

**IMPLEMENTASI DETEKSI OBJEK SECARA *REAL-TIME*
BERBASIS MOBILENETV2 UNTUK *AUGMENTED REALITY*
*USER MANUAL***

Skripsi



oleh:

**ALBERTUS ADRIAN SUSANTO
71190494**

**PROGRAM STUDI INFORMATIKA FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS KRISTEN DUTA WACANA**

2023

**IMPLEMENTASI DETEKSI OBJEK SECARA *REAL-TIME*
BERBASIS MOBILENETV2 UNTUK *AUGMENTED REALITY*
*USER MANUAL***

Skripsi



Diajukan kepada Program Studi Informatika Fakultas Teknologi Informasi
Universitas Kristen Duta Wacana
Sebagai Salah Satu Syarat dalam Memperoleh Gelar
Sarjana Komputer

Disusun oleh

ALBERTUS ADRIAN SUSANTO

71190494

PROGRAM STUDI INFORMATIKA FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS KRISTEN DUTA WACANA

2023

PERNYATAAN KEASLIAN SKRIPSI

Saya menyatakan dengan sesungguhnya bahwa skripsi dengan judul:

IMPLEMENTASI DETEKSI OBJEK SECARA REAL-TIME BERBASIS MOBILENETV2 UNTUK AUGMENTED REALITY USER MANUAL

yang saya kerjakan untuk melengkapi sebagian persyaratan menjadi Sarjana Komputer pada pendidikan Sarjana Program Studi Informatika Fakultas Teknologi Informasi Universitas Kristen Duta Wacana, bukan merupakan tiruan atau duplikasi dari skripsi kesarjanaan di lingkungan Universitas Kristen Duta Wacana maupun di Perguruan Tinggi atau instansi manapun, kecuali bagian yang sumber informasinya dicantumkan sebagaimana mestinya.

Jika dikemudian hari didapati bahwa hasil skripsi ini adalah hasil plagiasi atau tiruan dari skripsi lain, saya bersedia dikenai sanksi yakni pencabutan gelar kesarjanaan saya.

Yogyakarta, 30 Juni 2023



ALBERTUS ADRIAN SUSANTO
71190494

DUTA WACANA

HALAMAN PERSETUJUAN

Judul Skripsi : IMPLEMENTASI DETEKSI OBJEK SECARA
REAL-TIME BERBASIS MOBILENETV2 UNTUK
AUGMENTED REALITY USER MANUAL

Nama Mahasiswa : ALBERTUS ADRIAN SUSANTO

N I M : 71190494

Matakuliah : Skripsi (Tugas Akhir)

Kode : TI0366

Semester : Genap

Tahun Akademik : 2022/2023

Telah diperiksa dan disetujui di
Yogyakarta,
Pada tanggal 30 Juni 2023

Dosen Pembimbing I

Dosen Pembimbing II



Aditya Wikan Mahastama, S.Kom.,
M.Cs.



I Kadek Dendy S., S.T., M.Eng.

HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI
SKRIPSI/TESIS/DISERTASI UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademika Universitas Kristen Duta Wacana, saya yang bertanda tangan di bawah ini:

Nama : Albertus Adrian Susanto
NIM : 71190494
Program studi : Informatika
Fakultas : Fakultas Teknologi Informasi
Jenis Karya : Skripsi

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Kristen Duta Wacana **Hak Bebas Royalti Noneksklusif** (*None-exclusive Royalty Free Right*) atas karya ilmiah saya yang berjudul:

**“IMPLEMENTASI DETEKSI OBJEK SECARA *REAL-TIME* BERBASIS
MOBILENETV2 UNTUK *AUGMENTED REALITY USER MANUAL*”**

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti/Noneksklusif ini Universitas Kristen Duta Wacana berhak menyimpan, mengalih media/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama kami sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Yogyakarta
Pada Tanggal : 04 Juli 2023

Yang menyatakan

Adrian

(Albertus Adrian Susanto)

NIM.71190494

HALAMAN PENGESAHAN

IMPLEMENTASI DETEKSI OBJEK SECARA REAL-TIME BERBASIS MOBILENETV2 UNTUK AUGMENTED REALITY USER MANUAL

Oleh: ALBERTUS ADRIAN SUSANTO / 71190494

Dipertahankan di depan Dewan Penguji Skripsi
Program Studi Informatika Fakultas Teknologi Informasi
Universitas Kristen Duta Wacana - Yogyakarta
Dan dinyatakan diterima untuk memenuhi salah satu syarat memperoleh gelar
Sarjana Komputer
pada tanggal 16 Juni 2023

Yogyakarta, 30 Juni 2023
Mengesahkan,

Dewan Penguji:

1. Aditya Wikan Mahastama, S.Kom., M.Cs.
2. I Kadek Dendy S., S.T., M.Eng.
3. Antonius Rachmat C., S.Kom., M.Cs.
4. Yuan Lukito, S.Kom., M.Cs.



Dekan
(Restyandito, S.Kom., MSIS., Ph.D.)

Ketua Program Studi



(Gloria Virginia, Ph.D.)

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI
TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS
SECARA ONLINE
UNIVERSITAS KRISTEN DUTA WACANA YOGYAKARTA**

Saya yang bertanda tangan di bawah ini:

NIM : 71190494
Nama : Albertus Adrian Susanto
Prodi / Fakultas : Teknologi Informasi / Informatika
Judul Tugas Akhir : Implementasi Deteksi Objek secara *Real-Time*
Berbasis MobileNetV2 untuk *Augmented Reality*
User Manual

bersedia menyerahkan Tugas Akhir kepada Universitas melalui Perpustakaan untuk keperluan akademis dan memberikan **Hak Bebas Royalti Non Eksklusif** (*Non-exclusive Royalty-free Right*) serta bersedia Tugas Akhirnya dipublikasikan secara online dan dapat diakses secara lengkap (*full access*).

Dengan Hak Bebas Royalti Noneklusif ini Perpustakaan Universitas Kristen Duta Wacana berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk *database*, merawat, dan mempublikasikan Tugas Akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta. Demikian pernyataan ini saya buat dengan sebenar-benarnya.

Yogyakarta, 03 Juli 2023

Yang menyatakan,



(71190494 – Albertus Adrian Susanto)



Karya sederhana ini dipersembahkan

kepada Tuhan, Keluarga Tercinta,

dan Kedua Orang Tua



Segala sesuatu indah pada waktu-Nya

Anonim

Perjalanan ribuan mil dimulai dari langkah satu mil

(Pepatah Kuno)

KATA PENGANTAR

Segala puji dan syukur kepada Tuhan yang maha kasih, karena atas segala rahmat, bimbingan, dan bantuan-Nya maka akhirnya Skripsi dengan judul *IMPLEMENTASI DETEKSI OBJEK SECARA REAL-TIME BERBASIS MOBILENETV2 UNTUK AUGMENTED REALITY USER MANUAL* ini telah selesai disusun.

Penulis memperoleh banyak bantuan dari kerja sama baik secara moral maupun spiritual dalam penulisan Skripsi ini, untuk itu tak lupa penulis ucapkan terima kasih yang sebesar-besarnya kepada:

1. Tuhan yang maha kasih,
2. Orang tua yang selama ini telah sabar membimbing dan mendoakan penulis tanpa kenal untuk selama-lamanya,
3. Bapak Restyandito, S.Kom, MSIS., Ph.D selaku Dekan Fakultas Teknologi Informasi Universitas Kristen Duta Wacana
4. Ibu Gloria Virginia, S.Kom., MAI., Ph.D selaku Kepala Program Studi Informatika Universitas Kristen Duta Wacana
5. Bapak Aditya Wikan Mahastama, S.Kom., M.Cs. selaku Dosen Pembimbing 1, yang telah memberikan ilmunya dan dengan penuh kesabaran membimbing penulis,
6. Bapak I Kadek Dendy Senapartha. S. T., M. Eng., selaku Dosen Pembimbing 2 yang telah memberikan ilmu dan kesabaran dalam membimbing penulis,
7. Keluarga tercinta yang telah mendoakan, memberikan dukungan dan motivasi kepada penulis,
8. Teman-teman seperjuangan penulis, Bantolo Setiadi dan Michson Rabunto, serta
9. Semua pihak yang telah membantu penulis dalam menyelesaikan skripsi ini dan tidak dapat disebutkan satu persatu.

Laporan proposal/skripsi ini tentunya tidak lepas dari segala kekurangan dan kelemahan, untuk itu segala kritikan dan saran yang bersifat membangun guna

kesempurnaan skripsi ini sangat diharapkan. Semoga proposal/skripsi ini dapat bermanfaat bagi pembaca semua dan lebih khusus lagi bagi pengembangan ilmu komputer dan teknologi informasi.

Yogyakarta, 03 Juli 2023

Adrian

Penulis



DAFTAR ISI

PERNYATAAN KEASLIAN SKRIPSI.....	iii
HALAMAN PERSETUJUAN.....	iv
HALAMAN PENGESAHAN.....	v
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS SECARA ONLINE.....	vi
UNIVERSITAS KRISTEN DUTA WACANA YOGYAKARTA	vi
KATA PENGANTAR	ix
DAFTAR ISI.....	xi
DAFTAR TABEL.....	xiv
DAFTAR GAMBAR	xvi
INTISARI.....	xx
ABSTRACT.....	xxii
BAB I PENDAHULUAN	1
1.1. Latar Belakang Masalah.....	1
1.2. Perumusan Masalah	3
1.3. Batasan Masalah.....	3
1.4. Tujuan Penelitian	3
1.5. Manfaat Penelitian	4
1.6. Metodologi Penelitian	4
1.6.1. Metode Pengumpulan Data	5
1.6.2. Pengembangan Aplikasi.....	5
1.7. Sistematika Penelitian	12
BAB II TINJAUAN PUSTAKA DAN DASAR TEORI.....	14
2.1. Tinjauan Pustaka	14
2.2. Landasan Teori.....	18
2.2.1. <i>User manual</i>	18
2.2.2. Convolutional Neural Network (CNN).....	18

2.2.3. MobileNetV2	20
2.2.4. <i>Augmented Reality</i>	22
2.2.5. Multimedia Development Life Cycle (MDLC)	23
2.2.6. <i>Confusion Matrix</i>	26
2.2.7. <i>Cognitive Walkthrough</i>	27
BAB III METODOLOGI PENELITIAN	29
3.1. <i>Concept</i>	29
3.1.1. Analisis Kebutuhan Sistem	29
3.1.2. Diagram Alir Aplikasi.....	32
3.1.3. <i>Brainstorming</i>	33
3.1.4. Interview Tahap 1	34
3.1.5. <i>Interview Tahap 2</i>	34
3.1.6. <i>Interview Tahap 3</i>	35
3.1.7. <i>Structured Workshop Tahap 1</i>	36
3.1.8. <i>Structured Workshop Tahap 2</i>	36
3.1.9. <i>Structured Workshop Tahap 3</i>	38
3.2. <i>Design</i>	38
3.2.1. <i>Storyboarding</i>	38
3.2.2. <i>Prototyping</i>	42
3.3. <i>Material Collecting</i>	49
3.3.1. <i>User Manual berbasis Textbook</i>	49
3.3.2. Material untuk Objek Digital	51
3.3.3. Video Demonstrasi.....	52
3.3.4. Sampel <i>Remote Control</i>	52
BAB IV IMPLEMENTASI DAN PEMBAHASAN	56
4.1. <i>Assembly</i>	56

