%	±37%	±37%
Kecepatan inferensi mobile	Sangat tinggi	Tinggi	Tinggi
Kebutuhan RAM	Rendah	Sedang	Sedang
Cocok untuk perangkat mobile	Sangat cocok	Cukup cocok	Cocok

Alasan pemilihan: YOLOv5n dipilih karena memiliki ukuran model paling kecil, penggunaan RAM yang rendah, serta mampu melakukan inferensi secara real-time pada perangkat Android sehingga sesuai dengan kebutuhan aplikasi penelitian ini.

2.2.3 Optical Character Recognition (OCR)

Optical Character Recognition (OCR) merupakan teknologi yang digunakan untuk mengenali dan mengubah teks yang terdapat pada gambar, dokumen hasil pemindaian, maupun hasil tangkapan kamera menjadi data teks digital yang dapat diproses oleh komputer (Patel et al., 2012). Teknologi OCR banyak digunakan dalam berbagai aplikasi seperti digitalisasi dokumen, pembacaan plat nomor kendaraan, penerjemahan teks, hingga teknologi bantu bagi penyandang disabilitas.

Proses OCR umumnya terdiri dari beberapa tahapan, yaitu akuisisi citra, pra-pemrosesan (pre-processing), segmentasi karakter, ekstraksi fitur, dan pengenalan karakter. Pada tahap pra-pemrosesan dilakukan peningkatan kualitas gambar, seperti pengurangan noise, konversi ke grayscale, dan penyesuaian kontras agar teks lebih mudah dikenali. Selanjutnya sistem melakukan segmentasi untuk memisahkan karakter-karakter yang terdapat pada gambar. Setelah itu, fitur dari setiap karakter diekstraksi dan dibandingkan dengan pola yang telah dipelajari oleh model OCR untuk menentukan karakter yang sesuai.

Dalam penelitian ini, OCR digunakan untuk membantu penyandang gangguan penglihatan dalam membaca informasi berbentuk teks yang terdapat pada lingkungan sekitar. Hasil teks yang berhasil dikenali kemudian diproses menggunakan teknologi *Text-to-Speech* (TTS) sehingga dapat disampaikan kepada pengguna dalam bentuk suara (Patel et al., 2012).

Berbagai teknologi OCR tersedia untuk mengenali teks pada gambar, seperti *Tesseract OCR*, *Google ML Kit Text Recognition*, dan *EasyOCR*. Dalam penelitian ini dipilih *Tesseract OCR* karena bersifat open-source, dapat dijalankan sepenuhnya secara offline, serta memiliki ukuran pustaka yang relatif ringan sehingga sesuai untuk implementasi pada perangkat mobile tanpa ketergantungan terhadap layanan cloud. Selain itu, *Tesseract* telah mendukung berbagai bahasa dan mudah diintegrasikan dengan Flutter melalui plugin yang tersedia.

Tabel 3 Perbandingan Teknologi OCR

Aspek	Tesseract OCR	ML Kit Text Recognition	EasyOCR
Open Source	Ya	Tidak	Ya
Offline	Ya	Ya	Ya
Ukuran Library	Kecil	Sedang	Besar
Akurasi teks tidak terstruktur	Baik	Sangat Baik	Sangat Baik
Integrasi Flutter	Mudah	Mudah	Cukup kompleks
Kebutuhan RAM	Rendah	Sedang	Tinggi

Alasan pemilihan: *Tesseract* dipilih karena memiliki keseimbangan antara ukuran library, kemampuan bekerja secara offline, konsumsi sumber daya yang rendah, serta kemudahan integrasi dengan aplikasi Flutter.

2.2.4 Text To Speech

Text to Speech (TTS), atau Sintesis Suara, adalah teknologi vital yang berfungsi untuk mengubah teks bahasa manusia menjadi ucapan atau suara yang dapat didengar (Md Monsur Ali, 2024). Dalam konteks pengembangan teknologi bantu bagi penyandang gangguan penglihatan, *text to speech* menjadi antarmuka utama yang efektif untuk menyampaikan informasi visual yang telah dideteksi dan diklasifikasikan oleh algoritma seperti *YOLO*, sehingga memungkinkan pengguna untuk memahami lingkungan mereka.

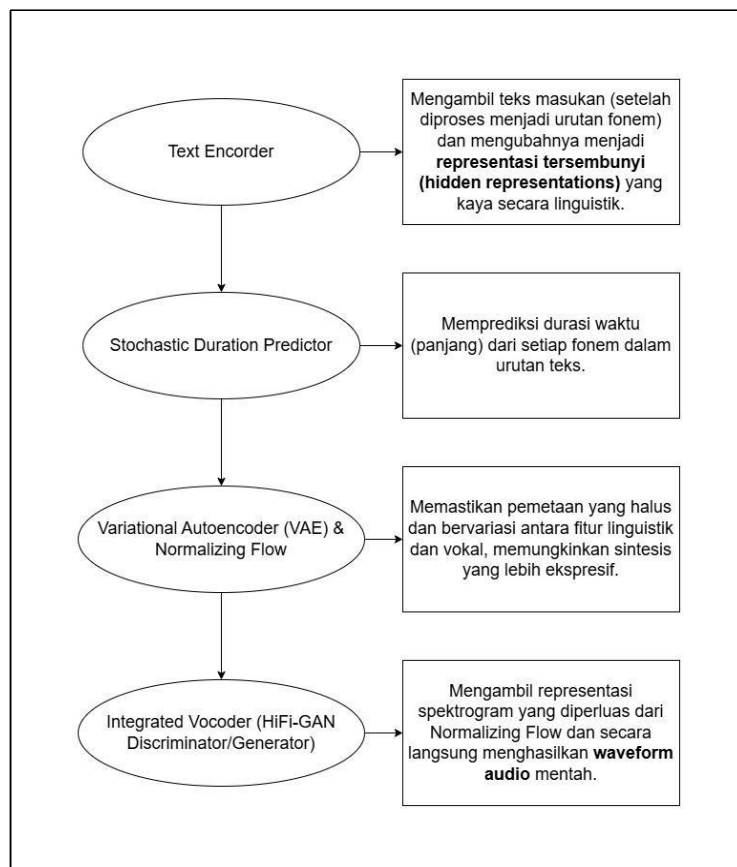
Sistem *text to speech* beroperasi melalui dua komponen utama. Pertama, Analisis Teks memproses input berupa teks mentah. Tahap ini meliputi normalisasi (mengubah angka, singkatan, dan simbol menjadi kata-kata lengkap), segmentasi kalimat, dan menentukan pelafalan yang benar (misalnya menggunakan kamus atau model *grapheme-to-phoneme*). Kedua, Sintesis Suara adalah tahap inti yang mengambil representasi linguistik yang sudah diproses dan menghasilkan bentuk gelombang audio yang sesuai, yang dapat dilakukan menggunakan metode berbasis konkatenasi (menggabungkan unit suara yang direkam) atau, yang lebih canggih, menggunakan model *deep learning* seperti *Tacotron* dan *Vocoder* untuk menghasilkan suara yang lebih alami dan ekspresif (Md Monsur Ali, 2024).

2.2.5 Piper TTS

Piper TTS merupakan implementasi *open-source* yang cepat dan ringan dari sistem VITS (*Variational Inference for Text-to-Speech*), menjadikannya algoritma ideal untuk aplikasi berbasis *mobile* atau perangkat dengan sumber daya terbatas. Berbeda dengan model dua tahap seperti Tacotron/WaveNet, Piper adalah sistem *End-to-End* yang mengintegrasikan semua langkah sintesis suara menjadi satu jaringan tunggal, yang secara langsung menghasilkan *waveform* audio dari teks masukan. Model ini memanfaatkan struktur *Variational Autoencoder* (VAE) dan *Normalizing Flow* untuk memodelkan hubungan antara teks dan fitur akustik, serta secara eksplisit menyertakan *Duration Predictor* untuk memastikan ritme bicara yang alami. Keunggulan utamanya terletak pada penggunaan *Generative Adversarial Network* (GAN) (seperti HiFi-GAN) sebagai *vocoder* terintegrasi, yang menghasilkan kecepatan sintesis yang sangat tinggi (seringkali lebih cepat dari *real-time*) dan kualitas suara yang realistis, sekaligus menjaga ukuran *file* model tetap kecil (Md

Monsur Ali, 2024)

Sebagai sistem End-to-End, Piper, melalui arsitektur VITS-nya, menawarkan keunggulan signifikan dibandingkan model TTS generasi sebelumnya seperti Tacotron 2 yang memerlukan dua jaringan terpisah (generator spektrogram dan vocoder terpisah). Dalam arsitektur *two-stage*, kesalahan atau ketidaksempurnaan pada keluaran spektrogram dapat terpropagasi ke *vocoder*, yang pada akhirnya menurunkan kualitas suara yang dihasilkan. Sebaliknya, Piper mengintegrasikan semua fungsi—dari pemrosesan teks, durasi *phoneme*, hingga sintesis *waveform*—ke dalam satu jaringan tunggal yang dilatih bersama-sama (*jointly trained*). Hal ini tidak hanya menyederhanakan *pipeline* implementasi tetapi juga menghasilkan suara yang lebih konsisten dan alami, dengan peningkatan signifikan pada kecepatan inferensi karena tidak adanya *bottleneck* transfer data antara dua model terpisah. Peningkatan efisiensi ini sangat penting, memungkinkan aplikasi bantu berbasis *mobile* untuk memberikan *feedback* audio secara *real-time* setelah deteksi objek oleh YOLO.



Gambar 4 Algoritma Piper TTS

Sumber: Penulis berdasarkan Ali (2024)

Proses sintesis ucapan modern diawali dengan Text Encoder, yang bertugas mengambil teks masukan, memprosesnya menjadi urutan fonem, dan mengubahnya menjadi representasi tersembunyi (*hidden representations*) yang kaya secara linguistik. Representasi ini kemudian diteruskan ke Stochastic Duration Predictor yang memperkirakan durasi waktu (panjang) yang bervariasi secara alami dari setiap fonem dalam urutan teks, meniru ritme ucapan manusia. Untuk mencapai sintesis yang ekspresif, sistem menggunakan gabungan Variational Autoencoder (VAE) & Normalizing Flow untuk memastikan pemetaan yang halus dan variatif antara fitur linguistik dan vokal. Tahap akhir adalah Integrated Vocoder, yang dalam diagram ini diwakili oleh arsitektur HiFi-GAN, yang mengambil representasi spektrogram yang diperluas dari tahap sebelumnya dan secara langsung menghasilkan *waveform audio* mentah untuk menghasilkan keluaran suara yang berakurasi tinggi dan berkualitas (Md Monsur Ali, 2024).

Selain Piper TTS, alternatif lain adalah Android Text-to-Speech (Google TTS) dan Coqui TTS. Android TTS mudah digunakan karena tersedia secara bawaan, tetapi bergantung pada layanan Google dan konfigurasi perangkat. Coqui TTS menghasilkan suara berkualitas tinggi, namun membutuhkan model dan sumber daya yang lebih besar. Piper TTS dipilih karena mendukung penggunaan offline, memiliki ukuran model yang ringan, kecepatan inferensi tinggi, serta menghasilkan suara yang alami sehingga cocok untuk aplikasi mobile real-time.

Tabel 4 Perbandingan Teknologi TTS

Aspek	Piper TTS	Android TTS	Coqui TTS
Open Source	Ya	Tidak	Ya
Offline	Ya	Bergantung perangkat	Ya
Ukuran Model	Kecil	Sangat kecil	Besar
Kualitas Suara	Sangat Baik	Baik	Sangat Baik
Kecepatan Inferensi	Sangat tinggi	Tinggi	Sedang
Cocok untuk Mobile	Sangat cocok	Cocok	Kurang cocok

Alasan pemilihan: Piper dipilih karena mampu menghasilkan suara yang alami dengan ukuran model yang ringan, mendukung pemrosesan offline, serta memberikan performa inferensi yang cepat sehingga sesuai dengan kebutuhan aplikasi bantu berbasis Android.