4.1.1. Pembuatan Model untuk Detektor Remote	56
4.1.2. Pembuatan Aplikasi	61
4.2. Testing	83
4.2.1. Hasil Pengujian Model MobileNetV2 dengan Metode <i>Confusion Matrix</i>	83
4.2.2. Hasil Pengujian <i>Usability</i> dengan Metode <i>Cognitive Walkthrough</i> ...	90
4.2.3. Pembahasan Hasil Pengujian Model MobileNetV2.....	98
4.2.4. Pembahasan Hasil Pengujian <i>Usability</i>	99
4.3. Distribution	100
BAB V KESIMPULAN DAN SARAN.....	101
5.1. Kesimpulan	101
5.2. Saran.....	101
DAFTAR PUSTAKA	103
LAMPIRAN A KODE SUMBER PROGRAM.....	106
LAMPIRAN B KARTU KONSULTASI DOSEN 1	148
LAMPIRAN C KARTU KONSULTASI DOSEN 2	149
LAMPIRAN D HASIL WAWANCARA RESPONDEN	150

DAFTAR TABEL

Tabel 1.1: Tabel <i>Task</i>	9
Tabel 1.2: Prosedur untuk <i>Task 1</i>	10
Tabel 1.3: Prosedur untuk <i>Task 2</i>	10
Tabel 1.4: Prosedur untuk <i>Task 3</i>	11
Tabel 2.1: <i>Bottleneck Residual Block</i> yang mengubah k menjadi k' <i>channel</i> dengan <i>stride</i> (s) dan <i>expansion factor</i> (t) (Sandler dkk., 2018).....	21
Tabel 2.2: Arsitektur MobileNetV2 (Paul-Louis Pröve, 2018).....	22
Tabel 2.3: Confusion Matrix	26
Tabel 3.1: Catatan <i>Brainstorming</i>	33
Tabel 3.2: Catatan <i>Interview</i> Tahap 1	34
Tabel 3.3: Catatan <i>Interview</i> Tahap 2	35
Tabel 3.4: Catatan <i>Interview</i> Tahap 3	35
Tabel 3.5: Catatan <i>Structured Workshop</i> Tahap 1	36
Tabel 3.6: Rincian Interaksi AR	37
Tabel 3.7: Catatan <i>Structured Workshop</i> Tahap 2	37
Tabel 3.8. Catatan <i>Structured Workshop</i> Tahap 3	38
Tabel 3.9: Sumber Referensi Video	52
Tabel 3.10: Referensi Program untuk Mengolah Sampel	52
Tabel 3.11: Referensi Sampel Data Sekunder.....	54
Tabel 4.1: Parameter untuk Augmentasi Gambar	56
Tabel 4.2. <i>Class</i> , <i>Variable</i> , dan <i>Method</i> untuk Menu <i>Image Path Monitor</i>	67
Tabel 4.3. <i>Class</i> , <i>Variable</i> , dan <i>Method</i> untuk Menu <i>Image Path Remote</i>	70
Tabel 4.4: Tingkat Kepercayaan Model.....	84
Tabel 4.5: Tabel <i>Confusion Matrix</i> Hasil Pengujian.....	89
Tabel 4.6: Tingkat Penyelesaian <i>Task</i> Iterasi 1.....	91
Tabel 4.7: Tingkat Penyelesaian <i>Task</i> Iterasi 2.....	91

Tabel 4.8: Rangkuman Jumlah <i>Task</i> Sukses Iterasi 1	92
Tabel 4.9: Rangkuman Jumlah <i>Task</i> Sukses Iterasi 2	92
Tabel 4.10. Jumlah Kesalahan Responden Iterasi 1	94
Tabel 4.11. Jumlah Kesalahan Responden Iterasi 2	95
Tabel 4.12. Kategori Kesalahan Iterasi 1	95
Tabel 4.13. Kategori Kesalahan Iterasi 2	96



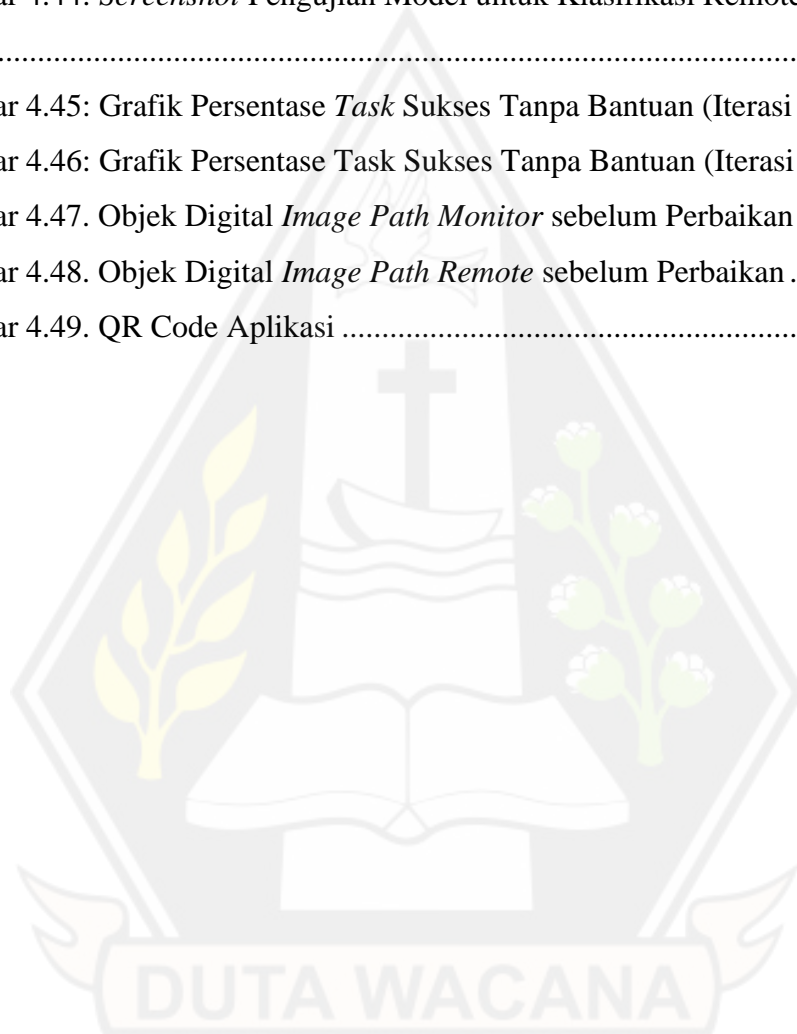
DAFTAR GAMBAR

Gambar 1.1: Diagram Alir Metodologi Penelitian.....	4
Gambar 2.1: Arsitektur Convolutional Neural Network (Qolbiyatul Lina, 2019)	19
Gambar 2.2: Representasi visual dari Convolution Layer (O’Shea & Nash, 2015)	20
Gambar 2.3: Linear Bottleneck dan Shortcut Connection dalam arsitektur MobileNetV2 (Rizqi Okta Ekoputris, 2018).....	21
Gambar 2.4: MDLC Luther	23
Gambar 3.1: <i>Flowchart</i> Menu <i>Image Path Monitor</i>	32
Gambar 3.2: <i>Flowchart</i> Menu <i>Image Path Remote</i>	32
Gambar 3.3: <i>Storyboarding</i> <i>Image Path Monitor</i>	39
Gambar 3.4: Lanjutan <i>Storyboarding</i> <i>Image Path Monitor</i>	40
Gambar 3.5: <i>Storyboarding</i> <i>Image Path Remote</i>	41
Gambar 3.6: <i>Prototype</i> Halaman <i>Splash Screen</i>	42
Gambar 3.7: <i>Prototype</i> Halaman <i>Onboarding 1</i>	43
Gambar 3.8: <i>Prototype</i> Halaman <i>Onboarding 2</i>	43
Gambar 3.9: <i>Prototype</i> Halaman <i>Onboarding 3</i>	43
Gambar 3.10: <i>Prototype</i> Halaman Utama.....	44
Gambar 3.11: <i>Prototype</i> Halaman <i>Scan Marker</i> untuk Menu <i>Image Path Monitor</i>	44
Gambar 3.12: <i>Prototype</i> Objek Digital <i>Image Path Monitor</i>	45
Gambar 3.13: <i>Prototype</i> Objek Digital Menu pada <i>Image Path Monitor</i>	45
Gambar 3.14: <i>Prototype</i> Objek Digital Fitur Utama pada <i>Image Path Monitor</i> ..	46
Gambar 3.15: <i>Prototype</i> Objek Digital Papan Petunjuk Port pada <i>Image Path</i> <i>Monitor</i>	46
Gambar 3.16: <i>Prototype</i> Objek Digital Papan Petunjuk Tombol pada <i>Image Path</i> <i>Monitor</i>	47

Gambar 3.17: <i>Prototype</i> Halaman <i>Scan Marker</i> untuk Menu <i>Image Path Monitor</i>	47
Gambar 3.18: <i>Prototype</i> Objek Digital <i>Image Path Remote</i>	48
Gambar 3.19: <i>Prototype</i> Objek Digital Papan Petunjuk Tombol pada <i>Image Path Remote</i>	49
Gambar 3.20: Petunjuk Port <i>Image Path Monitor</i>	50
Gambar 3.21. Petunjuk Tombol <i>Image Path Remote</i>	50
Gambar 3.22: Petunjuk <i>Image Path Remote</i>	51
Gambar 3.23: Video <i>Image Path Remote</i>	53
Gambar 3.24. Hasil Pengolahan Sampel Video menjadi Gambar	53
Gambar 3.25: Contoh sampel <i>Image Path Remote</i> sebelum Proses <i>Cropping</i>	54
Gambar 3.26: Contoh sampel <i>Image Path Remote</i> setelah Proses <i>Cropping</i>	54
Gambar 3.27: Contoh Sampel Remote dari Data Sekunder 1	55
Gambar 3.28: Contoh Sampel Remote dari Data Sekunder 2	55
Gambar 3.29: Contoh Sampel Remote dari Data Sekunder 3	55
Gambar 4.1: Contoh Hasil Augmentasi 1	57
Gambar 4.2: Contoh Hasil Augmentasi 2	57
Gambar 4.3: Contoh Hasil Augmentasi 3	57
Gambar 4.4: Layer pada Model	58
Gambar 4.5: Grafik <i>Training</i> dan <i>Validation Loss</i>	60
Gambar 4.6: Grafik <i>Training</i> dan <i>Validation Accuracy</i>	60
Gambar 4.7: Pembuatan 3D <i>Image Path Monitor</i>	61
Gambar 4.8: Port Monitor pada Objek 3D	62
Gambar 4.9: Tombol Monitor pada Objek 3D	62
Gambar 4.10: <i>Layout Port Monitor</i>	63
Gambar 4.11: <i>Port Audio Out</i>	63
Gambar 4.12: <i>Port Mic Out</i>	64
Gambar 4.13: Port PC USB dan Media USB	64
Gambar 4.14: <i>Port Touch USB</i>	64
Gambar 4.15: Port HDMI	64

Gambar 4.16: Layout Tombol Monitor.....	65
Gambar 4.17: Tombol Power	65
Gambar 4.18: Tombol Pengaturan	65
Gambar 4.19: Tombol <i>Volume Down</i>	66
Gambar 4.20: Tombol <i>Volume Up</i>	66
Gambar 4.21. Tombol <i>Gesture</i>	66
Gambar 4.22. <i>Layout Remote</i>	66
Gambar 4.23: <i>Screenshot</i> Halaman <i>Splash Screen</i> pada Aplikasi	74
Gambar 4.24: <i>Screenshot Onboarding Screen 1</i> pada Aplikasi.....	75
Gambar 4.25: <i>Screenshot Onboarding Screen 2</i> pada Aplikasi.....	75
Gambar 4.26: <i>Screenshot Onboarding Screen 3</i> pada Aplikasi.....	75
Gambar 4.27: <i>Screenshot</i> Menu Utama pada Aplikasi	76
Gambar 4.28: <i>Screenshot</i> Halaman Petunjuk Penggunaan Menu <i>Image Path Monitor</i> pada Aplikasi.....	76
Gambar 4.29: <i>Screenshot</i> Halaman Petunjuk Penggunaan Menu <i>Image Path Remote</i> pada Aplikasi.....	76
Gambar 4.30: <i>Screenshot</i> Tampilan Pemindaian <i>Marker</i> untuk Akses <i>User Manual Image Path Monitor</i> pada Aplikasi.....	77
Gambar 4.31: <i>Screenshot Prompt</i> Petunjuk Interaksi <i>Image Path Monitor</i> pada Aplikasi	78
Gambar 4.32: <i>Screenshot</i> Objek Digital <i>Image Path Monitor</i> pada Aplikasi.....	78
Gambar 4.33: <i>Screenshot</i> Papan Petunjuk Port Monitor pada Aplikasi 1	79
Gambar 4.34: <i>Screenshot</i> Papan Petunjuk Port Monitor pada Aplikasi 2	79
Gambar 4.35: <i>Screenshot</i> Papan Petunjuk Tombol Monitor pada Aplikasi 1	80
Gambar 4.36: <i>Screenshot</i> Papan Petunjuk Tombol Monitor pada Aplikasi 2	80
Gambar 4.37: <i>Screenshot</i> Papan berisi Fitur Utama Monitor pada Aplikasi.....	80
Gambar 4.38: <i>Screenshot</i> dari Papan yang Menampilkan Video <i>S-Write</i>	81
Gambar 4.39: <i>Screenshot</i> dari Papan yang Menampilkan Video <i>Sharing File</i> tanpa Kabel	81
Gambar 4.40: <i>Screenshot</i> Tampilan Pemindaian <i>Marker</i> untuk Akses <i>User Manual Image Path Remote</i> pada Aplikasi.....	82

Gambar 4.41: <i>Screenshot Prompt</i> Petunjuk Interaksi <i>Image Path Remote</i> pada Aplikasi	82
Gambar 4.42: <i>Screenshot</i> Objek Digital <i>Image Path Remote</i> pada Aplikasi.....	83
Gambar 4.43: <i>Screenshot</i> Pengujian Model untuk Klasifikasi <i>Image Path Remote</i>	84
Gambar 4.44: <i>Screenshot</i> Pengujian Model untuk Klasifikasi Remote dari Produk Lain	84
Gambar 4.45: Grafik Persentase <i>Task</i> Sukses Tanpa Bantuan (Iterasi 1).....	93
Gambar 4.46: Grafik Persentase <i>Task</i> Sukses Tanpa Bantuan (Iterasi 2).....	93
Gambar 4.47. Objek Digital <i>Image Path Monitor</i> sebelum Perbaikan	97
Gambar 4.48. Objek Digital <i>Image Path Remote</i> sebelum Perbaikan	98
Gambar 4.49. QR Code Aplikasi	100



INTISARI

IMPLEMENTASI DETEKSI OBJEK SECARA *REAL-TIME* BERBASIS MOBILENETV2 UNTUK *AUGMENTED REALITY USER MANUAL*

Oleh

ALBERTUS ADRIAN SUSANTO

71190494

User manual dapat membantu pengguna dalam memahami petunjuk terkait komponen maupun penggunaan suatu produk. Namun, *user manual* berbasis kertas masih dinilai rumit dan memakan waktu. Selain itu, *user manual* tersebut berpotensi untuk hilang atau rusak. Di sisi lain, PT. Miland Cipta Usaha sebagai perusahaan yang bergerak di bidang *wall display* sering sekali mengalami kesulitan pada saat melakukan demonstrasi karena produk yang ditawarkan ke klien tergolong besar dan berat.

Berdasarkan permasalahan tersebut, penelitian ini dilakukan untuk mengembangkan aplikasi *user manual* dengan menggabungkan teknologi *Augmented Reality* (AR) dan *Machine Learning* (ML). Teknologi AR digunakan untuk meningkatkan pengalaman pengguna selama menggunakan aplikasi, sedangkan teknologi ML digunakan untuk mengenali produk. Aplikasi akan dikembangkan menggunakan metode *Multimedia Development Life Cycle* (MDLC). Selain itu, penelitian ini akan menguji model yang dihasilkan oleh arsitektur MobileNetV2 dengan metode *confusion matrix*, sedangkan *usability* dari AR diuji menggunakan metode *cognitive walkthrough*. Kedua pengujian tersebut dilakukan untuk mengetahui efektivitas dari aplikasi yang dihasilkan sebagai *user manual*.

Hasil pengujian menggunakan *confusion matrix* menunjukkan model berhasil mencapai akurasi sebesar 0,7875 diikuti dengan nilai presisi sebesar 1

(satu), nilai *recall* sebesar 0,575, dan rata-rata tingkat kepercayaan (*confidence*) sebesar 0,775. Selain itu, hasil perbaikan AR berdasarkan metode *cognitive walkthrough* berhasil meningkatkan rata-rata persentase *task* sukses tanpa bantuan dari yang awalnya 47,67% menjadi 100% diikuti dengan penurunan jumlah kesalahan dari yang awalnya sebanyak 36 kali menjadi 16 kali. Berdasarkan hasil pengujian model dan *usability* AR tersebut, aplikasi yang dihasilkan dalam penelitian ini dinilai efektif untuk menggantikan *user manual* berbasis kertas.

Kata-kata kunci : *user manual, augmented reality, machine learning, MDLC, confusion matrix, cognitive walkthrough*



ABSTRACT

MOBILENETV2-BASED REAL-TIME OBJECT DETECTION IMPLEMENTATION FOR AUGMENTED REALITY USER MANUAL

By

ALBERTUS ADRIAN SUSANTO

71190494

User manuals can assist users in understanding instructions related to components and the use of a product. However, paper-based user manuals are still considered complicated and time-consuming. In addition, the user manual has the potential to be lost or damaged. On the other hand, PT. Miland Cipta Usaha as a company engaged in the field of wall displays often experiences difficulties during demonstrations because the products offered to clients are large and heavy.

Based on these problems, this research was conducted to develop user manual applications by combining Augmented Reality (AR) and Machine Learning (ML) technologies. AR technology is used to enhance the user experience while using the application, while ML technology is used to identify products. Applications will be developed using the Multimedia Development Life Cycle (MDLC) method. In addition, this study will test the model generated by the MobileNetV2 architecture using the confusion matrix method, while the usability of AR is tested using the cognitive walkthrough method. Both tests were carried out to determine the effectiveness of the application that was produced as a user manual.

The test results using the confusion matrix show that the model succeeded in achieving an accuracy of 0.7875 followed by a precision value of 1 (one), a recall value of 0.575, and an average confidence level of 0.775. In addition, the results of AR improvements based on the cognitive walkthrough method managed to increase the average percentage of successful tasks without assistance from initially 47.67%

to 100% followed by a decrease in the number of errors from initially 36 times to 16 times. Based on the results of testing the AR model and usability, the application produced in this study is considered effective in replacing paper-based user manuals.

Keywords : user manual, augmented reality, machine learning, MDLC, confusion matrix, cognitive walkthrough



BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Teknologi *Augmented Reality* (AR) terus mengalami perkembangan seiring dengan berjalannya waktu. Perkembangan tersebut membuat AR dapat digunakan dengan lebih bebas dan mudah (*mobile*). Pada awalnya AR hanya berkembang pada *wearable device*, tetapi saat ini AR dapat digunakan di *smartphone* (Arth dkk., 2015).

Saat ini, banyak bidang telah menggunakan AR, seperti pemasaran, hiburan, tamasya, industri, *fashion*, permainan, dan obat-obatan. Dalam bidang pemasaran, AR dapat digunakan untuk memamerkan produk terbaru dan meningkatkan potensi penjualan produk. Perusahaan MINI menggunakan AR untuk memasarkan produk mereka dengan menyediakan foto 3D dari mobil yang dirilis pada saat itu (Berryman, 2012).

User manual berfungsi untuk membantu pengguna dalam memahami komponen maupun memberikan petunjuk terkait penggunaan produk. Dalam beberapa kasus, *user manual* berbasis kertas dinilai rumit dan memakan waktu apabila pengguna ingin mencari petunjuk yang tepat untuk tugas tertentu (Kolla dkk., 2021). Selain itu, *user manual* berbasis kertas dinilai dapat membingungkan sebagian konsumen (Widiyanto & Rifa, 2014). Berdasarkan perkembangan AR saat ini dan manfaatnya di berbagai bidang, peneliti ingin menggunakan teknologi AR untuk mengembangkan aplikasi yang berfungsi sebagai *user manual*. Selain itu, teknologi ini dinilai dapat memberikan *user experience* yang lebih baik dalam menampilkan *user manual* (Müller dkk., 2013). Dengan demikian, aplikasi *user manual* berteknologi AR diharapkan dapat memudahkan konsumen dalam memahami komponen dan petunjuk penggunaan produk.

PT. Miland Cipta Usaha adalah perusahaan yang bergerak di bidang IT dan menyediakan berbagai macam produk, seperti *video wall display*, *video wall processor*, dan *control system*. Dalam rangka meningkatkan penjualan, tim

marketing perlu melakukan promosi kepada pengguna dengan mendemonstrasikan produk. Namun, tim *marketing* sering sekali mengalami kesulitan karena ukuran produk yang tergolong besar dan berat untuk dibawa ke mana-mana.

Berdasarkan masalah yang dialami oleh tim *marketing* tersebut, penelitian ini perlu dilakukan untuk menciptakan aplikasi dengan teknologi AR yang dapat membantu tim *marketing* dalam mendemonstrasikan produk tanpa perlu membawanya. Dengan demikian, aplikasi *user manual* ini diharapkan juga dapat membantu tim *marketing* dalam mempromosikan produk sehingga meningkatkan potensi penjualan produk.

Aplikasi *user manual* berteknologi AR yang ingin dikembangkan pada dasarnya membutuhkan *deep learning* dalam proses pembuatannya. *Deep learning* sebagai salah satu cabang dari *machine learning* telah memberikan banyak kontribusi dalam bidang *computer vision* (Wu dkk., 2017). CNN (*Convolutional Neural Network*) merupakan salah satu algoritma *deep learning* yang banyak digunakan di bidang pengenalan pola dalam gambar (O'Shea & Nash, 2015). Salah satu arsitektur yang dikembangkan dari algoritma CNN yaitu MobileNetV2. Arsitektur ini dirancang agar dapat digunakan di perangkat *mobile* (Sandler dkk., 2018).