2.2.6 Flutter

Flutter merupakan *framework open-source* yang dikembangkan oleh Google untuk membangun aplikasi mobile, web, dan desktop menggunakan satu basis kode (*single codebase*) (Google, 2024). Flutter menggunakan bahasa pemrograman Dart dan menyediakan berbagai komponen antarmuka (*widget*) yang memungkinkan pengembangan aplikasi dengan tampilan yang konsisten serta performa yang tinggi.

Salah satu keunggulan Flutter adalah kemampuannya dalam melakukan pengembangan lintas platform (*cross-platform*), sehingga aplikasi dapat dijalankan pada berbagai sistem operasi dengan kode yang sama. Selain itu, Flutter memiliki fitur *Hot Reload* yang memungkinkan pengembang melihat perubahan kode secara langsung tanpa perlu melakukan kompilasi ulang secara keseluruhan.

Dalam penelitian ini, Flutter digunakan sebagai framework utama untuk mengembangkan aplikasi teknologi bantu berbasis Android. Flutter dipilih karena mampu mendukung integrasi berbagai teknologi yang digunakan dalam penelitian, seperti model YOLO untuk deteksi objek, OCR untuk pengenalan teks, dan *Text-to-Speech* (TTS) untuk penyampaian informasi dalam bentuk suara (Google, 2024).

2.2.7 Aksesibilitas Aplikasi Mobile

Aksesibilitas aplikasi mobile merupakan konsep perancangan dan pengembangan aplikasi yang memastikan seluruh pengguna, termasuk penyandang disabilitas, dapat mengakses dan menggunakan aplikasi secara efektif, efisien, serta mandiri. Pada platform Android, aksesibilitas menjadi aspek penting dalam menciptakan pengalaman pengguna yang inklusif, terutama bagi penyandang gangguan penglihatan yang mengandalkan umpan balik suara maupun sentuhan untuk berinteraksi dengan aplikasi (Google, 2024).

Dalam pengembangan aplikasi yang aksesibel, terdapat beberapa prinsip utama yang perlu diperhatikan. Pertama, kontras warna (*color contrast*) yang memadai agar informasi tetap dapat dibaca oleh pengguna dengan gangguan penglihatan ringan (*low vision*). Kedua, ukuran target sentuh (*touch target size*) yang cukup besar sehingga tombol mudah ditekan dan meminimalkan kesalahan interaksi. Ketiga, setiap komponen antarmuka harus memiliki *semantic labels* atau deskripsi yang dapat dikenali oleh layanan aksesibilitas seperti screen reader. Selain itu, urutan perpindahan fokus (*focus order*) antar komponen juga harus disusun secara logis sehingga pengguna dapat melakukan navigasi dengan

mudah menggunakan gestur aksesibilitas maupun keyboard (Google, 2024).

Google melalui Android Accessibility Guidelines merekomendasikan agar aplikasi menyediakan informasi yang dapat dipahami baik secara visual maupun non-visual. Pedoman tersebut mencakup penggunaan label yang jelas, navigasi yang konsisten, ukuran elemen interaktif minimal 48×48 dp, serta dukungan terhadap fitur aksesibilitas bawaan Android seperti TalkBack, pembesaran tampilan (*magnification*), dan penyesuaian ukuran teks (*font scaling*) (Google, 2024). Selain itu, pengembangan aplikasi juga dapat mengacu pada Web Content Accessibility Guidelines (WCAG) 2.2 yang menekankan empat prinsip utama aksesibilitas, yaitu Perceivable, Operable, Understandable, dan Robust (POUR). Meskipun WCAG awalnya dikembangkan untuk aplikasi berbasis web, prinsip-prinsip tersebut juga banyak diterapkan sebagai acuan dalam pengembangan aplikasi mobile karena memiliki tujuan yang sama, yaitu memastikan seluruh pengguna dapat mengakses informasi dan fungsi aplikasi secara setara (W3C, 2023).

Tabel 5 Prinsip Aksesibilitas Aplikasi Mobile

Prinsip	Deskripsi	Implementasi pada Penelitian
Kontras warna	Perbedaan warna yang cukup antara teks dan latar belakang	Menggunakan kombinasi warna dengan kontras tinggi agar mudah dibaca pengguna low vision
Ukuran target sentuh	Tombol memiliki ukuran yang cukup besar untuk mengurangi kesalahan sentuh	Seluruh tombol utama memiliki ukuran minimal sesuai rekomendasi Android
Semantic labels	Setiap komponen UI memiliki deskripsi yang dapat dibaca screen reader	Widget Flutter diberikan label semantik sehingga dikenali oleh TalkBack
Focus order	Urutan perpindahan fokus yang logis	Navigasi antar komponen mengikuti urutan tampilan aplikasi
Audio feedback	Informasi disampaikan melalui suara	Menggunakan Piper TTS sebagai umpan balik audio
Haptic feedback	Memberikan getaran sebagai konfirmasi tindakan	Digunakan sebagai umpan balik saat tombol ditekan atau proses tertentu selesai

Salah satu teknologi aksesibilitas utama pada sistem operasi Android adalah TalkBack, yaitu layanan screen reader yang mengubah informasi visual menjadi keluaran suara dan umpan balik getaran (*haptic feedback*). Ketika pengguna menyentuh suatu komponen antarmuka, TalkBack akan membaca label semantik dari komponen tersebut dan memberikan informasi mengenai fungsi maupun statusnya. Oleh karena itu, pengembang harus memastikan setiap komponen antarmuka memiliki deskripsi (*content description* atau *semantic label*) yang jelas serta urutan fokus navigasi

yang konsisten agar aplikasi dapat digunakan secara optimal oleh penyandang tunanetra (Google, 2024).

Kebutuhan aksesibilitas juga berbeda berdasarkan tingkat gangguan penglihatan yang dimiliki pengguna. Pengguna low vision umumnya masih memiliki sisa kemampuan penglihatan sehingga memerlukan antarmuka dengan kontras tinggi, ukuran teks yang dapat diperbesar, serta ikon yang mudah dikenali. Sebaliknya, pengguna tunanetra total bergantung sepenuhnya pada keluaran audio melalui screen reader, navigasi berbasis gestur, serta umpan balik getaran untuk memahami kondisi aplikasi. Oleh karena itu, aplikasi yang dikembangkan dalam penelitian ini dirancang agar mendukung kedua kelompok pengguna tersebut melalui integrasi TalkBack, Piper Text-to-Speech, serta haptic feedback sehingga informasi hasil deteksi objek maupun pembacaan teks dapat diterima secara efektif oleh pengguna.

Berdasarkan prinsip-prinsip aksesibilitas tersebut, aplikasi yang dikembangkan pada penelitian ini tidak hanya berfokus pada akurasi deteksi objek menggunakan YOLO maupun pengenalan teks menggunakan OCR, tetapi juga memastikan bahwa seluruh informasi dapat diakses oleh penyandang gangguan penglihatan melalui antarmuka yang sesuai dengan pedoman aksesibilitas Android. Pendekatan ini diharapkan mampu meningkatkan kemudahan penggunaan (*usability*) sekaligus mendukung kemandirian pengguna dalam berinteraksi dengan lingkungan sekitar.

2.3. Metode Pengembangan Produk

Metode pengembangan yang digunakan dalam penelitian ini adalah metode *Agile*, yaitu pendekatan pengembangan perangkat lunak yang bersifat iteratif dan fleksibel. Metode ini memungkinkan proses pengembangan dilakukan secara bertahap dan berulang sehingga sistem dapat terus disempurnakan berdasarkan hasil evaluasi pada setiap tahapan. Adapun tahapan dalam metode *Agile* yang diterapkan adalah sebagai berikut.



Gambar 5 Metode Agile

Sumber: BINAR (2024)

1. Requirements

Tahap awal yang berfokus pada pengumpulan dan analisis kebutuhan sistem. Pada tahap ini dilakukan identifikasi kebutuhan fungsional dan non-fungsional aplikasi berdasarkan permasalahan yang dihadapi pengguna, khususnya penyandang gangguan penglihatan.

2. Design

Tahap perancangan sistem berdasarkan kebutuhan yang telah dikumpulkan, meliputi perancangan arsitektur sistem, basis data, dan antarmuka pengguna (*wireframe*).

3. **Development**

Tahap implementasi atau pengkodean sistem sesuai dengan rancangan yang telah dibuat, termasuk integrasi teknologi YOLO, OCR, dan TTS ke dalam aplikasi Android.

4. **Testing**

Tahap pengujian sistem untuk memastikan seluruh fitur berjalan sesuai dengan kebutuhan yang telah ditetapkan, menggunakan metode *Black Box Testing* dan *User Acceptance Testing* (UAT).

5. **Deployment**

Tahap penerapan aplikasi pada perangkat Android untuk digunakan dan diuji secara langsung oleh pengguna.

6. **Review**

Tahap evaluasi terhadap hasil pengembangan dan pengujian. Apabila ditemukan kekurangan atau kebutuhan baru, proses dapat kembali ke tahap awal untuk dilakukan perbaikan dan penyempurnaan secara iteratif.

2.3.1 Analisis Kebutuhan Sistem

Pada tahap ini dilakukan identifikasi kebutuhan sistem berdasarkan permasalahan yang dihadapi oleh penyandang gangguan penglihatan. Analisis kebutuhan dilakukan melalui studi literatur terhadap penelitian terdahulu, jurnal ilmiah, dan berbagai sumber yang berkaitan dengan *assistive technology*.

Berdasarkan hasil analisis, diperoleh beberapa kebutuhan utama yang harus dipenuhi oleh aplikasi, yaitu kemampuan mendeteksi objek di sekitar pengguna secara *real-time*, membaca teks dari lingkungan sekitar menggunakan teknologi OCR, serta menyampaikan informasi dalam bentuk suara melalui teknologi *Text-to-Speech* (TTS). Selain kebutuhan fungsional, ditentukan pula kebutuhan non-fungsional seperti kompatibilitas pada perangkat Android, kemudahan penggunaan, dan dukungan fitur aksesibilitas.

Hasil dari tahap ini menjadi dasar dalam perancangan dan pengembangan aplikasi yang sesuai dengan kebutuhan pengguna.

2.3.2 Perancangan Sistem

Tahap perancangan sistem dilakukan untuk menghasilkan gambaran mengenai struktur dan alur kerja aplikasi sebelum proses implementasi dimulai. Perancangan meliputi penyusunan use case diagram, activity diagram, Entity Relationship Diagram (ERD), wireframe, serta rancangan antarmuka pengguna.

Pada tahap ini juga dilakukan perencanaan integrasi teknologi yang digunakan dalam aplikasi. Model YOLOv5n digunakan untuk mendeteksi objek melalui kamera perangkat, OCR digunakan untuk mengenali teks dari gambar, sedangkan Piper TTS digunakan untuk mengubah hasil deteksi objek maupun hasil OCR menjadi informasi suara yang dapat didengar oleh pengguna.

Hasil perancangan ini menjadi acuan dalam proses pengembangan aplikasi sehingga implementasi dapat dilakukan secara terstruktur dan sesuai dengan kebutuhan yang telah dianalisis sebelumnya.

2.3.3 Implementasi dan Integrasi Sistem

Tahap implementasi merupakan proses pengembangan aplikasi berdasarkan rancangan yang telah dibuat sebelumnya. Aplikasi dikembangkan menggunakan framework Flutter dengan bahasa pemrograman Dart untuk membangun antarmuka dan logika sistem.

Pada tahap ini dilakukan integrasi berbagai komponen utama aplikasi, yaitu model YOLOv5n untuk deteksi objek, OCR untuk pengenalan teks, dan Piper TTS untuk konversi teks menjadi suara. Selain itu, fitur riwayat deteksi, penyimpanan data, serta dukungan aksesibilitas juga diimplementasikan untuk meningkatkan kemudahan penggunaan aplikasi bagi penyandang gangguan penglihatan.

Setelah seluruh fitur berhasil diintegrasikan, dilakukan pengujian menggunakan metode Black Box Testing dan User Acceptance Testing (UAT) untuk memastikan bahwa setiap fungsi berjalan sesuai dengan kebutuhan yang telah ditentukan. Hasil pengujian kemudian digunakan sebagai bahan evaluasi dan perbaikan pada iterasi berikutnya sesuai dengan prinsip metode Agile.

BAB III ANALISIS DAN PERANCANGAN

3.1. Analisis Kebutuhan.

Bagian ini menjelaskan kebutuhan sistem secara menyeluruh, dimulai dari pemahaman terhadap kondisi saat ini hingga rancangan awal sistem yang akan diusulkan.

3.1.1 Persona Pengguna

Sebelum merancang sistem, perlu dilakukan identifikasi terhadap karakteristik pengguna yang akan menggunakan aplikasi. Persona pengguna digunakan untuk menggambarkan kebutuhan, karakteristik, serta kendala yang dihadapi oleh calon pengguna sehingga proses perancangan sistem dapat lebih berorientasi pada kebutuhan nyata. Berdasarkan tujuan penelitian, aplikasi ini dirancang untuk dua kelompok pengguna utama, yaitu penyandang low vision dan tunanetra total.

Tabel 6 Pengguna Low Vision

Aspek	Deskripsi
Nama Persona	Pengguna Low Vision
Karakteristik	Memiliki sisa kemampuan penglihatan tetapi mengalami kesulitan melihat objek secara jelas, terutama pada jarak tertentu atau kondisi pencahayaan yang kurang baik.
Kebutuhan	Kontras warna tinggi, ukuran tombol besar, ukuran teks yang dapat diperbesar, serta informasi dalam bentuk visual dan audio secara bersamaan.
Kendala	Sulit mengenali objek kecil, membaca teks berukuran kecil, dan membedakan warna dengan kontras rendah.
Solusi pada aplikasi	Antarmuka menggunakan tema kontras tinggi, tombol berukuran besar, deteksi objek real-time, OCR, serta keluaran suara melalui Text-to-Speech.

Tabel 7 Pengguna Tunanetra Total

Aspek	Deskripsi
Nama Persona	Pengguna Tunanetra Total
Karakteristik	Tidak memiliki kemampuan melihat sehingga seluruh interaksi bergantung pada indera pendengaran dan sentuhan.
Kebutuhan	Navigasi non-visual, TalkBack, semantic labels, haptic feedback, serta informasi yang seluruhnya disampaikan melalui audio.
Kendala	Tidak dapat melihat posisi tombol, membaca informasi visual, maupun mengetahui hasil deteksi tanpa bantuan audio.
Solusi pada aplikasi	Integrasi penuh dengan TalkBack, Piper TTS sebagai media penyampaian informasi, serta umpan balik getaran (haptic feedback) pada setiap interaksi penting.

Berdasarkan kedua persona tersebut, dapat diketahui bahwa kebutuhan pengguna berbeda sesuai dengan tingkat gangguan penglihatan yang dimiliki. Pengguna low vision masih memanfaatkan informasi visual sehingga antarmuka aplikasi perlu menggunakan kontras warna yang tinggi, ukuran teks yang cukup besar, dan ikon yang mudah dikenali. Sebaliknya, pengguna tunanetra total bergantung sepenuhnya pada keluaran audio dan umpan balik sentuhan sehingga aplikasi harus menyediakan navigasi yang kompatibel dengan TalkBack, semantic labels pada setiap komponen antarmuka, serta haptic feedback sebagai konfirmasi setiap interaksi.

Perbedaan karakteristik tersebut menjadi dasar dalam perancangan fitur aplikasi sehingga sistem mampu memberikan pengalaman penggunaan yang inklusif bagi kedua kelompok pengguna.

Berdasarkan hasil identifikasi persona pengguna tersebut, kebutuhan fungsional sistem disusun untuk mengakomodasi karakteristik masing-masing pengguna. Fitur deteksi objek menggunakan YOLO dan pengenalan teks menggunakan OCR dirancang untuk membantu pengguna memperoleh informasi visual dari lingkungan sekitar. Informasi yang dihasilkan kemudian dikonversi menjadi suara melalui fitur Text-to-Speech sehingga dapat

diakses oleh pengguna dengan gangguan penglihatan. Selain itu, integrasi TalkBack dan umpan balik berupa audio serta haptic feedback diterapkan untuk mendukung navigasi non-visual, khususnya bagi pengguna tunanetra total. Dengan demikian, kebutuhan fungsional yang dirumuskan pada Tabel 2 merupakan hasil analisis terhadap kebutuhan kedua persona pengguna sehingga sistem yang dikembangkan sesuai dengan tujuan penelitian.

3.1.2 Proses Bisnis Berjalan

3.1.2.1 Deskripsi Naratif Proses Berjalan Saat Ini

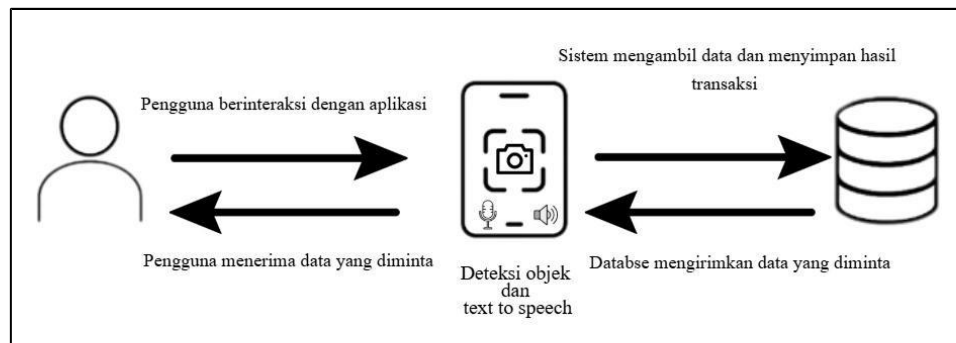
Aktivitas sehari-hari penyandang gangguan penglihatan saat ini masih sangat bergantung pada teknologi bantu sederhana dan bantuan manusia. Dalam hal navigasi dan mengenali lingkungan, pengguna umumnya mengandalkan tongkat putih (*white cane*) untuk mendeteksi rintangan fisik secara langsung. Selain itu, mereka seringkali memerlukan panduan dari orang lain untuk berpindah tempat, mengenali lingkungan yang dinamis, atau mendeteksi rintangan secara *real-time* yang tidak terjangkau oleh tongkat. Sementara itu, akses terhadap informasi visual dilakukan dengan mengandalkan perangkat braille atau dengan meminta bantuan orang lain untuk membacakan teks visual. Keterbatasan penggunaan alat bantu yang ada, terutama dalam mendeteksi rintangan secara *real-time* dan mengenali lingkungan dinamis, seringkali membatasi kemandirian dan partisipasi penyandang gangguan penglihatan dalam berbagai aspek kehidupan, termasuk pendidikan, pekerjaan, serta akses terhadap informasi dan layanan publik.

3.1.2.2 Permasalahan atau Kendala pada Proses Berjalan

kendala utama yang dihadapi oleh penyandang gangguan penglihatan saat ini adalah keterbatasan deteksi *real-time*, di mana alat bantu tradisional seperti tongkat putih memiliki batasan dalam mendeteksi rintangan secara seketika atau mengenali lingkungan yang dinamis. Selanjutnya, masalah muncul pada akses terhadap informasi visual yang dinamis, sebab pengguna sulit

membaca informasi visual yang semakin banyak digunakan di era digital. Akibatnya, mereka sering kali harus meminta bantuan orang lain, yang pada akhirnya mengurangi kemandirian. Keterbatasan-keterbatasan ini secara keseluruhan memberikan dampak signifikan terhadap kemandirian dan partisipasi penyandang disabilitas penglihatan dalam berbagai aspek kehidupan, seperti pendidikan, pekerjaan, dan akses layanan publik. Oleh karena itu, muncul kebutuhan akan solusi inklusif yang inovatif, mengingat perkembangan teknologi informasi saat ini belum sepenuhnya diiringi oleh solusi *assistive technology* yang canggih, efektif, dan mudah diakses

3.1.3 Gambaran Umum Sistem



Gambar 6 Gambaran Umum Sistem

Gambar 6 diatas menggambarkan sebuah arsitektur sistem yang berpusat pada perangkat mobile, menunjukkan interaksi antara tiga komponen utama: Pengguna, Perangkat (Smartphone), dan Penyimpanan Data. Proses dimulai ketika Pengguna berinteraksi dengan Perangkat. Perangkat ini bertindak sebagai pusat pemrosesan utama, yang didukung oleh Modul Lokal. Kehadiran Modul Lokal menunjukkan bahwa perangkat mampu melakukan pemrosesan data tertentu secara independen atau on-device, seringkali untuk kecepatan dan efisiensi. Selain itu, Perangkat juga memiliki komunikasi dua arah yang krusial dengan Penyimpanan Data (seperti database). Interaksi ini memungkinkan Perangkat untuk mengambil data yang dibutuhkan atau menyimpan hasil transaksi dan update data secara berkala. Akhirnya, hasil dari seluruh pemrosesan baik yang dilakukan secara lokal maupun yang

melibatkan akses data dikirimkan kembali sebagai respons kepada Pengguna, melengkapi siklus interaksi sistem tersebut.

3.2. Perancangan

3.2.1 Kebutuhan Fungsional

Tabel 8 Kebutuhan Fungsional

Kode	Keterangan
FR- 01	Sistem dapat mengenali objek di sekitar pengguna menggunakan kamera dan model deteksi objek (YOLO)
FR- 02	Sistem dapat membaca teks dari gambar melalui fitur OCR (Optical Character Recognition).
FR- 03	Sistem dapat mengubah teks hasil OCR menjadi suara dengan fitur <i>Text-to-Speech</i> (TTS).
FR- 04	Sistem terintegrasi dengan <i>TalkBack</i> agar seluruh elemen aplikasi dapat diakses oleh pengguna tunanetra.
FR- 05	Sistem memberikan umpan balik audio dan getaran (<i>haptic feedback</i>) saat pengguna melakukan interaksi penting.
FR- 06	Sistem dapat menyimpan riwayat hasil deteksi objek, OCR, dan TTS ke dalam basis data lokal.

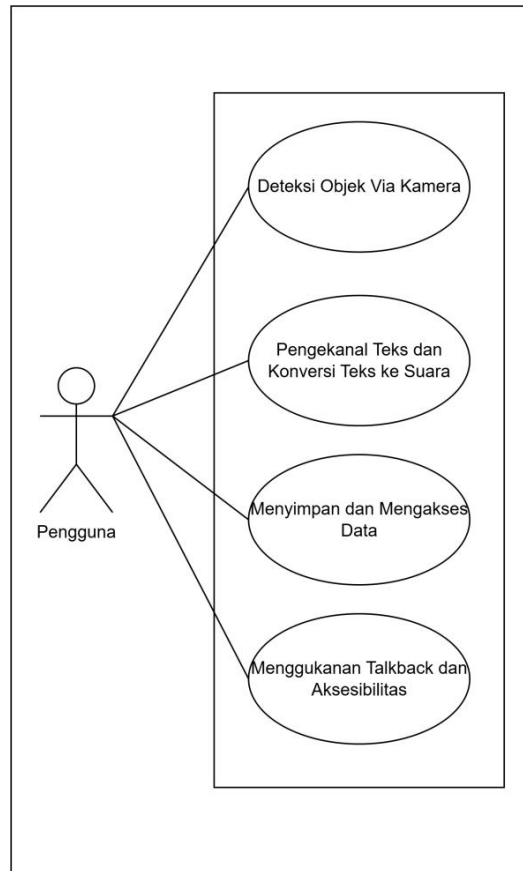
3.2.2 Kebutuhan Non-Fungsional

Tabel 9 Non-Fungsional

<i>Safety</i>	Sistem memberikan notifikasi suara apabila kamera gagal diakses atau tidak tersedia.
<i>Safety</i>	Sistem memberikan peringatan bahwa aplikasi hanya berfungsi sebagai alat bantu dan tidak menggantikan tongkat atau pendamping.
<i>Privacy</i>	Gambar hasil deteksi dan OCR tidak disimpan tanpa persetujuan pengguna.