Dalam penelitian ini, aplikasi yang dihasilkan akan terintegrasi dengan model untuk deteksi objek secara *real-time*. Sampel yang akan digunakan dalam penelitian ini berupa gambar produk dari PT. Miland Cipta Usaha dan gambar produk serupa dari perusahaan lain. Dataset dibuat dengan mengolah citra yang ada pada sampel. Selanjutnya, dataset dibagi menjadi 80% sebagai data uji, sedangkan 20% akan digunakan sebagai data uji dan data validasi. Proses pelatihan untuk menghasilkan model dilakukan menggunakan arsitektur MobileNetV2. Setelah proses pembuatan model berhasil, maka proses selanjutnya yaitu implementasi AR pada aplikasi.

1.2. Perumusan Masalah

Berdasarkan uraian latar belakang di atas, maka masalah yang akan diteliti pada skripsi ini dapat dirumuskan sebagai berikut:

1. Seberapa baik akurasi model yang dibangun menggunakan arsitektur MobileNetV2 digunakan untuk deteksi objek pada aplikasi AR?
2. Seberapa efektif teknologi AR digunakan untuk *user manual* produk?

1.3. Batasan Masalah

Pada penelitian ini akan dibatasi hal-hal sebagai berikut:

1. Produk yang akan digunakan sebagai sampel penelitian adalah *Interactive Collaboration Tool* dari PT. Miland Cipta Usaha berupa *Image Path Monitor* beserta dengan remotenya yang akan disebut sebagai *Image Path Remote*,
2. *User manual* untuk *Image Path Monitor* akan memanfaatkan deteksi gambar menggunakan teknologi *Augmented Image*, sedangkan *user manual* untuk *Image Path Remote* akan memanfaatkan deteksi objek,
3. *User manual* untuk *Image Path Monitor* hanya akan berfokus pada bagian *front panel* beserta dengan menu di dalamnya,
4. Model yang dibuat untuk deteksi objek remote hanya akan berfokus untuk mengenali *Image Path Remote* dari sisi depan, dan
5. Penelitian ini hanya dilakukan pada perangkat Android yang didukung oleh AR Core.

1.4. Tujuan Penelitian

Tujuan utama dari penelitian ini yaitu mengembangkan dan menguji efektivitas aplikasi berteknologi AR menggunakan deteksi objek secara *real-time* untuk menampilkan *user manual* produk. Implementasi dari deteksi objek secara *real-time* akan digunakan sebagai *marker* untuk menampilkan *user manual* produk dalam bentuk AR.

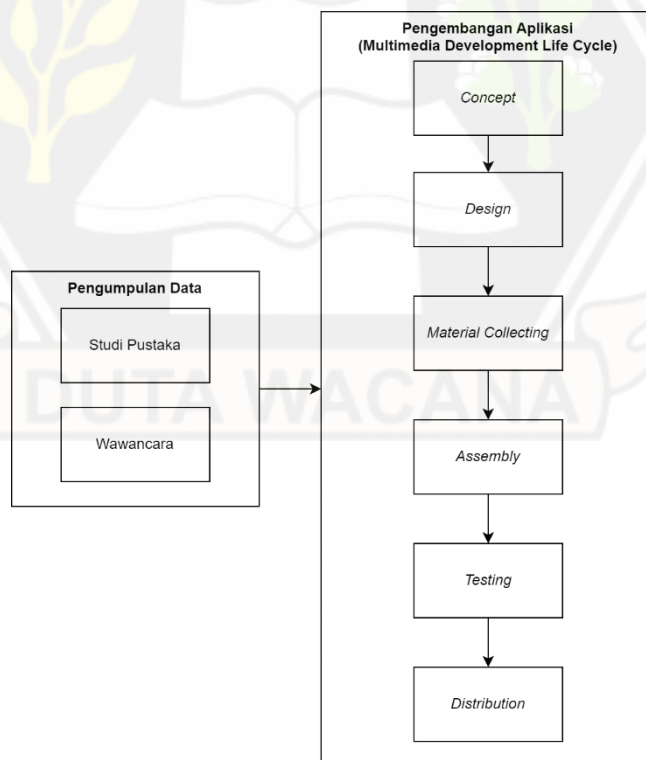
1.5. Manfaat Penelitian

Berikut adalah manfaat yang dapat diperoleh dari penelitian ini:

1. Hasil dari penelitian pada skripsi ini diharapkan dapat digunakan sebagai alternatif pengganti *user manual* berbasis kertas,
2. Hasil penelitian diharapkan dapat membantu penelitian berikutnya dalam mengembangkan aplikasi *user manual* berbasis AR dengan memanfaatkan deteksi objek secara *real-time*,
3. Hasil penelitian diharapkan dapat membantu pembaca untuk mengetahui akurasi arsitektur MobileNetV2 dalam mengenali benda secara *real-time*, dan
4. Hasil dari penelitian ini diharapkan dapat diterapkan untuk produk lain.

1.6. Metodologi Penelitian

Penelitian pada skripsi ini akan mengikuti alur metodologi penelitian seperti pada Gambar 1.1.



Gambar 1.1: Diagram Alir Metodologi Penelitian

Berikut merupakan metode yang akan digunakan untuk masing-masing alur penelitian pada skripsi ini:

1.6.1. Metode Pengumpulan Data

Metode pengumpulan data yang digunakan dalam penelitian ini yaitu studi literatur dan wawancara.

1. Studi Literatur

Studi literatur terkait AR dan deteksi objek dilakukan sebagai dasar dalam membangun aplikasi. Selain itu, peneliti juga melakukan studi literatur terkait *user manual* dari produk untuk menentukan bentuk interaksi dari AR yang akan diimplementasikan pada aplikasi.

Dalam melakukan persiapan untuk membangun aplikasi, peneliti melakukan studi literatur terkait aplikasi-aplikasi yang memanfaatkan teknologi AR. Studi literatur tersebut dilakukan untuk mengetahui pemanfaatan AR di berbagai bidang dan mendapat gambaran terkait interaksi yang mungkin dilakukan dengan teknologi tersebut. Selain itu, perlu dilakukan juga studi literatur terkait arsitektur yang ringan dan akurat untuk membangun model yang dapat ditanamkan di aplikasi *mobile*.

2. Wawancara

Selain itu, peneliti juga perlu melakukan wawancara terhadap salah satu perwakilan dari PT. Miland Cipta Usaha untuk mendapatkan informasi secara rinci terkait produk, serta bentuk interaksi dari *user manual* ingin ditampilkan dengan teknologi AR.

1.6.2. Pengembangan Aplikasi

Metode yang akan digunakan untuk mengembangkan aplikasi yaitu MDLC (*Multimedia Development Life Cycle*). Metode ini terdiri dari beberapa tahapan, yaitu *concept, design, material collecting, assembly, testing, dan distribution*. Berikut merupakan rincian dari masing-masing tahapan yang akan dilakukan selama proses pengembangan aplikasi tersebut, yaitu:

1. *Concept*

Tahap ini dilakukan untuk mengetahui target pengguna dan tujuan dari pembuatan aplikasi. Selain itu, tahap ini dilakukan untuk mengetahui spesifikasi dan rancangan dari aplikasi, serta *requirement* yang dibutuhkan dalam membangun aplikasi. Ada tiga hal yang dapat dilakukan untuk mencapai tujuan pada tahap ini, yaitu:

a) *Brainstorming*

Diskusi (*brainstorming*) dengan perwakilan dari PT. Miland Cipta Usaha perlu dilakukan untuk mendapatkan informasi terkait produk yang akan digunakan dalam penelitian ini. Hasil dari diskusi tersebut akan memberikan gambaran terkait bentuk interaksi *user manual* yang akan ditampilkan dengan menggunakan teknologi AR.

b) *Interview*

Wawancara (*interview*) dengan perwakilan dari PT. Miland Cipta Usaha perlu dilakukan untuk mengumpulkan *requirement* yang diperlukan dalam membangun aplikasi.

c) *Structured Workshop*

Structured workshop perlu diadakan agar perwakilan dari PT. Miland Cipta Usaha dapat memperoleh gambaran mengenai aplikasi yang akan dikembangkan dalam penelitian ini. Selain itu, *workshop* ini diperlukan untuk mendapatkan saran dan masukan, serta mencegah terjadinya miskonsepsi.

2. *Design*

Tahap ini perlu dilakukan untuk mendapatkan gambaran mengenai arsitektur, alur, dan antarmuka dari aplikasi.

a) *Storyboarding*

Storyboarding dari aplikasi perlu dibuat untuk memberikan gambaran terkait alur aplikasi dari awal hingga akhir.

b) *Prototyping*

Setelah mendapat gambaran mengenai alur aplikasi, *prototype* perlu dibuat untuk memberikan gambaran kepada

perwakilan dari PT. Miland Cipta Usaha dan memberikan kemudahan dalam membangun aplikasi.

3. *Material Collecting*

Pada tahap ini, bahan-bahan perlu dikumpulkan untuk membangun model, serta menampilkan objek digital yang interaktif dan membantu pengguna dalam mendapatkan petunjuk terkait produk.

Dalam pembangunan model, sampel data yang akan dikumpulkan sebagai bahan untuk pembuatan dataset terbagi ke dalam dua kategori, yaitu data primer dan data sekunder. Data primer yang akan digunakan dalam penelitian ini merupakan gambar dan video dari *Image Path Remote* yang diambil dengan menggunakan *smartphone*. Selanjutnya, proses augmentasi perlu dilakukan untuk memperbanyak data tersebut. Selain itu, dalam penelitian ini, data sekunder yang akan digunakan berupa gambar-gambar *remote control* yang diambil dari internet.

Ikon-ikon juga perlu dikumpulkan sebagai bahan untuk membuat objek-objek digital yang interaktif. Ikon tersebut akan diambil dari berbagai sumber di internet.

4. *Assembly*

Tahap ini perlu dilakukan untuk menghasilkan aplikasi yang akan diuji dalam penelitian ini. Ada dua hal yang perlu dilakukan untuk tujuan dari tahap ini, yaitu:

a) Pembuatan Model untuk Deteksi *Remote*

Sampel yang dikumpulkan pada tahap sebelumnya akan diolah menjadi dataset yang terbagi ke dalam dua kategori, yaitu dataset untuk *Image Path Remote* dan dataset dari *remote control* yang lain sebagai pembandingan. Selanjutnya, dataset tersebut akan dilatih menggunakan arsitektur MobileNetV2 untuk menghasilkan model yang dapat mengklasifikasikan objek. Setelah itu, peneliti akan melakukan evaluasi terhadap model untuk melihat akurasi yang dihasilkan oleh arsitektur MobileNetV2. Apabila akurasi dari

model yang dihasilkan sudah sesuai kebutuhan, maka model akan diekspor ke dalam bentuk TFLite untuk diintegrasikan pada aplikasi Android. Selama melakukan tahap ini, peneliti akan menggunakan Jupyter Notebook dan Google Colab, serta library Tensorflow Lite Model Maker.

b) Pembuatan Aplikasi

Pada tahap ini, aplikasi akan dibangun dengan menggunakan Android Studio. Selain itu, *package* dari Sceneform dan ML Kit perlu diinstal ke dalam proyek Android. Sceneform digunakan untuk mengimplementasikan teknologi AR, sedangkan ML Kit digunakan untuk mengintegrasikan model pada aplikasi.

5. *Testing*

Pada tahap ini, pengujian akan dilakukan untuk menjawab rumusan masalah yang menjadi dasar dalam penelitian ini. Pengujian dibagi menjadi dua, yaitu pengujian untuk model yang dihasilkan oleh arsitektur MobileNetV2 dan pengujian untuk *usability* dari AR.

Pengujian model perlu dilakukan untuk melihat akurasi yang dihasilkan dalam mendeteksi dan mengklasifikasikan objek sesuai dengan produk yang menjadi sampel dalam penelitian ini. Pengujian tersebut dilakukan dengan menggunakan metode *confusion matrix*. Metode ini akan mengukur *accuracy*, *precision*, dan *recall* yang dihasilkan oleh sistem.

Selanjutnya, *usability* dari aplikasi juga perlu diuji dengan menggunakan metode *Cognitive Walkthrough*. Pengujian ini bertujuan untuk mengetahui efektivitas AR sebagai *user manual*. Metode ini akan mencatat tingkat penyelesaian *task*, jumlah kesalahan yang terjadi, serta cerita atau pendapat dari responden saat menggunakan aplikasi. Berikut adalah beberapa tahap untuk melakukan pengujian ini, yaitu:

a) Penentuan responden

Menurut Nielsen (2012) dalam (Putri dkk., 2021), jumlah responden untuk pengujian *usability* adalah 5 (lima) orang. Selain

itu, 5 (lima) responden dapat menunjukkan 80% dari semua masalah *usability* (Alroobaea & Mayhew, 2014). Oleh karena itu, jumlah responden dalam penelitian ini terdiri dari 14 orang dan merupakan mahasiswa dari program studi Informatika di Universitas Kristen Duta Wacana. Masing-masing iterasi memerlukan tujuh orang mahasiswa sebagai responden. Secara umum, responden yang dipilih memiliki wawasan yang cukup luas di bidang teknologi dan memiliki kemahiran dalam mengoperasikan *smartphone*.

b) Pembuatan *task*

Ada beberapa *task* yang perlu dibuat untuk dikerjakan oleh responden pada saat melakukan pengujian terhadap aplikasi *usability* dari AR. Pada pengujian ini, ada tiga *task* yang mengharuskan responden untuk menggunakan aplikasi agar bisa menyelesaikan *task* tersebut. *Task* pertama dan kedua berkaitan dengan penggunaan aplikasi sebagai *user manual* untuk Image Path Monitor, sedangkan *task* ketiga berkaitan dengan penggunaan aplikasi sebagai *user manual* untuk Image Path Remote. Tabel 1.1 merupakan skenario *task* yang sudah peneliti buat untuk dikerjakan oleh responden.

Tabel 1.1: Tabel *Task*





No	Tujuan	Tahapan
1	Mencari tahu fungsi setiap <i>port</i> dan tombol yang ada di <i>front panel Miland Image Path Monitor</i>	18
2	Mendapatkan informasi terkait penggunaan fitur/menu <i>whiteboard</i> yang ada pada Image Path Monitor	7
3	Melakukan <i>screenshot</i>	8

c) Pembuatan prosedur untuk menyelesaikan setiap *task*

Dalam melakukan pengujian, perlu adanya prosedur yang tepat untuk menyelesaikan setiap *task*. Selama pengujian, observasi

perlu dilakukan untuk membandingkan langkah-langkah yang dilakukan oleh pengguna dengan prosedur yang telah disusun sebelumnya.



Tabel 1.2: Prosedur untuk *Task 1*

(a)	Buka aplikasi
(b)	Pilih opsi menu Image Path Monitor
(c)	Baca petunjuk penggunaan aplikasi
(d)	Scan gambar kartu Miland sesuai dengan frame yang telah tersedia
(e)	Tunggu hingga objek digital berupa 3D Image Path Monitor muncul
(f)	Baca petunjuk penggunaan aplikasi (tombol-tombol)
(g)	Baca informasi terkait port audio out: (i) Pilih tombol dengan icon  (ii) Baca informasi terkait fungsi port tersebut
(h)	Baca informasi terkait port touch USB: (i) Pilih tombol dengan icon  (ii) Baca informasi terkait fungsi port tersebut
(i)	Baca informasi terkait tombol setting: (i) Pilih tombol dengan icon  (ii) Baca informasi terkait fungsi port tersebut
(j)	Baca informasi terkait tombol gesture: (i) Pilih tombol dengan icon  (ii) Baca informasi terkait fungsi port tersebut

Tabel 1.2 menunjukkan prosedur yang benar untuk menyelesaikan *task 1*.


Tabel 1.3: Prosedur untuk *Task 2*

(a)	Buka aplikasi
(b)	Pilih opsi menu Image Path Monitor
(c)	Baca petunjuk penggunaan aplikasi

- (d) Scan gambar kartu Miland sesuai dengan frame yang telah tersedia
- (e) Tunggu hingga objek digital berupa 3D Image Path Monitor muncul
- (f) Menyalakan 3D Image Path Monitor menggunakan tombol power
 - (i) Pilih tombol dengan icon  untuk menyalakan 3D Image Path Monitor
- (g) Mencari petunjuk penggunaan fitur/menu Whiteboard
 - (i) Pilih tombol dengan icon  untuk menonton video petunjuk penggunaan fitur/menu whiteboard

Tabel 1.3 menunjukkan prosedur yang benar untuk menyelesaikan *task 2*.

Tabel 1.4: Prosedur untuk *Task 3*

- (a) Buka aplikasi
- (b) Pilih opsi menu Image Path Remote
- (c) Baca petunjuk penggunaan aplikasi
- (d) Scan objek remote hingga muncul objek digital dari Image Path Remote
- (e) Baca informasi terkait tombol snapshot
 - (i) Pilih tombol dengan icon 
 - (ii) Baca informasi terkait fungsi tombol tersebut
- (f) Tekan tombol *snapshot* yang ada di Image Path Remote untuk melakukan *screenshot*

Tabel 1.4 menunjukkan prosedur yang benar untuk menyelesaikan *task 3*.

d) Analisis hasil pengujian

Pada tahapan ini, pengujian efektivitas dari teknologi AR untuk menampilkan *user manual*. Pengujian efektivitas dilakukan dengan melihat tingkat keberhasilan pengguna dalam menyelesaikan setiap *task* dengan benar. Efektivitas dapat diketahui

dengan dari banyaknya aksi yang perlu dilakukan oleh pengguna untuk mencapai suatu tujuan. Banyaknya aksi tersebut dapat dilihat dari kesalahan-kesalahan yang terjadi pada saat pengguna menggunakan aplikasi. Selain itu, efektivitas juga dapat diukur dari banyaknya tugas yang diselesaikan dalam interval waktu tertentu (Şengel, 2013).

e) Identifikasi masalah dan rekomendasi perbaikan

Pada tahapan ini, peneliti mengidentifikasi masalah-masalah yang muncul pada saat proses pengujian berlangsung. Identifikasi masalah dilakukan dengan melihat kesesuaian antara skenario dengan langkah-langkah yang dilakukan responden. Setiap masalah yang dialami pengguna dalam menjalankan *task* akan dicatat oleh peneliti. Setelah itu, responden juga akan diminta untuk menceritakan pengalamannya selama berinteraksi dengan AR serta memberikan saran yang dapat menjadi bahan pertimbangan untuk perbaikan.

f) Perbaikan AR

Setelah mengetahui permasalahan yang terjadi dan memperoleh berbagai saran dari responden, peneliti melakukan perbaikan terhadap antarmuka dari AR untuk meminimalisir kesalahan dan kegagalan responden dalam menggunakan aplikasi.

6. *Distribution*

Pada tahap ini, peneliti akan menyimpan *source code*, *notebook*, dan aplikasi ke dalam Github dan Google Drive.

1.7. Sistematika Penelitian

Penelitian dalam skripsi ini terdiri atas lima bab dengan sistematika sebagai berikut:

BAB I. PENDAHULUAN

Bab ini berisi uraian terkait latar belakang, perumusan dan batasan masalah, tujuan penelitian, manfaat penelitian, serta metodologi penelitian.

BAB II. TINJAUAN PUSTAKA DAN LANDASAN TEORI

Bab ini berisi uraian dari tinjauan pustaka tentang penelitian-penelitian terkait serta teori-teori yang menjadi landasan dalam penelitian ini.

BAB III. METODOLOGI PENELITIAN

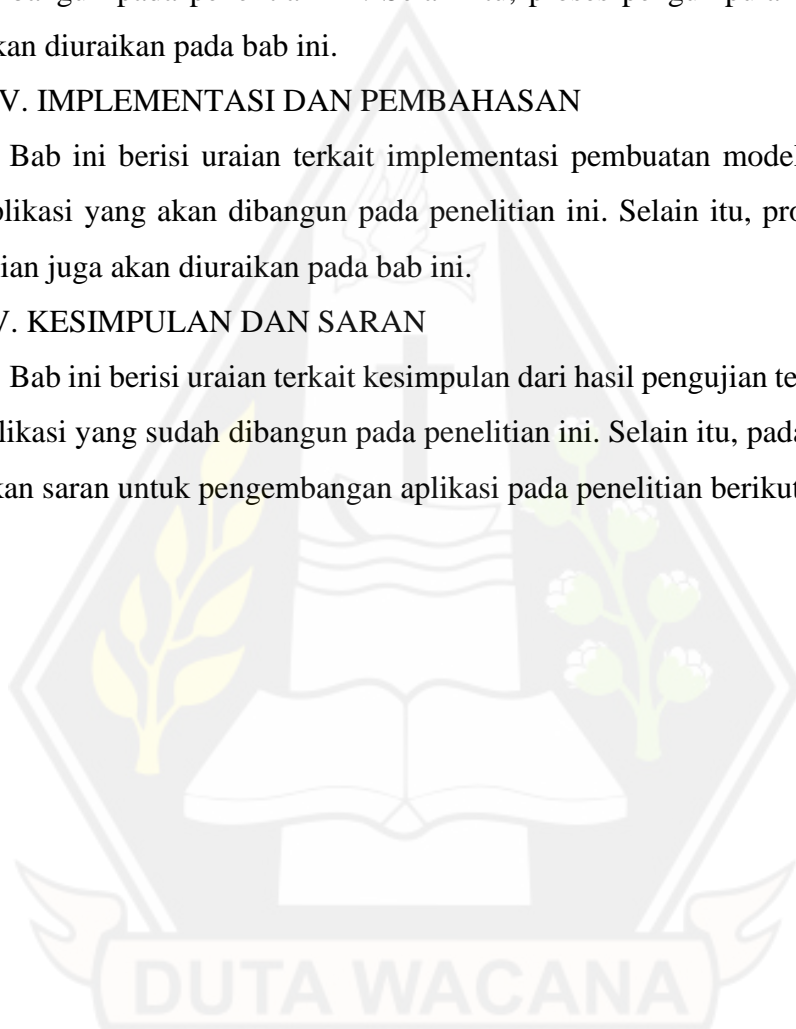
Bab ini berisi uraian terkait konsep dan desain/rancangan dari aplikasi yang akan dibangun pada penelitian ini. Selain itu, proses pengumpulan bahan-bahan juga akan diuraikan pada bab ini.