<i>Privacy</i>	Data riwayat disimpan secara lokal pada perangkat pengguna.
<i>Reliability</i>	Sistem dapat menjalankan fungsi deteksi objek, OCR, dan TTS tanpa mengalami kegagalan yang signifikan.
<i>Usability</i>	Antarmuka dirancang sederhana, dengan kontras tinggi dan ukuran tombol besar.
<i>Accessibility</i>	Sistem mendukung TalkBack dan umpan balik suara untuk membantu pengguna tunanetra.
<i>Performance</i>	Waktu respon deteksi objek, OCR, dan TTS kurang dari 2 detik pada kondisi normal.
<i>Maintainability</i>	Struktur kode disusun secara terorganisir untuk memudahkan pemeliharaan dan pengembangan lebih lanjut.
<i>Documentation</i>	Sistem memiliki dokumentasi yang lengkap, rapi, dan teratur

3.2.3 Diagram Use Case



Gambar 7 Usecase Diagram

Diagram use case ini memodelkan sistem aplikasi teknologi bantu berbasis Android yang berfokus pada fitur pengenalan visual dan aksesibilitas bagi penyandang tunanetra dan *low vision*. Sistem melibatkan satu aktor utama, yaitu Pengguna. Pengguna dapat melakukan registrasi dan login ke dalam aplikasi untuk mengakses seluruh fitur yang tersedia. Fitur utama yang disediakan meliputi Deteksi Objek via Kamera untuk mengenali objek atau rintangan di lingkungan sekitar menggunakan teknologi YOLO, Pengenalan Teks (*Optical Character Recognition / OCR*) untuk membaca teks dari gambar, serta Konversi Teks ke Suara (*Text-to-Speech / TTS*) untuk menyampaikan hasil deteksi dan pembacaan teks dalam bentuk audio. Selain itu, pengguna dapat menyimpan dan mengakses riwayat hasil deteksi maupun OCR, serta memanfaatkan fitur TalkBack dan aksesibilitas lainnya untuk mendukung penggunaan aplikasi secara mandiri. Dengan fitur-fitur tersebut, aplikasi diharapkan dapat

membantu pengguna memperoleh informasi mengenai lingkungan sekitar dan informasi berbasis teks secara lebih mudah dan aksesibel.

3.2.4 Skenario Use Case

Tabel 10 Scenario Usecase (Deteksi Objek via Kamera)

Identifikasi	
Nomor	1
Nama	Deteksi Objek via Kamera
Deskripsi	Sistem menggunakan kamera dan model AI untuk mengenali objek.
Aktor	Pengguna
Langkah – Langkah:	
1. Pengguna memilih menu Deteksi Objek. 2. Sistem mengaktifkan kamera perangkat. 3. Pengguna mengarahkan kamera ke objek. 4. Sistem memproses citra menggunakan model YOLO. 5. Sistem mengenali objek yang terdeteksi. 6. Sistem membacakan hasil deteksi menggunakan TTS. 7. Sistem menyimpan hasil deteksi ke riwayat.	
Kondisi Gagal:	
1. A1. Jika kamera tidak dapat diakses - sistem memberikan notifikasi suara kepada pengguna. 2. A2. Jika objek tidak berhasil dikenali - sistem memberikan informasi bahwa objek tidak dikenali.	

Tabel 11 Scenario Usecase (Pengenal Teks (OCR) & Konversi Teks ke Suara (TTS))

Identifikasi	
Nomor	2
Nama	Pengenal Teks (OCR) & Konversi Teks ke Suara (TTS)
Deskripsi	Sistem mengenali teks dari gambar kemudian membacakannya.
Aktor	Pengguna
Langkah – Langkah:	
1. Pengguna memilih menu Baca Teks. 2. Pengguna mengambil gambar atau memilih gambar dari galeri. 3. Sistem memproses gambar menggunakan OCR. 4. Sistem menampilkan hasil teks. 5. Sistem mengubah teks menjadi suara menggunakan TTS. 6. Sistem membacakan hasil kepada pengguna. 7. Sistem menyimpan hasil ke riwayat.	
Kondisi Gagal:	
1. A1. Tidak ditemukan teks pada gambar. - Sistem memberikan notifikasi bahwa teks tidak ditemukan.	

Tabel 12 Scenario Usecase (Menyimpan & Mengakses Data)

Identifikasi	
Nomor	3
Nama	Menyimpan & Mengakses Data
Deskripsi	Sistem menyimpan data hasil aktivitas ke server atau lokal.
Aktor	Pengguna
Langkah – Langkah:	
1. Sistem menyimpan hasil deteksi objek atau OCR. 2. Data disimpan pada basis data lokal. 3. Pengguna membuka menu Riwayat. 4. Sistem menampilkan daftar riwayat. 5. Pengguna memilih salah satu data. 6. Sistem menampilkan detail riwayat.	

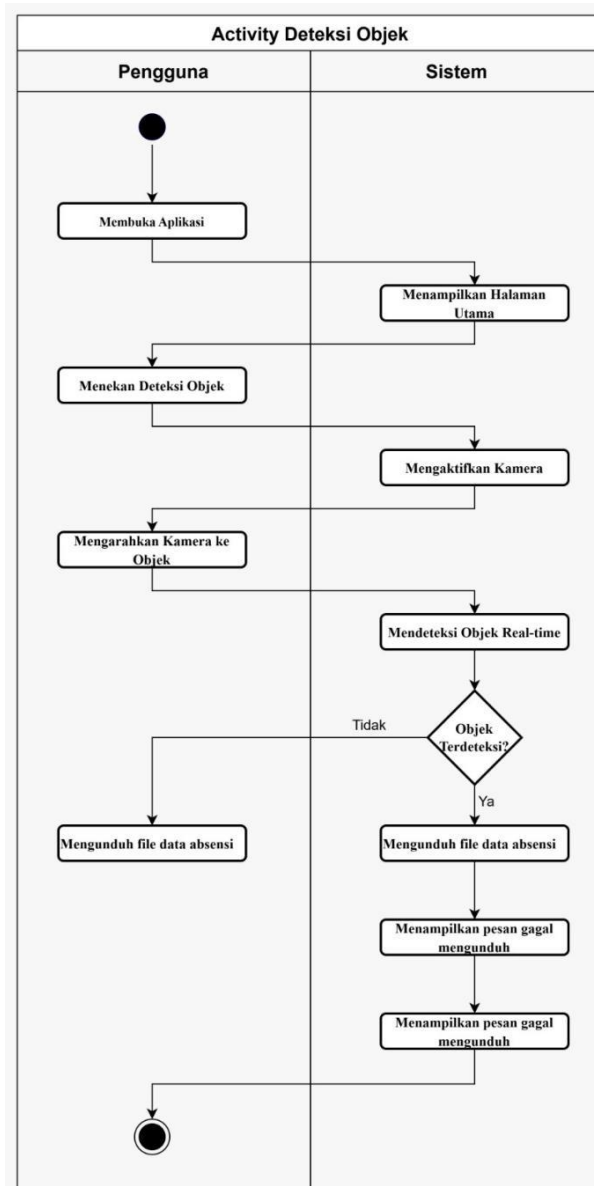
Tabel 13 Scenario User (Menggunakan TalkBack & Aksesibilitas)

Identifikasi	
Nomor	4
Nama	Menggunakan TalkBack & Aksesibilitas
Deskripsi	Sistem terintegrasi dengan TalkBack dan umpan balik suara.
Aktor	Pengguna
Langkah – Langkah:	
1. Pengguna mengaktifkan TalkBack. 2. Pengguna membuka aplikasi. 3. Pengguna melakukan navigasi menu. 4. Sistem membacakan nama dan fungsi setiap elemen. 5. Pengguna memilih fitur yang diinginkan. 6. Sistem memberikan umpan balik suara dan getaran.	
Kondisi Gagal:	
1. A1. <i>TalkBack</i> tidak aktif. - Sistem tetap dapat digunakan dengan tombol dan TTS bawaan.	

3.2.5 Activity Diagram

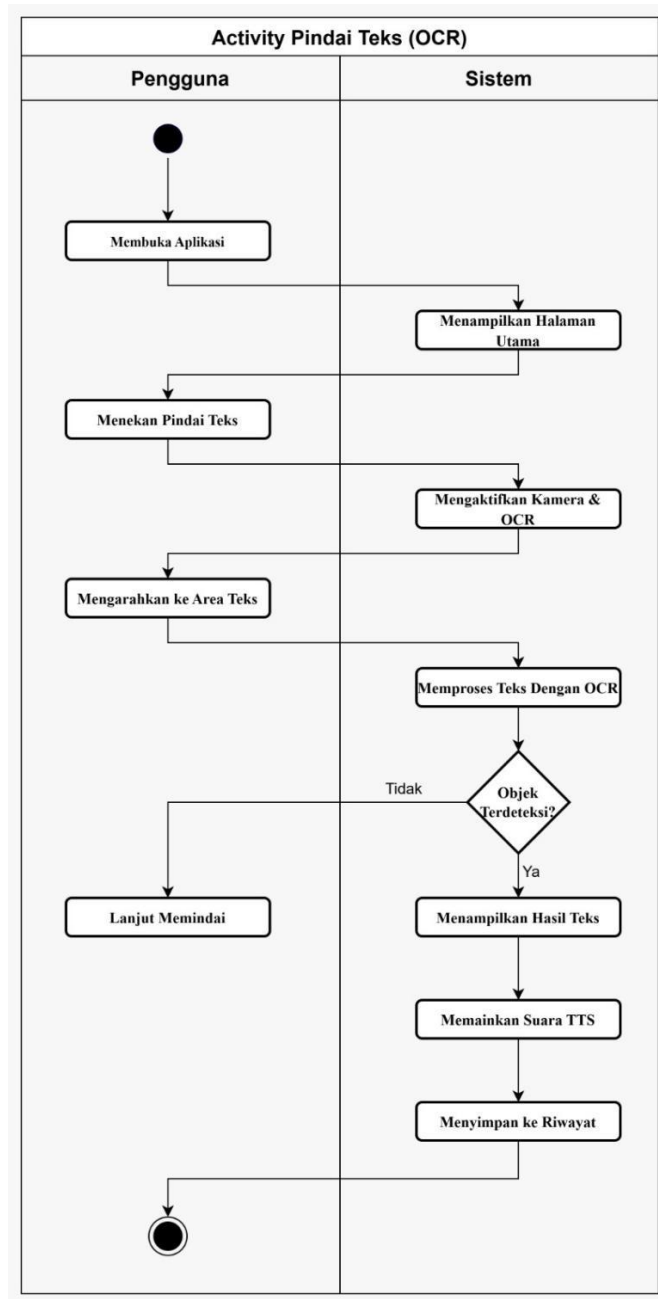
Activity diagram menggambarkan alur aktivitas yang terjadi antara pengguna dan sistem dalam setiap fitur aplikasi. Terdapat lima activity diagram yang masing-masing merepresentasikan alur penggunaan fitur utama aplikasi.