BAB IV. IMPLEMENTASI DAN PEMBAHASAN

Bab ini berisi uraian terkait implementasi pembuatan model dan aplikasi dari aplikasi yang akan dibangun pada penelitian ini. Selain itu, proses dan hasil pengujian juga akan diuraikan pada bab ini.

BAB V. KESIMPULAN DAN SARAN

Bab ini berisi uraian terkait kesimpulan dari hasil pengujian terhadap model dan aplikasi yang sudah dibangun pada penelitian ini. Selain itu, pada bab ini akan diberikan saran untuk pengembangan aplikasi pada penelitian berikutnya.



BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1. Tinjauan Pustaka

Penelitian terkait implementasi AR sebagai *user manual* masih tergolong sedikit. Penelitian ini penting karena *user manual* dalam bentuk buku (*textbook*) dirasa masih menyulitkan pengguna dalam memahami berbagai hal terkait produk, mulai dari komponen hingga petunjuk penggunaan. Selain itu, berbagai penelitian sebelumnya menunjukkan jika *user manual* dalam bentuk AR dinilai dapat memberikan *user experience* yang lebih baik dan dapat meningkatkan potensi penjualan produk. Pada umumnya aplikasi *user manual* lain memanfaatkan gambar/kode QR sebagai *trigger* untuk menampilkan objek (*Augmented Image*). Namun, seiring berjalannya waktu, teknologi untuk melakukan deteksi objek secara *real-time* sudah semakin cepat, akurat, dan ringan. Oleh karena itu, peneliti ingin mengembangkan aplikasi untuk menampilkan *user manual* suatu produk dengan menggabungkan teknologi AR dan *real-time object detection*.

Widiyanto dan Rifa'I mengembangkan aplikasi yang berfungsi sebagai *user manual* dengan menampilkan proses perakitan mainan anak dengan menggunakan teknologi AR. Teknologi ini dinilai dapat menampilkan objek virtual ke dalam perspektif lingkungan asli. Dalam penelitian tersebut, mereka menggunakan *marker model* sebagai *trigger* untuk AR menampilkan proses perakitan mainan. Sampel yang digunakan dalam penelitian ini yaitu mainan anak yang dihasilkan oleh perusahaan bernama KUBE. Pengujian dilakukan dengan menggunakan dua perangkat yaitu laptop dan komputer yang disertai dengan *webcam*. Hasil dari pengujian sistem menunjukkan AR dapat menampilkan animasi proses perakitan mainan dengan mengarahkan *marker* ke *webcam* (Widiyanto & Rifa, 2014).

Di sisi lain, aplikasi GuideMe yang dikembangkan oleh M'uller, dkk menggunakan AR untuk menampilkan *user manual*. Sampel yang digunakan dalam penelitian ini yaitu peralatan rumah tangga. Pengujian dilakukan dengan membandingkan aplikasi GuideMe dengan *user manual* dalam bentuk kertas dan

video. Hasil pengujian menunjukkan aplikasi GuideMe lebih disukai oleh 9 dari 10 orang partisipan karena *user experience* yang lebih baik (Müller dkk., 2013).

Selain itu, penelitian ini memerlukan fitur deteksi objek secara *real-time* untuk meningkatkan kemudahan dalam menggunakan aplikasi. Dalam penelitian yang dilakukan oleh Nagrath, dkk, mereka menggunakan *deep learning*, TensorFlow, Keras, dan OpenCV untuk mendeteksi masker. Dalam penelitian tersebut, mereka menggunakan *Single Shot Multibox Detector* sebagai pendeteksi wajah, sedangkan arsitektur MobilenetV2 digunakan untuk klasifikasi. Arsitektur MobilenetV2 dinilai sangat ringan sehingga dapat digunakan oleh *embedded devices* seperti Jetson Nano dan Raspberry pi untuk melakukan deteksi masker secara *real-time*. Sampel yang digunakan dalam penelitian ini terdiri atas data set yang diambil dan dikumpulkan dari berbagai macam sumber. Hasil dari penelitian tersebut menunjukkan akurasi sebesar 0,9264 dan skor F1 sebesar 0,93 untuk mengenali masker (Nagrath dkk., 2021).

Yuli Cahyaningsih mengembangkan aplikasi untuk promosi berbasis Android dengan memanfaatkan teknologi AR. Penelitian tersebut bertujuan untuk mengembangkan media alternatif untuk promosi produk di Astra Motor Majenang. Aplikasi yang dihasilkan dapat menampilkan objek 3D motor beserta dengan spesifikasinya dalam bentuk teks dan audio. Dalam mengembangkan aplikasi tersebut, Yuli menggunakan metode MDLC (*Multimedia Development Life Cycle*). Berdasarkan hasil pengujian *alpha*, aplikasi dapat berjalan dengan baik tanpa cacat, sedangkan hasil pengujian *beta* menunjukkan 91,2% responden setuju apabila aplikasi AR dinilai sangat menarik sebagai media promosi pada brosur Honda Genio dan dapat diterima oleh pengguna (Cahyaningsih, 2020).

Di sisi lain, evaluasi perlu dilakukan untuk menguji *usability* dari aplikasi AR yang dihasilkan dari penelitian ini. Putri, dkk menggunakan metode *Cognitive Walkthrough* untuk menguji website forum diskusi daring dalam penelitiannya. Penelitian tersebut bertujuan untuk mengetahui desain antarmuka website forum diskusi yang ramah pengguna. Berdasarkan hasil penelitian tersebut, terdapat penurunan tingkat kesalahan dan peningkatan tingkat penyelesaian skenario tugas

terhadap hasil perbaikan tampilan web setelah diuji dengan metode *Cognitive Walkthrough* (Putri dkk., 2021).

Rangkuman dari penelitian-penelitian terdahulu dapat dilihat pada tabel x.

No	Penulis	Tahun	Judul/Domain Masalah Penelitian	Metode	Hasil
1.	Widiyanto & Rifa	2014	<i>User Manual with Augmented Reality to Support Packaging Products</i>	<i>Marker-based tracking</i>	AR dapat menampilkan animasi proses perakitan mainan dengan mengarahkan marker ke webcam.
2.	Müller dkk.	2013	<i>GuideMe: A mobile augmented reality system to display user manuals for home appliances.</i>	NASA TLX	Aplikasi disukai 9 dari 10 partisipan karena <i>user experience</i> yang baik.
3.	Nagrath dkk.	2021	<i>SSDMNV2: A real time DNN-based face mask detection system using single shot multibox</i>	<i>Confusion matrix, AUC, classification report, perbandingan antar model</i>	Model yang dibangun dengan SSD dan MobileNetV2 menghasilkan akurasi sebesar 0,9264 dan skor F1 sebesar 0,93

			<i>detector and MobileNetV2.</i>		
4.	Cahyaningsih	2020	Teknologi Augmented Reality pada Promosi Berbasis Android.	MDLC (<i>Multimedia Development Life Cycle</i>)	Aplikasi dapat berjalan dengan baik tanpa cacat (<i>alpha test</i>) dan 91,2% responden setuju apabila aplikasi AR menarik sebagai media promosi dan dapat diterima oleh pengguna (<i>beta test</i>).
5.	Putri dkk.	2021	Uji Kegunaan Perangkat Lunak menggunakan Metode <i>Cognitive Walkthrough</i> Studi Kasus Web Forum PKL 60 Politeknik Statistika STIS (<i>Software Usability Testing using</i>	<i>Cognitive Walkthrough</i>	Hasil perbaikan tampilan web setelah diuji dengan metode <i>Cognitive Walkthrough</i> menunjukkan terjadinya penurunan tingkat kesalahan dan peningkatan tingkat penyelesaian skenario tugas

			<i>Cognitive Walkthrough : A Case Study of PKL 60 Politeknik Statistika STIS Web Forum)</i>		
--	--	--	---	--	--

Berbeda penelitian-penelitian terdahulu, penelitian ini menggabungkan teknologi AR dengan *machine learning* untuk deteksi objek secara *real-time*. Jika pada penelitian-penelitian sebelumnya *marker* yang digunakan hanya dalam bentuk gambar, maka penelitian ini akan memanfaatkan objek nyata sebagai *marker*. Pada penelitian ini, model yang dibangun menggunakan arsitektur MobileNetV2 akan diintegrasikan pada aplikasi Android sehingga perlu ada teknik tertentu agar tidak terjadi penurunan yang signifikan dalam hal akurasi. Selain itu, penelitian ini juga berfokus untuk mengembangkan AR yang interaktif agar efektif sebagai *user manual* dengan melakukan pengujian terhadap *usability*.

2.2. Landasan Teori

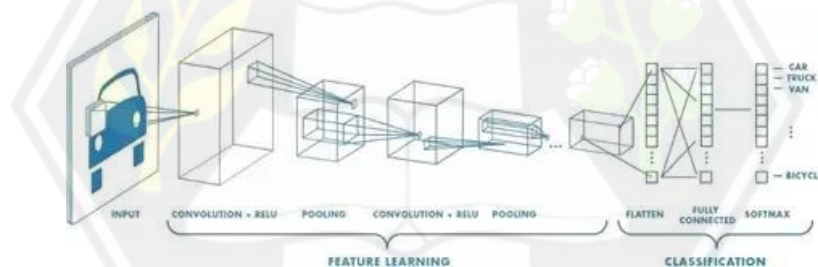
2.2.1. User manual

User manual merupakan bagian dari dokumentasi teknis suatu produk. Saat ini, format standar dari *user manual* masih berbasis buku cetak. Pada umumnya, *user manual* berisi penjelasan terkait fungsi dan tahap penggunaan produk. Selain itu, ada juga ilustrasi dalam bentuk gambar untuk menyampaikan informasi tertentu dalam bentuk visual agar efektif. Dalam beberapa kasus, *user manual* disertakan dalam bentuk dokumen digital seperti PDF melalui yang dapat akses melalui CD atau diunduh menggunakan internet (Müller dkk., 2013).

2.2.2. Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) merupakan algoritma *deep learning* yang dikembangkan dari MLP (*Multi-Layer Perceptron*) dan dirancang untuk

mengolah data dalam bentuk dua dimensi, seperti gambar atau suara. CNN menggunakan metode *supervised learning* untuk mengklasifikasikan data yang diberi *label*. *Supervised learning* bekerja dengan melatih data kemudian mengelompokkannya sesuai dengan *label* yang ada. CNN terdiri dari neuron-neuron yang diatur dalam 3 dimensi, yaitu dimensi spasial input (lebar dan tinggi) dan kedalaman. Ukuran lapisan digambarkan oleh dimensi lebar dan tinggi, sedangkan jumlah lapisan digambarkan oleh kedalaman. Secara garis besar, CNN terdiri dari 3 tipe lapisan (*layer*), yaitu *convolutional layers*, *pooling layers*, dan *fully connected layers*. *Convolutional layer* dan *pooling layer* berfungsi untuk ekstraksi fitur (*feature extraction*), sedangkan *fully connected layer* berfungsi untuk klasifikasi (*classification*). *Feature extraction layer* akan menerima input gambar kemudian diproses untuk menghasilkan output berupa vektor yang akan diolah pada lapisan berikutnya. *Classification layer* akan menerima input berupa output dari *feature extraction layer*. Input tersebut kemudian diolah dengan *hidden layer* untuk menghasilkan output berupa kelas hasil klasifikasi beserta dengan akurasi. menunjukkan arsitektur dari CNN (Ilahiyah & Nilogiri, 2018).



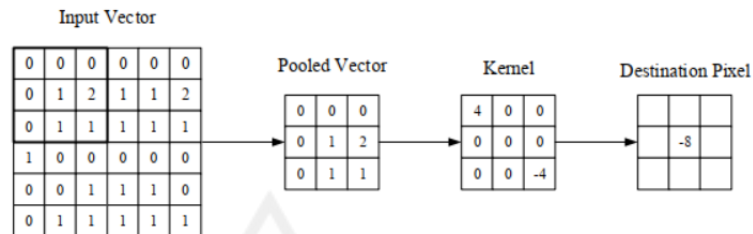
Gambar 2.1: Arsitektur Convolutional Neural Network
(Qolbiyatul Lina, 2019)

Berikut merupakan fungsi dari masing-masing *layer* yang ada pada arsitektur CNN (O'Shea & Nash, 2015):

1. *Convolution Layer*

Convolution layer berfungsi untuk menghasilkan output berupa *activation/feature maps*. Parameter pada *convolution layer* berfokus pada suatu kernel (filter). Filter ini akan bergeser pada seluruh area dari input gambar untuk menghasilkan *activation/feature maps* dalam bentuk 2D. *Activation/feature maps* menunjukkan fitur unik dari gambar

asli. Representasi visual dari *convolution layer* dapat dilihat pada Gambar 2.2.



Gambar 2.2: Representasi visual dari Convolution Layer

(O'Shea & Nash, 2015)

Convolution layer dapat mengurangi kompleksitas dari model dengan mengoptimisasi output. Hal ini dapat dilakukan dengan menggunakan 3 *hyperparameters*, yaitu *depth*, *stride*, dan *zero-padding*.

2. Pooling Layer

Pooling layer bertujuan untuk *downsampling* yaitu mengurangi dimensi dari *activation/feature maps* sehingga proses komputasi menjadi lebih cepat karena parameter yang digunakan lebih sedikit. Pada *max-pooling layer*, terdapat filter yang bergeser pada area *activation/feature maps* untuk mengambil nilai terbesar dari setiap pooling kernel menggunakan fungsi "MAX".

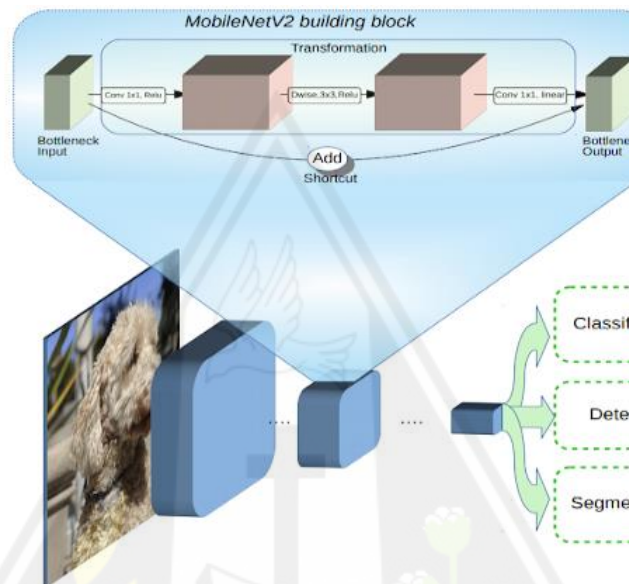
3. Fully Connected Layer

Fully connected layer bertujuan untuk klasifikasi.

2.2.3. MobileNetV2

MobileNetV2 adalah arsitektur yang dikembangkan dari MobileNetV1 untuk meningkatkan performa model dengan menggunakan modul terbaru yang disebut *inverted residual with linear bottleneck*. Modul ini cocok dirancang untuk kebutuhan *mobile* karena dapat mengurangi akses memori utama ke *embedded device* dengan memori *cache* yang kecil. Perbedaan antara MobileNetV1 dengan MobileNetV2, yaitu adanya 2 fitur baru yang ditambahkan dalam arsitektur MobileNetV2. Fitur tersebut yaitu *linear bottlenecks* antar *layer* dan *shortcut connections* antar *bottleneck*. *Bottlenecks* akan melakukan *encoding* terhadap model, sedangkan *layer* di dalamnya akan mengenkapsulasi kemampuan model

untuk berubah dari *lower-level concepts* seperti piksel menjadi *higher level descriptors* seperti kategori gambar. *Shortcut connections* antar *bottleneck* berfungsi agar proses *training* menjadi lebih cepat dan menghasilkan akurasi yang lebih baik (Sandler dkk., 2018).



Gambar 2.3: Linear Bottleneck dan Shortcut Connection dalam arsitektur MobileNetV2 (Rizqi Okta Ekoputris, 2018)

Dalam MobileNetV2, *bottleneck residual block* terbagi menjadi 2 tipe, yaitu *residual block* dengan *stride* bernilai 1 dan *residual block* dengan *stride* bernilai 2. Masing-masing *block* terdiri dari 3 layer, yaitu 1×1 convolution dengan ReLU6, *depthwise convolution*, dan 1×1 convolution tanpa ReLU. Layer dalam *residual block* dapat dilihat pada Tabel 2.1.

Tabel 2.1: *Bottleneck Residual Block* yang mengubah k menjadi k' channel dengan *stride* (s) dan *expansion factor* (t) (Sandler dkk., 2018)

Input	Operator	Output
$h \times w \times k$	1×1 conv2d, ReLU6	$h \times w \times (tk)$
$h \times w \times tk$	3×3 dwise $s=s$, ReLU6	$\frac{h}{s} \times \frac{w}{s} \times (tk)$
$\frac{h}{s} \times \frac{w}{s} \times tk$	Linear 1×1 conv2d	$\frac{h}{s} \times \frac{w}{s} \times k'$

Arsitektur pada MobileNetV2 terdiri atas 53 layer dan gambaran terkait arsitektur ini dapat dilihat pada Tabel 2.2. Dalam arsitektur ini terdapat *fully convolution layer* dengan 32 filter diikuti dengan 19 *residual bottleneck layers*.

ReLU 6 digunakan karena kemampuan yang lebih baik, meskipun komputasi rendah. Ukuran kernel yang digunakan yaitu 3×3 sebagai standar serta menggunakan *dropout* dan *batch normalization* selama *training*.

Tabel 2.2: Arsitektur MobileNetV2 (Paul-Louis Pröve, 2018)

<i>Input</i>	<i>Operator</i>	<i>t</i>	<i>c</i>	<i>n</i>	<i>s</i>
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1×1	-	1280	1	1
$7^2 \times 1280$	avgpool 7×7	-	-	1	-
$1 \times 1 \times 1280$	conv2d 1×1	-	k	-	-

2.2.4. Augmented Reality

Augmented Reality (AR) merupakan teknologi yang menghadirkan konten virtual, seperti grafik 2D, grafik 3D, suara, dan video ke dunia nyata secara *real-time*. Beberapa perangkat seperti *smartphone*, *tablet*, atau *desktop* yang telah menerapkan teknologi ini dapat membantu manusia dalam meningkatkan kapasitas indera. Banyak bidang telah menggunakan AR seperti hiburan, industri, militer, komersial, kesehatan, dan aplikasi pemasaran (de Paiva Guimarães & Martins, 2014).

Dalam menampilkan AR, perlu adanya marker. *Marker* adalah penanda yang memiliki pola tertentu dan dapat dikenali oleh kamera untuk menampilkan objek 3D. Marker pada *Augmented Reality* terbagi ke dalam dua kategori, yaitu *marker-based tracking* dan *markerless* (Wahyudi dkk., 2019). Marker dibagi menjadi 2, yaitu:

1. *Marker-based tracking*

Marker-based tracking adalah salah satu metode pada Augmented Reality yang bekerja dengan mengenali posisi dan orientasi marker untuk menciptakan dunia virtual 3D (Wahyudi dkk., 2019).

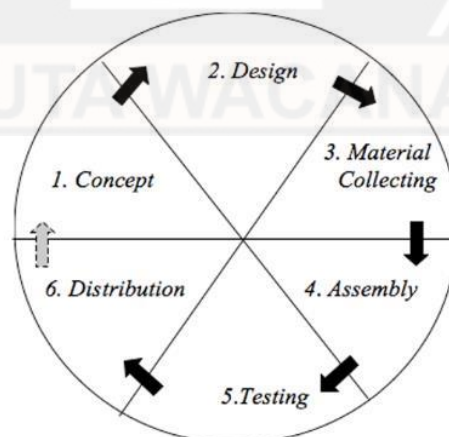
2. *Markerless*

Markerless adalah salah satu metode pada Augmented Reality tanpa menggunakan *frame marker* sebagai *tracking object*. Metode ini dapat menggantikan *marker* yang menghabiskan ruang. *Markerless* menggunakan gambar atau permukaan apapun yang berisi tulisan, logo, atau gambar sebagai *tracking object* sehingga penggunaan Augmented Reality dapat lebih interaktif dan terlihat hidup (Rizki, 2012).

Aplikasi yang dibangun pada penelitian ini menggunakan metode *marker-based tracking* untuk membedakan *user manual* yang akan ditampilkan untuk masing-masing produk.

2.2.5. Multimedia Development Life Cycle (MDLC)

Metode yang digunakan dalam mengembangkan aplikasi berbasis Augmented Reality yaitu MDLC (Multimedia Development Life Cycle). Metode yang dikembangkan oleh Luther (1994) ini terdiri atas enam tahap, yaitu *concept*, *design*, *material collecting*, *assembly*, *testing*, dan *distribution* (Octraviano Rotinsulu dkk., 2018). Gambar 2.4 menunjukkan tahapan-tahapan yang diperlukan dalam menggunakan metode MDLC.



Gambar 2.4: MDLC Luther

Selain itu, peneliti juga mengadaptasi metode *requirement engineering* yang dikembangkan oleh Börsting dkk ke dalam beberapa tahap dalam proses MDLC. Metode *software engineering* yang dikembangkan oleh Börsting, dkk bertujuan untuk membangun aplikasi berteknologi AR. Metode ini terdiri atas beberapa tahap, yaitu *requirement engineering*, *UI design*, *implementation*, dan *testing* (Börsting dkk., 2022). Menurut Binanto (2010), keenam tahap dari metode MDLC tersebut dapat dijabarkan sebagai berikut (Sugiarto, 2018):

1. *Concept*

Tahap *concept* (konsep) merupakan tahap untuk mengetahui tujuan dari pembuatan aplikasi dan siapa pengguna dari aplikasi tersebut. Pada tahap ini, peneliti mengadaptasi metode *requirement engineering* yang dikembangkan oleh Börsting, dkk. Menurut Börsting, dkk, tahap *requirement engineering* merupakan tahap untuk mengidentifikasi, meninjau, mendiskusikan, memahami, dan mendokumentasikan *requirement* sistem. Teknik untuk melakukan *requirement engineering* dibagi menjadi 3 (tiga), yaitu:

a) *Brainstorming*

Teknik *brainstorming* bertujuan untuk mendapatkan ide desain maupun *requirement* yang diperlukan dari pembuatan *software AR* dan biasanya dilakukan pada tahap awal pengembangan.

b) *Structured Workshops*

Teknik *structured workshops* bertujuan untuk menghasilkan dan mengevaluasi *requirement* yang diperlukan dalam pembuatan *software AR* secara cepat karena dilakukan dalam suatu grup partisipan.

c) *Interviews*

Teknik *interview* bertujuan untuk membuat *requirement* dalam pengembangan *software AR* menjadi lebih rinci melalui pertanyaan spesifik dan bisa dilakukan di setiap tahap.