3.2.5.1 Activity Deteksi Objek



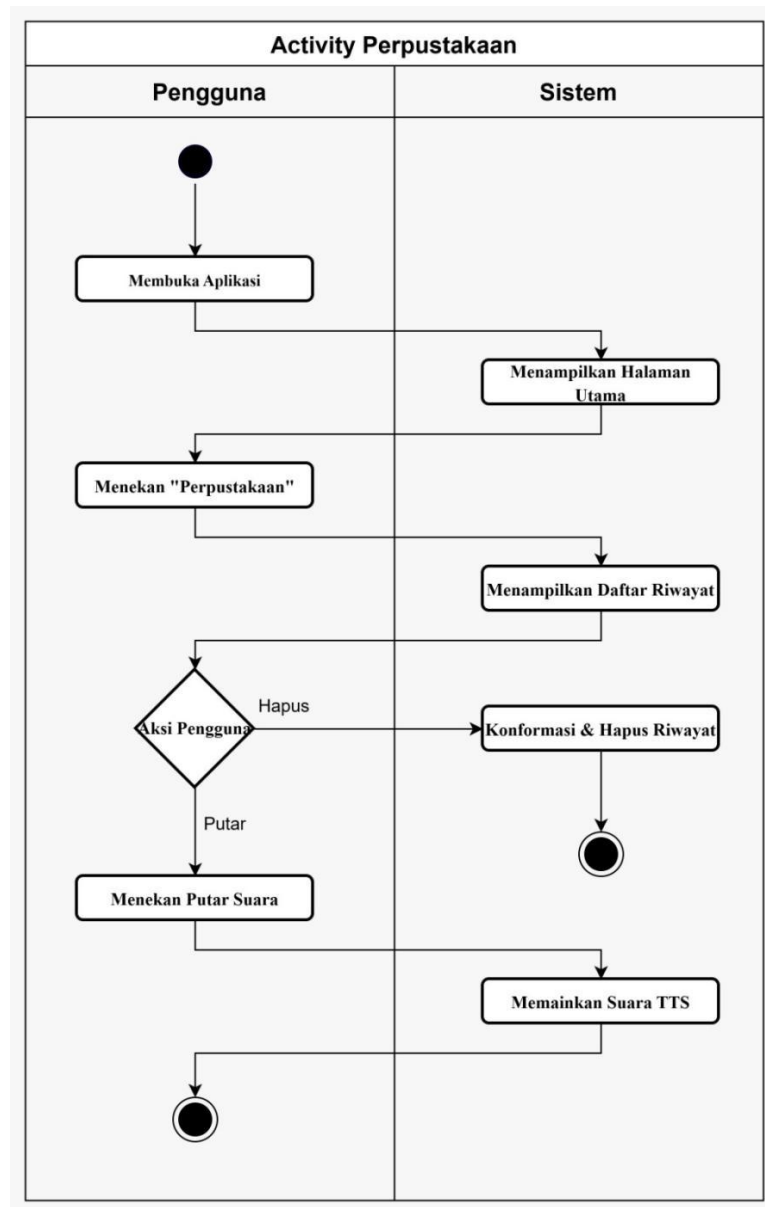
Gambar 8 Activity Deteksi Objek

3.2.5.2 Activity Pindai Teks (OCR)



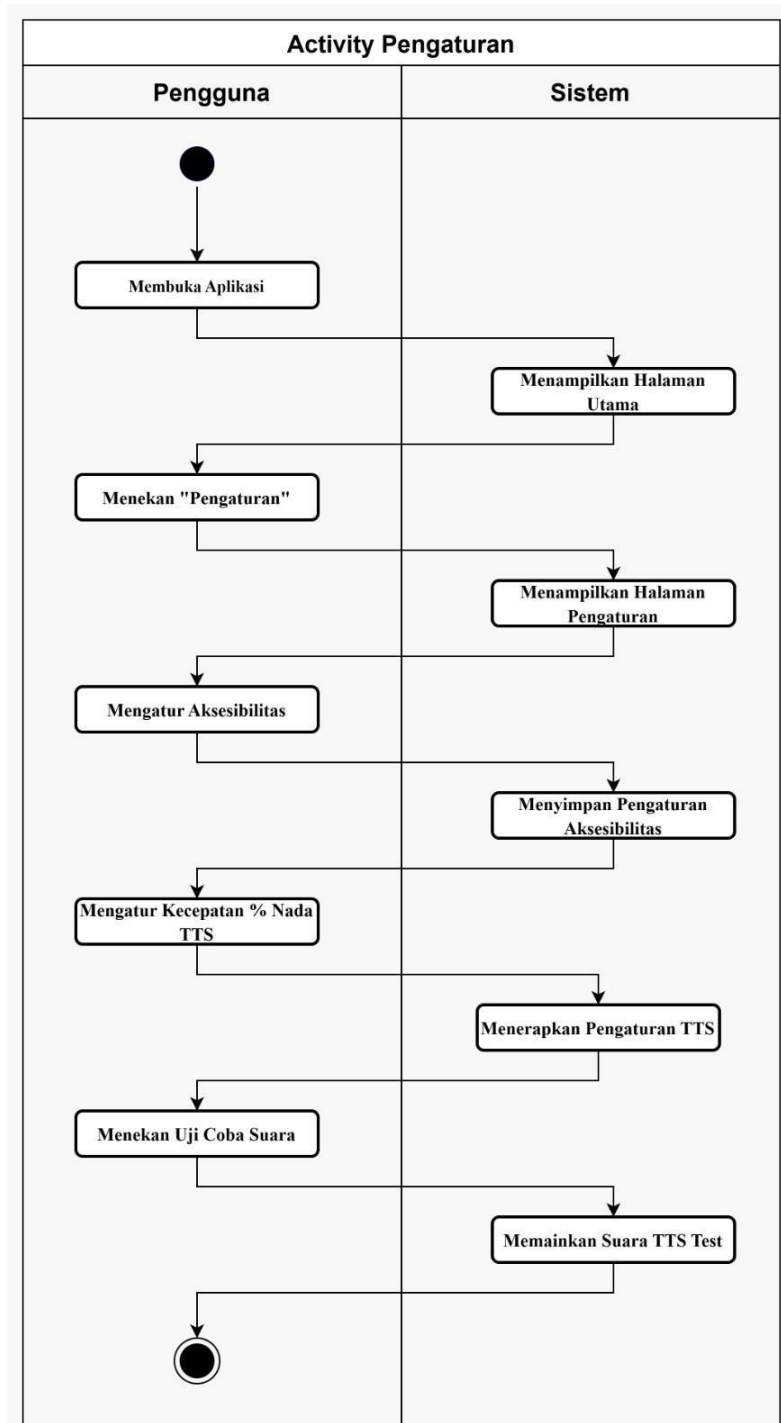
Gambar 9 Activity Pindai Teks (OCR)

3.2.5.3 Activity Perpustakaan



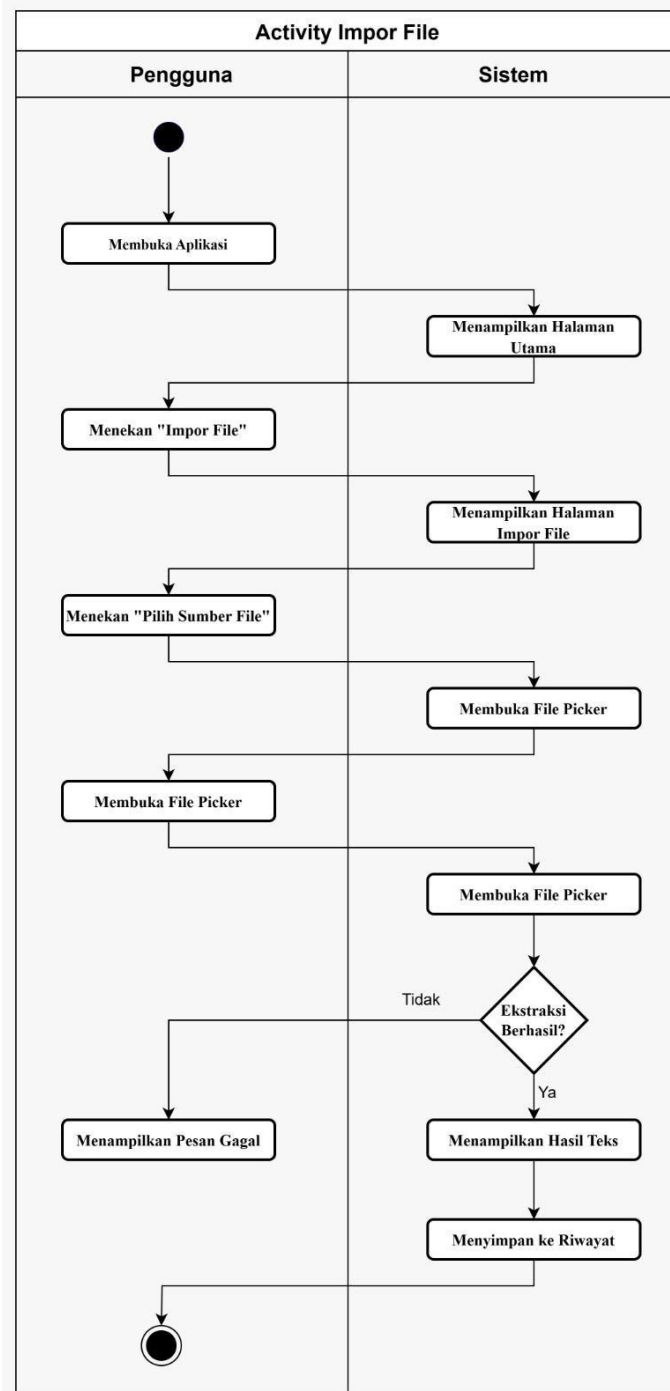
Gambar 10 Activity Perpustakaan

3.2.5.4 Activity Pengaturan



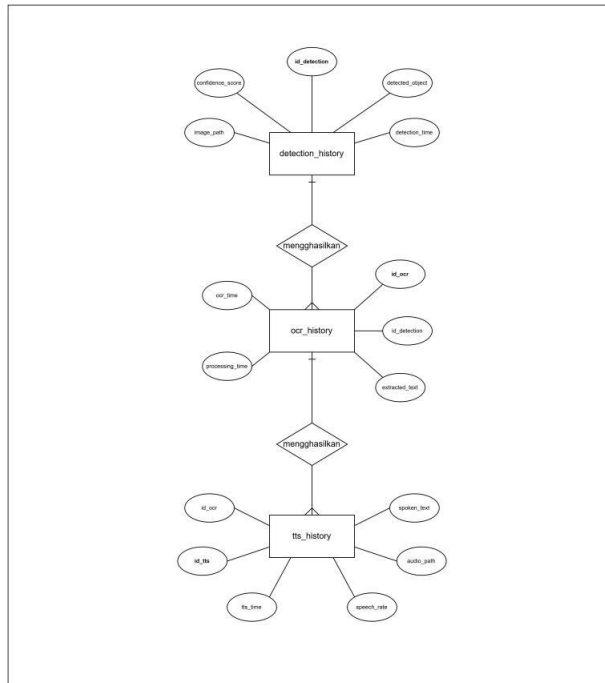
Gambar 11 Activity Pengaturan

3.2.5.5 Activity Impor File



Gambar 12 Activity Impor File

3.2.6 ER Diagram



Gambar 13 ER-Diagram sistem

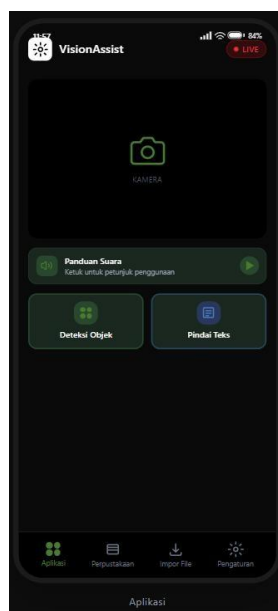
ERD pada Gambar 13 menggambarkan struktur basis data yang terdiri dari tiga entitas utama, yaitu **detection_history**, **ocr_history**, dan **tts_history**. Entitas **detection_history** menyimpan riwayat hasil deteksi objek, seperti nama objek, tingkat keyakinan deteksi, lokasi gambar, dan waktu deteksi. Data hasil deteksi selanjutnya diproses pada entitas **ocr_history** yang menyimpan hasil pengenalan teks (OCR), termasuk teks yang berhasil diekstraksi dan waktu pemrosesan. Hasil OCR kemudian digunakan oleh entitas **tts_history** untuk menyimpan riwayat konversi teks ke suara, meliputi teks yang dikonversi, file audio yang dihasilkan, dan waktu proses. Ketiga entitas tersebut saling terhubung sehingga membentuk alur data dari deteksi objek, pengenalan teks, hingga keluaran suara

3.2.7 Perancangan Antarmuka (Wireframe)

Perancangan antarmuka aplikasi VisionAssist menggunakan tema gelap (*dark mode*) dengan kontras warna tinggi dan ukuran elemen yang besar guna memudahkan pengguna dengan gangguan penglihatan. Navigasi utama menggunakan *bottom navigation bar* dengan empat menu, yaitu Aplikasi, Perpustakaan, Impor File, dan Pengaturan, sehingga pengguna dapat berpindah halaman dengan cepat dan mudah. Setiap halaman dirancang dengan fungsi yang spesifik dan terfokus, mulai dari deteksi objek secara *real-time*, pengelolaan riwayat, pemrosesan file eksternal, hingga kustomisasi pengaturan aplikasi. Antarmuka juga mendukung aksesibilitas Android seperti TalkBack, *Text-to-Speech*, dan *haptic feedback* agar pengguna tetap dapat memperoleh informasi meskipun tidak melihat layar secara langsung.

3.2.7.1 Halaman Utama (Beranda)

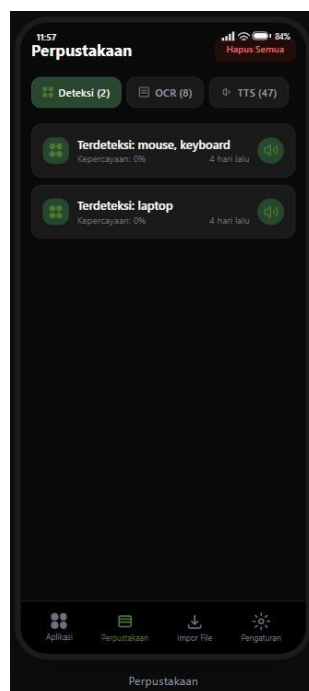
Halaman Utama (Beranda) Halaman utama menampilkan area kamera *real-time* untuk deteksi objek, tombol Panduan Suara sebagai petunjuk penggunaan berbasis audio, serta dua tombol utama yaitu Deteksi Objek dan Pindai Teks. Navigasi antar halaman tersedia melalui *bottom navigation bar* di bagian bawah layar.



Gambar 14 Halaman Utama (Beranda)

3.2.7.2 Halaman Perpustakaan

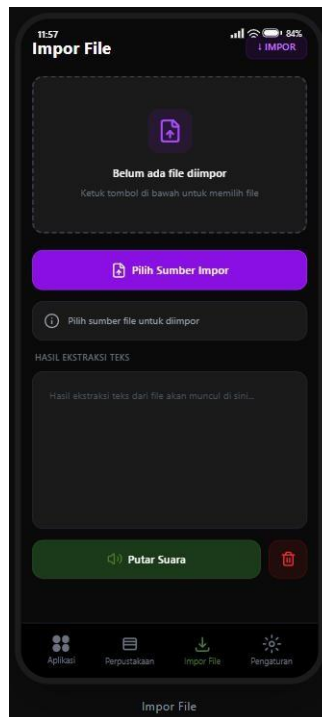
Halaman Perpustakaan Halaman ini menampilkan riwayat aktivitas pengguna yang dapat disaring berdasarkan tiga kategori melalui tab filter Deteksi, OCR, dan TTS. Setiap riwayat ditampilkan dalam bentuk kartu berisi hasil aktivitas, tingkat kepercayaan, dan waktu pencatatan. Tersedia pula tombol Hapus Semua untuk menghapus seluruh riwayat sekaligus.



Gambar 15 Halaman Perpustakaan.

3.2.7.3 Halaman Impor File

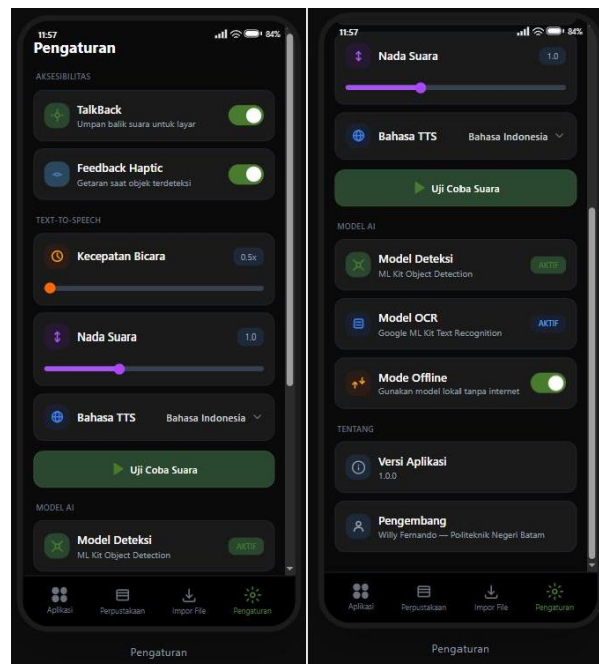
Halaman Impor File Halaman ini memungkinkan pengguna mengimpor file dari perangkat untuk diproses menggunakan fitur OCR. Hasil ekstraksi teks ditampilkan langsung pada area teks yang tersedia, kemudian dapat diputar sebagai audio melalui tombol Putar Suara atau dihapus menggunakan tombol hapus.



Gambar 16 Halaman Impor File

3.2.7.4 Halaman Pengaturan

Halaman Pengaturan Halaman pengaturan terbagi menjadi tiga kelompok, yaitu Aksesibilitas yang mencakup pengaturan TalkBack dan *Feedback Haptic*, *Text-to-Speech* yang mencakup kecepatan bicara, nada suara, bahasa TTS, dan uji coba suara, serta Model AI yang menampilkan informasi model deteksi objek yang sedang aktif.



Gambar 17 Halaman Pengaturan

BAB IV IMPLEMENTASI DAN PENGUJIAN

4.1. Implementasi Metode Agile

Penelitian ini menerapkan metode Agile sebagaimana dijelaskan pada Bab II. Proses pengembangan dilakukan secara bertahap melalui beberapa iterasi (sprint), sehingga setiap fitur yang dikembangkan dapat dievaluasi dan disempurnakan berdasarkan hasil pengujian. Pendekatan ini memungkinkan pengembangan aplikasi berlangsung secara fleksibel tanpa harus menunggu seluruh sistem selesai dibangun terlebih dahulu.

Pada penelitian ini, setiap sprint menghasilkan peningkatan terhadap fungsi aplikasi, mulai dari implementasi deteksi objek, pengenalan teks, sintesis suara, hingga penyempurnaan aspek aksesibilitas. Ringkasan pelaksanaan sprint ditunjukkan pada Tabel 14.

Tabel 14 Implementasi Sprint Agile

Sprint	Tahap Agile	Aktivitas	Hasil
Sprint 1	Requirements & Design	Analisis kebutuhan pengguna dan perancangan antarmuka	Use Case, Activity Diagram, Wireframe
Sprint 2	Development	Implementasi YOLOv8n untuk deteksi objek	Deteksi objek berhasil berjalan secara real-time
Sprint 3	Development	Integrasi OCR Tesseract dan Piper TTS	Pembacaan teks dan keluaran suara berhasil diimplementasikan
Sprint 4	Testing & Review	Pengujian fitur, integrasi TalkBack dan haptic feedback	Perbaikan antarmuka serta peningkatan stabilitas aplikasi
Sprint 5	Deployment	Instalasi APK pada perangkat Android dan pengujian langsung	Aplikasi siap digunakan pada perangkat Android

4.1.1 Hasil Evaluasi Setiap Iterasi

Selama proses pengembangan dilakukan beberapa perbaikan berdasarkan hasil evaluasi pada setiap sprint. Salah satu perubahan dilakukan pada modul deteksi objek menggunakan Google ML Kit Object Detection.

Pada iterasi awal digunakan confidence threshold sebesar 0,5. Berdasarkan hasil pengujian, model masih menghasilkan beberapa false positive, yaitu objek yang sebenarnya bukan target tetapi tetap terdeteksi. Oleh karena itu, pada iterasi berikutnya nilai confidence threshold dinaikkan menjadi 0,6.

Perubahan ini berhasil mengurangi jumlah false positive tanpa menurunkan kemampuan model dalam mengenali objek utama secara signifikan.

Selain itu, pada modul OCR dilakukan penambahan tahap preprocessing berupa konversi citra ke grayscale dan peningkatan kontras sebelum proses pengenalan karakter. Perubahan ini meningkatkan keterbacaan teks terutama pada kondisi pencahayaan yang kurang optimal.

Pada modul Piper TTS juga dilakukan optimasi proses inferensi sehingga waktu respons suara menjadi lebih cepat. Selanjutnya dilakukan integrasi fitur TalkBack dan haptic feedback untuk meningkatkan aksesibilitas bagi pengguna tunanetra maupun low vision.

4.2. Hasil Implementasi

Pada tahap ini dilakukan implementasi sistem berdasarkan hasil analisis kebutuhan dan perancangan yang telah dijelaskan pada Bab III. Implementasi merupakan proses penerapan rancangan sistem menjadi sebuah aplikasi nyata yang dapat dijalankan. Sistem yang dikembangkan berupa aplikasi mobile berbasis Android yang dirancang sebagai teknologi bantu bagi penyandang gangguan penglihatan, baik *low vision* maupun tunanetra.

Aplikasi ini dibangun dengan mengintegrasikan beberapa teknologi kecerdasan buatan, yaitu deteksi objek, pengenalan teks (*Optical Character Recognition/OCR*), dan konversi teks ke suara (*Text-to-Speech/TTS*). Ketiga fitur utama tersebut saling terhubung dalam satu alur sistem yang memungkinkan pengguna memperoleh informasi visual dalam bentuk audio secara *real-time*, sehingga dapat membantu pengguna dalam mengenali lingkungan sekitar dan membaca informasi.

Dalam implementasinya, sistem memanfaatkan kamera smartphone sebagai input utama untuk menangkap gambar. Data visual yang diperoleh kemudian diproses menggunakan model deteksi objek berbasis YOLO serta teknologi OCR untuk mengenali teks. Hasil dari kedua proses tersebut selanjutnya dikonversi menjadi suara menggunakan teknologi TTS, sehingga informasi dapat disampaikan secara langsung kepada pengguna dalam bentuk audio.

Selain itu, sistem juga dirancang dengan memperhatikan aspek aksesibilitas, seperti integrasi dengan fitur TalkBack, penggunaan antarmuka yang sederhana, serta pemberian umpan balik berupa suara dan getaran. Dengan demikian, aplikasi yang dikembangkan tidak hanya fungsional, tetapi juga mudah digunakan oleh penyandang gangguan penglihatan dalam mendukung aktivitas sehari-hari secara mandiri.

4.2.1 Proses Implementasi Sistem

Proses implementasi sistem dilakukan melalui beberapa tahapan sebagai berikut:

4.2.1.1 Instalasi dan Setup Lingkungan

- Menggunakan Android Studio sebagai IDE pengembangan.
- Mengintegrasikan model deteksi objek YOLO ke dalam aplikasi menggunakan TensorFlow Lite.
- Mengintegrasikan OCR menggunakan Tesseract.
- Mengintegrasikan *Text-to-Speech* menggunakan Piper TTS.

4.2.1.2 Konfigurasi Sistem

- Konfigurasi akses kamera untuk deteksi objek dan OCR.
- Pengaturan permission (kamera, audio, storage).
- Integrasi dengan fitur aksesibilitas seperti TalkBack.

4.2.1.3 Deployment Aplikasi

- Aplikasi diinstal pada perangkat Android untuk pengujian langsung.
- Pengujian dilakukan secara *real-time* menggunakan kamera perangkat.

4.2.2 Implementasi Model AI

Implementasi model kecerdasan buatan pada penelitian ini meliputi YOLOv8n sebagai model deteksi objek, Google ML Kit Text Recognition untuk proses *Optical Character Recognition* (OCR), serta Android Text-to-Speech (TTS) yang diimplementasikan menggunakan package flutter_tts. Ketiga komponen tersebut diintegrasikan ke dalam aplikasi Android berbasis Flutter sehingga mampu bekerja secara *offline* pada fungsi utamanya dan memberikan informasi kepada pengguna secara *real-time*. Model YOLOv8n digunakan untuk mengenali objek di sekitar pengguna, Google ML Kit digunakan untuk mengenali teks dari gambar, sedangkan Android Text-to-Speech digunakan untuk membacakan hasil deteksi objek maupun OCR dalam bentuk suara.

4.2.2.1 Implementasi Model YOLOv8n

Implementasi deteksi objek pada penelitian ini menggunakan model YOLOv8n (You Only Look Once Version 8 Nano) dalam format TensorFlow Lite (.tflite). Model ini dipilih karena memiliki ukuran yang ringan, kecepatan inferensi yang tinggi, serta sesuai untuk dijalankan pada perangkat Android dengan keterbatasan sumber daya. Model dijalankan menggunakan package `tflite_flutter`, sedangkan proses pengambilan citra dilakukan melalui package `camera`.

4.2.2.2 Implementasi OCR Google ML Kit

Fitur pengenalan teks pada aplikasi diimplementasikan menggunakan Google ML Kit Text Recognition. Teknologi ini memungkinkan proses OCR dilakukan secara on-device, sehingga tidak memerlukan koneksi internet ketika mengenali teks. OCR digunakan untuk membaca teks dari hasil tangkapan kamera maupun gambar yang dipilih dari penyimpanan perangkat. Sebelum proses pengenalan dilakukan, citra diproses terlebih dahulu agar memiliki kualitas yang lebih baik untuk meningkatkan akurasi pembacaan teks. Selanjutnya, hasil OCR ditampilkan pada layar aplikasi dan dapat diteruskan ke fitur Text-to-Speech agar dibacakan kepada pengguna.

4.2.2.3 Implementasi Android Text-to-Speech

Fitur Text-to-Speech (TTS) diimplementasikan menggunakan package `flutter_tts` yang memanfaatkan layanan Android Text-to-Speech pada perangkat. Teknologi ini digunakan untuk mengubah hasil deteksi objek maupun hasil OCR menjadi keluaran suara sehingga informasi visual dapat diterima oleh pengguna melalui indera pendengaran. Selain membacakan informasi secara otomatis, aplikasi juga menyediakan pengaturan kecepatan bicara sehingga pengguna dapat menyesuaikan kecepatan suara sesuai dengan kebutuhan masing-masing. Implementasi Android Text-to-Speech dipilih karena mudah diintegrasikan dengan Flutter, mendukung pemrosesan secara lokal (*offline* apabila engine TTS tersedia di perangkat), serta kompatibel dengan berbagai perangkat Android.

4.2.2.4 Integrasi ke Android

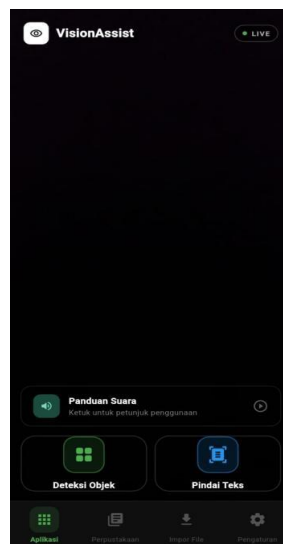
Seluruh komponen kecerdasan buatan diintegrasikan ke dalam aplikasi Android menggunakan framework Flutter. Pengambilan gambar dilakukan melalui package camera, kemudian frame diproses oleh model YOLOv8n menggunakan tflite_flutter untuk mendeteksi objek. Apabila pengguna memilih fitur OCR, gambar diproses menggunakan Google ML Kit Text Recognition untuk mengenali teks yang terdapat pada citra. Hasil deteksi objek maupun OCR selanjutnya diteruskan ke modul Android Text-to-Speech melalui package flutter_tts sehingga informasi dapat dibacakan kepada pengguna secara otomatis. Seluruh proses tersebut berlangsung secara terintegrasi sehingga aplikasi mampu memberikan informasi visual dalam bentuk audio secara *real-time* tanpa memerlukan koneksi internet untuk fungsi utama yang tersedia.

4.2.3 Tampilan dan Fitur Aplikasi

Berikut merupakan hasil implementasi antarmuka aplikasi beserta fungsinya:

1. Halaman Utama (Beranda)

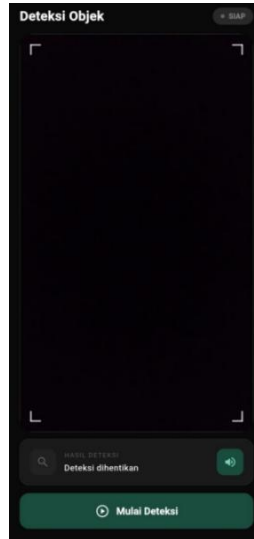
Halaman utama menampilkan area kamera *real-time* di bagian tengah layar yang siap digunakan untuk deteksi objek. Terdapat tombol Panduan Suara sebagai petunjuk penggunaan berbasis audio, serta dua tombol utama yaitu Deteksi Objek dan Pindai Teks untuk mengakses fitur inti aplikasi.



Gambar 18 Halaman Utama (Beranda)

2. Fitur Deteksi Objek

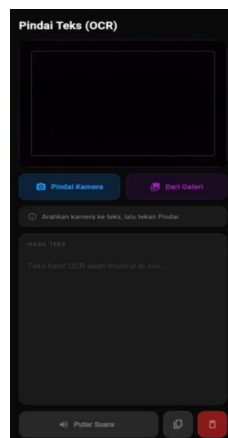
Fitur ini menggunakan kamera perangkat untuk mendeteksi objek secara *real-time* menggunakan model YOLO. Setiap objek yang terdeteksi akan ditampilkan beserta nama objek dan tingkat kepercayaan (*confidence score*), kemudian hasilnya langsung disampaikan kepada pengguna dalam bentuk suara melalui TTS.



Gambar 19 Fitur Deteksi Objek

3. Fitur Pindai Teks (OCR)

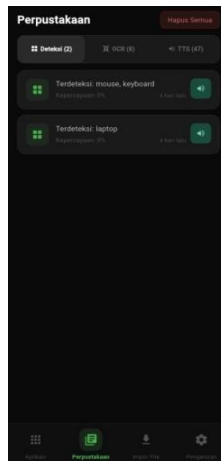
Fitur pindai teks memungkinkan pengguna mengambil gambar melalui kamera atau memilih gambar dari galeri untuk dikenali teksnya menggunakan teknologi OCR. Teks yang berhasil diekstraksi akan ditampilkan di layar dan secara otomatis dikonversi menjadi suara menggunakan TTS.



Gambar 20 Fitur Pindai Teks

4. Fitur Perpustakaan (Riwayat)

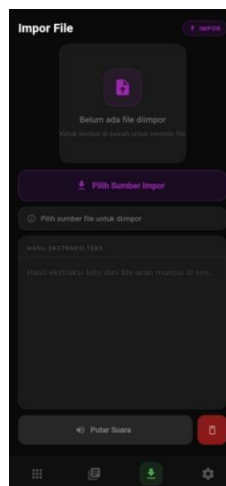
Fitur perpustakaan menyimpan seluruh riwayat aktivitas pengguna, meliputi hasil deteksi objek, OCR, dan TTS. Riwayat dapat disaring berdasarkan kategori menggunakan tab filter dan ditampilkan dalam bentuk kartu yang memuat informasi hasil, tingkat kepercayaan, serta waktu aktivitas. Pengguna dapat mendengarkan ulang informasi yang sebelumnya telah diproses.



Gambar 21 Fitur Perpustakaan

5. Fitur Impor File

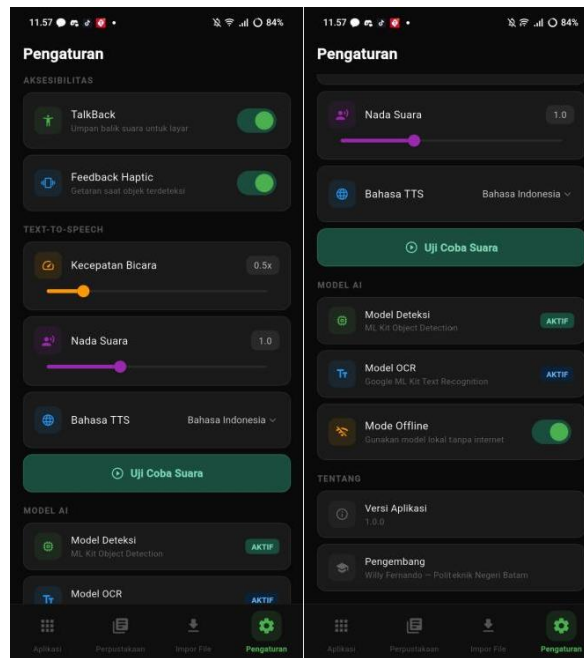
Fitur ini memungkinkan pengguna mengimpor file gambar dari penyimpanan perangkat untuk kemudian diproses menggunakan fitur OCR. Hasil ekstraksi teks ditampilkan pada area yang tersedia dan dapat diputar sebagai audio melalui tombol Putar Suara.



Gambar 22 Fitur Impor File

6. Fitur Pengaturan

Halaman pengaturan menyediakan berbagai opsi kustomisasi yang terbagi menjadi tiga kelompok, yaitu Aksesibilitas (TalkBack dan *Feedback Haptic*), *Text-to-Speech* (kecepatan bicara, nada suara, bahasa, dan uji coba suara), serta Model AI yang menampilkan informasi model deteksi yang sedang aktif.



Gambar 23 Fitur Pengaturan

4.2.4 Alur Penggunaan Sistem

Alur penggunaan sistem pada aplikasi ini dirancang sederhana agar mudah digunakan oleh penyandang gangguan penglihatan. Aplikasi terdiri dari lima fitur utama, yaitu deteksi objek, pindai teks, perpustakaan, impor file, dan pengaturan.

1. Fitur Deteksi Objek

Pengguna memilih menu deteksi objek, kemudian sistem akan mengaktifkan kamera secara otomatis. Aplikasi akan melakukan proses deteksi objek secara *real-time* menggunakan model AI. Hasil deteksi akan langsung disampaikan dalam bentuk suara, sehingga pengguna dapat mengetahui objek di sekitarnya tanpa perlu melihat layar.

2. Fitur Pindai Teks (OCR)

Pada fitur ini, pengguna dapat mengambil gambar menggunakan kamera atau memilih gambar dari galeri. Sistem kemudian akan mengenali teks yang terdapat pada gambar menggunakan teknologi OCR. Hasil teks yang diperoleh akan langsung dikonversi menjadi suara menggunakan TTS.

3. Fitur Perpustakaan

Fitur perpustakaan digunakan untuk menyimpan dan mengelola data hasil deteksi objek maupun hasil pembacaan teks. Pengguna dapat mengakses kembali riwayat tersebut untuk mendengarkan ulang informasi yang sebelumnya telah diproses.

4. Fitur Impor File

Fitur ini memungkinkan pengguna untuk mengimpor file gambar dari penyimpanan perangkat. File yang diimpor kemudian dapat diproses menggunakan fitur OCR untuk dibaca dan dikonversi menjadi suara.

5. Fitur Pengaturan

Fitur pengaturan digunakan untuk menyesuaikan preferensi aplikasi sesuai kebutuhan pengguna. Berdasarkan implementasi yang telah dibuat, menu pengaturan terdiri dari:

- Aksesibilitas, meliputi:
 - Aktivasi TalkBack untuk umpan balik suara
 - Feedback haptic berupa getaran saat objek terdeteksi

- Pengaturan *Text-to-Speech* (TTS):
 - Pengaturan kecepatan bicara
 - Pengaturan nada suara
 - Pemilihan bahasa (Bahasa Indonesia)
 - Tombol uji coba suara
- Pengaturan Model AI:
 - Status model deteksi objek
 - Status model OCR
 - Mode offline untuk penggunaan tanpa internet
- Informasi Aplikasi:
 - Versi aplikasi
 - Informasi pengembang

6. Alur Umum Penggunaan

Secara umum, pengguna membuka aplikasi, memilih fitur sesuai kebutuhan (deteksi objek atau pindai teks), kemudian sistem akan memproses input dan memberikan output berupa suara. Hasil penggunaan dapat disimpan dan diakses kembali melalui fitur perpustakaan. Pengguna juga dapat menyesuaikan pengaturan aplikasi untuk meningkatkan kenyamanan penggunaan.

4.3. Pengujian *Black Box*

Pengujian sistem dilakukan untuk memastikan bahwa seluruh fitur aplikasi berjalan sesuai dengan kebutuhan yang telah ditetapkan. Metode pengujian yang digunakan adalah ***Black Box Testing***. Black Box Testing digunakan untuk menguji fungsi sistem berdasarkan masukan dan keluaran tanpa melihat struktur kode program.

4.3.1 *Black Box Testing*

Pengujian Black Box dilakukan pada seluruh fitur utama aplikasi dengan memeriksa kesesuaian keluaran sistem terhadap masukan yang diberikan pengguna.

Tabel 15 Black Box Testing

no	Fitur	Skenario Pengujian	Input	Output yang Diharapkan	Hasil Pengujian	Status
1	Deteksi Objek	Pengguna membuka fitur deteksi objek	Menekan tombol Deteksi Objek	Kamera aktif dan sistem mulai mendeteksi objek secara <i>real-time</i>	Kamera aktif dan deteksi berjalan	Berhasil
2	OCR	Pengguna memindai gambar berisi teks	Mengarahkan kamera ke teks	Sistem mengenali teks dan menampilkan hasilnya di layar	Teks berhasil dikenali dan ditampilkan	Berhasil
3	TTS	Pengguna	Hasil teks	Sistem mengubah	Suara berhasil	Berhasil

		menjalankan pembacaan teks	dari OCR	teks menjadi suara dan memutarinya	diputar sesuai teks	
4	Riwayat	Pengguna menyimpan hasil deteksi	Hasil deteksi objek	Data tersimpan ke basis data lokal	Data riwayat berhasil disimpan	Berhasil
5	Riwayat	Pengguna membuka data riwayat	Menekan menu Perpustakaan	Sistem menampilkan daftar riwayat aktivitas	Daftar riwayat ditampilkan	Berhasil

6	Impor File	Pengguna memilih gambar dari galeri	File gambar dari penyimpanan perangkat	Sistem memproses gambar dengan OCR dan menampilkan hasil teks	Teks berhasil diekstraksi dari file	Berhasil
7	TalkBack	Pengguna mengaktifkan TalkBack	Mengaktifkan TalkBack pada pengaturan	Sistem membacakan nama dan fungsi setiap elemen antarmuka	Elemen antarmuka berhasil dibacakan	Berhasil
8	Pengaturan TTS	Pengguna mengubah kecepatan suara	Menggeser <i>slider</i> kecepatan bicara	Kecepatan suara TTS berubah sesuai pengaturan	Kecepatan suara berhasil diubah	Berhasil
9	<i>Feedback</i> Haptic	Objek berhasil terdeteksi	Objek masuk dalam jangkauan kamera	Perangkat memberikan getaran saat objek terdeteksi	Getaran berhasil dirasakan	Berhasil

BAB V KESIMPULAN

5.1. Kesimpulan

Berdasarkan hasil perancangan, implementasi, dan pengujian sistem yang telah dilakukan, maka dapat disimpulkan sebagai berikut.

1. Aplikasi teknologi bantu berbasis mobile untuk penyandang gangguan penglihatan berhasil dikembangkan sesuai dengan tujuan penelitian. Sistem mampu mengintegrasikan fitur deteksi objek, *Optical Character Recognition (OCR)*, *Text-to-Speech (TTS)*, TalkBack, dan *haptic feedback* dalam satu aplikasi sehingga informasi visual dapat disampaikan kepada pengguna dalam bentuk audio secara *real-time*.
2. Berdasarkan hasil Black Box Testing, seluruh 9 skenario pengujian (100%) memperoleh status Berhasil. Hasil tersebut menunjukkan bahwa seluruh fitur utama aplikasi telah berhasil diimplementasikan dan berfungsi sesuai dengan kebutuhan fungsional yang telah dirancang.
3. Berdasarkan hasil implementasi dan pengujian awal, aplikasi ini berpotensi membantu penyandang gangguan penglihatan dalam mengenali objek dan mengakses informasi visual melalui keluaran suara. Namun, manfaat aplikasi terhadap peningkatan kemandirian pengguna belum dapat disimpulkan secara menyeluruh karena proses Pengujian belum melibatkan penyandang *low vision* maupun tunanetra sebagai pengguna sasaran. Oleh karena itu, pengujian terhadap pengguna sasaran perlu dilakukan pada penelitian selanjutnya.

Meskipun demikian, sistem masih memiliki beberapa keterbatasan, seperti keterbatasan model deteksi objek, pengaruh kondisi lingkungan terhadap akurasi deteksi dan OCR, serta belum dilakukannya pengujian performa aplikasi secara kuantitatif. Oleh karena itu, penelitian ini masih memiliki peluang untuk dikembangkan lebih lanjut agar menghasilkan aplikasi yang lebih optimal dan sesuai dengan kebutuhan pengguna.

5.2. Saran

Berdasarkan hasil pengembangan dan pengujian sistem, maka saran yang dapat diberikan adalah sebagai berikut:

1. Penelitian selanjutnya disarankan melibatkan lebih banyak responden, khususnya penyandang *low vision*, tunanetra, atau tenaga profesional seperti guru SLB maupun terapis rehabilitasi. Hal ini bertujuan agar hasil evaluasi lebih representatif terhadap kebutuhan pengguna sasaran.
2. Model deteksi objek dapat dioptimalkan dengan menggunakan model yang lebih ringan atau lebih mutakhir sehingga akurasi dan kecepatan deteksi pada perangkat Android dapat ditingkatkan.
3. Akurasi *Optical Character Recognition* (OCR) perlu ditingkatkan, terutama untuk mengenali teks dengan tata letak yang kompleks, ukuran huruf kecil, maupun kondisi pencahayaan yang kurang baik.
4. Pengujian sistem sebaiknya dilakukan pada berbagai kondisi lingkungan, seperti *indoor* dan *outdoor*, kondisi terang maupun redup, serta variasi jarak objek terhadap kamera untuk memperoleh hasil yang lebih komprehensif.
5. Penelitian selanjutnya disarankan melakukan pengujian performa aplikasi secara kuantitatif, seperti waktu respons, penggunaan RAM, dan konsumsi baterai pada berbagai spesifikasi perangkat Android sebagai dasar optimasi sistem.
6. Untuk pengembangan jangka panjang, aplikasi dapat dikembangkan menjadi sistem yang siap digunakan secara luas (*production-ready*) dengan memperhatikan aspek keamanan data, skalabilitas sistem, serta kemudahan pemeliharaan.

Untuk pengembangan jangka panjang, sistem ini disarankan untuk dikembangkan menjadi aplikasi yang siap digunakan secara luas (*production-ready*) dengan memperhatikan aspek keamanan data, skalabilitas sistem, serta integrasi dengan layanan atau teknologi lainnya

5.3. Keterbatasan Penelitian

Penelitian ini memiliki beberapa keterbatasan yang dapat menjadi perhatian untuk pengembangan penelitian selanjutnya, yaitu sebagai berikut.

1. Model deteksi objek yang digunakan masih memanfaatkan model bawaan Google ML Kit berbasis dataset Microsoft COCO yang mampu mengenali 80 kelas objek. Oleh karena itu, aplikasi hanya dapat mendeteksi objek-objek yang tersedia pada model tersebut dan belum mampu mengenali objek khusus di luar kelas yang didukung.
2. Performa aplikasi bergantung pada spesifikasi perangkat Android yang digunakan. Pada perangkat dengan spesifikasi rendah, proses deteksi objek, OCR, dan Text-to-Speech berpotensi mengalami penurunan performa, seperti waktu respons yang lebih lambat dibandingkan perangkat dengan spesifikasi yang lebih tinggi.
3. Akurasi deteksi objek dipengaruhi oleh kondisi lingkungan, seperti intensitas pencahayaan, kualitas kamera, jarak objek terhadap kamera, serta pergerakan objek. Pada kondisi pencahayaan yang kurang baik atau objek bergerak cepat, kemampuan deteksi dapat mengalami penurunan.
4. Akurasi OCR masih memiliki keterbatasan pada teks dengan tata letak yang kompleks, ukuran huruf kecil, jenis huruf yang tidak standar, teks yang miring, maupun tulisan tangan, sehingga hasil pengenalan teks belum selalu optimal.
5. Pengujian belum melibatkan penyandang *low vision* maupun tunanetra sebagai pengguna sasaran utama aplikasi. Pengujian hanya dilakukan oleh satu orang responden yang merupakan mahasiswa Program Studi Teknik Informatika sebagai *end user*, sehingga penelitian ini belum dapat mengevaluasi pengalaman penggunaan (*user experience*) maupun tingkat penerimaan aplikasi secara langsung oleh pengguna sasaran.
6. Pengujian performa aplikasi belum dilakukan secara kuantitatif, seperti pengukuran waktu respons, penggunaan CPU dan RAM, serta konsumsi baterai. Oleh karena itu, efisiensi aplikasi pada berbagai spesifikasi perangkat Android masih perlu dievaluasi lebih lanjut.

DAFTAR PUSTAKA

AMKA, A. (2021). Pengembangan Mobile Application Berbasis Android Sebagai Alat Identifikasi Peserta Didik Berkebutuhan Khusus di Lingkungan Lahan Basah. Repository Universitas Lambung Mangkurat. <https://repository.unlam.ac.id/>

Google. (2024). Flutter Documentation. <https://docs.flutter.dev>

Herviani, V. K., Kuncahyono, Suwandayani, B. I., & Arifin, B. (2022). Pengembangan teknologi asistif "Dif-Able Apps" untuk mahasiswa dengan hambatan penglihatan dan pendengaran. *Jurnal Pendidikan Kebutuhan Khusus*, 6(1), 42–48.

Hidayat, L. (2020). Assistive Technology pada Aplikasi Android untuk Tunanetra. *Education For Exceptional Children*, 144–152.

Kementerian Koordinator Bidang Pembangunan Manusia dan Kebudayaan. (2023). Menko PMK: Penyandang Tunanetra Berhak Kerja Formal di Instansi Pemerintah. <https://www.kemenkopmk.go.id/menko-pmk-penyandang-tunanetra-berhak-kerja-formal-di-instansi-pemerintah>

Marito, Y., Tampubolon, J. M. P., Kristin, D., Sembiring, A. B., Manurung, G. S., Sitorus, R. F., & Sinaga, A. A. K. (2024). Alat Bantu yang Digunakan Anak Tuna Netra dalam Proses Pembelajaran di SLB Negeri Pembina Medan. *Jurnal Intelek dan Cendikiawan Nusantara*, 1(6), 10081–10092.

Marpaung, F., Aulia, F., & Nabila, R. C. (2022). *Computer Vision dan Pengolahan Citra Digital*. Surabaya: Pustaka Aksara.

Md Monsur Ali. (2024). How to Achieve 10x Faster, Real-Time, Offline, Human-Like Voice Text-to-Speech with Piper TTS: A Tutorial for Local and Google Colab Setup. Medium.

Patel, C. I., Patel, A., & Patel, D. (2012). Optical Character Recognition by Open Source OCR Tool Tesseract: A Case Study. *International Journal of Computer Applications*, 55(10), 50–56.

Prakash, J. (2021). Non Maximum Suppression: Theory and Implementation in PyTorch. LearnOpenCV. <https://learnopencv.com/non-maximum-suppression-theory-and-implementation-in-pytorch/>

Putra, S. (2024). Pengembangan Aplikasi Android untuk Membantu Penyandang Disabilitas Netra dalam Beraktivitas Sehari-hari. Repository Universitas Sumatera Utara. <https://www.researchgate.net/publication/381434544>

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 779–788.

World Health Organization. (2020). World Report on Vision. Geneva: World Health Organization.

World Health Organization & UNICEF. (2022). Global Report on Assistive Technology. Geneva: World Health Organization.

Google. (2024). *Android Accessibility Principles*. <https://developer.android.com/guide/topics/ui/accessibility>

World Wide Web Consortium (W3C). (2023). *Web Content Accessibility Guidelines (WCAG) 2.2*. <https://www.w3.org/TR/WCAG22/>

DAFTAR LAMPIRAN

Lampiran A. Link Produk

<https://github.com/Venthoven/VisionAssist.git>