2. *Design*

Tahap *design* (perancangan) merupakan tahap untuk membuat spesifikasi terkait arsitektur, gaya, tampilan, dan kebutuhan material. Pada tahap ini, peneliti mengadaptasi metode UI design yang dikembangkan oleh Börsting, dkk. Menurut Börsting, dkk, tahap UI design bertujuan agar interaksi antara pengguna dengan software AR menjadi lebih efektif. Teknik untuk melakukan tahap UI design dibagi menjadi 3 (tiga), yaitu:

a) *Prototyping*

Teknik *prototyping* bertujuan untuk menghasilkan *requirement* sistem, artefak *software*, dan antarmuka pengguna.

b) *Storyboarding*

Teknik *storyboarding* bertujuan untuk mengilustrasikan interaksi pengguna melalui rangkaian sketsa UI yang mewakili alur, konteks dan *environment* dari aplikasi.

3. *Material Collecting*

Tahap *material collecting* merupakan tahap untuk mengumpulkan bahan sesuai dengan kebutuhan.

4. *Assembly*

Tahap *assembly* (pembuatan) merupakan tahap untuk membuat program berdasarkan pada tahap *design*. Tahap ini dapat dikerjakan secara paralel dengan tahap sebelumnya yaitu tahap *material collecting*.

5. *Testing*

Tahap *testing* merupakan tahap untuk menjalankan aplikasi dan melihat kemungkinan terjadinya kesalahan. Pengujian dilakukan oleh pembuat atau lingkungan pembuat sehingga disebut juga sebagai tahap pengujian *alpha (alpha test)*.

6. *Distribution*

Tahap *distribution* merupakan tahap untuk menyimpan aplikasi dalam suatu media tertentu. Apabila media tersebut tidak cukup, maka aplikasi akan dikompres menjadi lebih kecil.

2.2.6. Confusion Matrix

Confusion matrix adalah tabel yang digunakan sebagai metode untuk menggambarkan pengukuran tingkat akurasi deteksi objek (Manajang dkk., 2021). *Confusion matrix* merupakan teknik untuk evaluasi hasil klasifikasi dengan menghitung ukuran statistik, yaitu *True Positive (TP)*, *True Negative (TN)*, *False Positive (FP)*, dan *False Negative (FN)* (Maria Navin & Pankaja, 2016). Berikut adalah istilah yang digunakan dalam *confusion matrix*.

1. *True Positive (TP)*: jumlah data yang menunjukkan hasil prediksi dengan kondisi aktual sama-sama bernilai positif
2. *True Negative (TN)*: jumlah data yang menunjukkan hasil prediksi dengan kondisi aktual sama-sama bernilai negatif
3. *False Positive (FP)*: jumlah data yang menunjukkan hasil prediksi bernilai positif, sedangkan kondisi aktual bernilai negatif
4. *False Negative (FN)*: jumlah data yang menunjukkan hasil prediksi bernilai negatif, sedangkan kondisi aktual bernilai positif

Tabel 2.3: Confusion Matrix

		<i>Actual Values</i>	
		<i>True</i>	<i>False</i>
<i>Prediction Values</i>	<i>True</i>	<i>TP (Correct Result)</i>	<i>FP (Unexpected Result)</i>
	<i>False</i>	<i>FN (Missing Result)</i>	<i>TN (Correct absence of result)</i>

Nilai dari akurasi (*accuracy*) dan presisi (*precision*) dapat diperoleh dari keempat klasifikasi hasil pengukuran di atas (Rahmad dkk., 2020). *Accuracy* merupakan hasil perhitungan tingkat akurasi sistem dalam mendeteksi objek. Persamaan 2.1 menunjukkan rumus untuk melakukan perhitungan *accuracy*:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.1)$$

Precision merupakan jumlah prediksi benar dibanding dengan seluruh hasil yang dideteksi oleh sistem. Persamaan 2.2 menunjukkan rumus untuk melakukan perhitungan *precision*:

$$Precision = \frac{TP}{(TP + FP)} \quad (2.2)$$

Recall adalah jumlah prediksi benar dibanding dengan seluruh hasil yang sebenarnya. Persamaan 2.3 menunjukkan rumus untuk melakukan perhitungan *recall*:

$$Recall = \frac{TP}{(TP + FN)} \quad (2.3)$$

2.2.7. Cognitive Walkthrough

Cognitive walkthrough adalah metode untuk mengevaluasi interaksi pengguna dengan suatu produk. Metode ini sangat berguna untuk memahami proses pemikiran dan pengambilan keputusan oleh pengguna pada saat berinteraksi dengan sistem, terutama untuk pengguna yang baru pertama kali atau jarang menggunakan produk tersebut (Sheikh dkk., 2021).

Ada beberapa proses yang perlu dilakukan untuk melakukan *Cognitive Walkthrough*, yaitu (Wharton dkk., 1994):

1. Menetapkan input untuk proses *walkthrough*.
 - a) Siapa penggunanya,
 - b) Contoh tugas untuk evaluasi,
 - c) Urutan tindakan untuk menyelesaikan tugas, dan
 - d) Deskripsi atau implementasi antarmuka
2. Mengumpulkan para analis
3. *Walkthrough* setiap aksi untuk menyelesaikan tugas. Proses *walkthrough* tersebut perlu dilakukan agar peneliti dapat memperoleh jawaban atas pertanyaan-pertanyaan berikut:
 - a) *Will the user try to achieve right effect?* (Akankah pengguna mencoba mencapai efek yang tepat?)
 - b) *Will the user notice that the correct action is available?* (Akankah pengguna memperhatikan bahwa)
 - c) *Will the user associate the correct action with the effect they are trying to achieve?* (Akankah pengguna mengaitkan tindakan yang benar dengan efek yang ingin mereka capai?)

d) *If the correct action is performed, will the user see that progress is being made toward solution of their task?* (Jika tindakan yang benar dilakukan, akankah pengguna melihat kemajuan yang dibuat menuju penyelesaian tugas mereka?)

4. Rekam informasi penting, seperti:

- a) Pengetahuan pengguna
- b) Asumsi dari pengguna
- c) Catatan terkait masalah dan perubahan desain
- d) Keberhasilan pengguna dalam menyelesaikan task

5. Revisi *interface* untuk memperbaiki masalah.

Cognitive walkthrough bertujuan untuk mengetahui interaksi pengguna dengan antarmuka melalui beberapa tugas yang telah ditentukan sebelumnya (Schaeffer, 2014). Fokus utama dalam metode ini yaitu seberapa baik pengguna pemula dapat menggunakan antarmuka tanpa mendapat pelatihan sebelumnya. Ada tiga cara untuk mencapai tujuan tersebut, yaitu:

1. Evaluator membandingkan tujuan pengguna dan tujuan yang diperlukan untuk mengoperasikan antarmuka pengguna,
2. Evaluator menentukan masalah yang mungkin dimiliki pengguna dalam memilih tindakan yang tepat untuk mencapai tujuan yang telah ditentukan peneliti, dan
3. Fasilitator mengevaluasi seberapa besar kemungkinan tujuan pengguna dapat berubah dengan tindakan pengguna yang benar dan respon sistem.

Peneliti mempersiapkan kertas berisi empat pertanyaan dan daftar kemungkinan aksi yang dilakukan oleh pengguna untuk menyelesaikan *task*. Selanjutnya, peneliti mencatat langkah perbaikan yang dilakukan oleh evaluator secara individu. Di akhir sesi, evaluator diminta untuk memberikan kesimpulan secara keseluruhan terkait produk yang diuji tersebut.

BAB III

METODOLOGI PENELITIAN

3.1. Concept

3.1.1. Analisis Kebutuhan Sistem

Analisis kebutuhan sistem bertujuan untuk mengetahui kebutuhan yang diperlukan dalam membangun aplikasi. Dalam membangun aplikasi, kebutuhan sistem dibagi menjadi dua bagian, yaitu kebutuhan fungsional dan non fungsional. Berikut adalah rincian dari masing-masing kebutuhan fungsional:

1. Kebutuhan Fungsional

Kebutuhan fungsional bertujuan untuk mengetahui proses-proses yang dibutuhkan pada saat aplikasi berjalan. Berikut adalah kebutuhan fungsional dari aplikasi:

- a) Aplikasi dapat melakukan proses pemindaian dan mengenali gambar dari kartu PT. Miland Cipta Usaha,
- b) Aplikasi dapat mendeteksi objek *remote control* secara *real-time* dan mengklasifikasikan objek sesuai dengan *output* yang sebenarnya, dan
- c) Aplikasi dapat menampilkan objek digital yang interaktif dan dapat digunakan untuk memberikan petunjuk terkait penggunaan produk

2. Kebutuhan Non Fungsional

Kebutuhan non fungsional dibagi menjadi dua bagian, yaitu kebutuhan terkait perangkat lunak (*software*) dan perangkat keras (*hardware*). Berikut adalah rincian dari masing-masing kebutuhan non fungsional:

a) *Hardware*:

1) Perangkat Android dengan spesifikasi sebagai berikut:

- Perangkat memiliki Google Play Store
- Perangkat memiliki versi Android 7.0 atau lebih

- Perangkat didukung oleh ARCore melalui Google Play Services for AR.

b) *Software:*

1) *Android Studio Flamingo | 2022.2.1 Patch 2*

Platform ini digunakan sebagai *Integrated Development Environment (IDE)* untuk membangun aplikasi Android dalam penelitian ini.

2) *Google Colab*

Platform ini digunakan sebagai IDE untuk melakukan augmentasi sampel dan membangun model.

3) *Anaconda version 2019.07*

Platform ini digunakan untuk menciptakan *environment* yang mempermudah proses pengolahan sampel dan dataset.

4) *Jupyter Notebook*

Platform ini digunakan sebagai IDE untuk mengolah sampel dan dataset.

c) *Python*

Ada 2 versi *Python* yang digunakan dalam penelitian ini, yaitu versi 3.7.0 dan versi 3.10.12. *Python* versi 3.7.0 digunakan di *Jupyter Notebook*, sedangkan *Python* versi 3.10.12 digunakan di *Google Colab*.

d) *Library:*

1) *Tensorflow Lite Model Maker*

Library ini digunakan untuk menyederhanakan proses pelatihan model menggunakan *custom* dataset. Proses untuk melakukan *transfer learning* menjadi lebih cepat dan mudah dengan menggunakan library ini. Selain itu, library ini juga memudahkan proses konversi model ke dalam bentuk TF Lite yang dapat langsung diintegrasikan pada aplikasi Android.

2) *Keras*

Library ini digunakan untuk mempercepat proses komputasi algoritma *deep learning*. Dalam penelitian ini, *library* ini digunakan untuk melakukan augmentasi sampel gambar.

3) *Sceneform version 1.21.0*

Library ini digunakan untuk memudahkan proses pengembangan aplikasi Android dengan teknologi AR. *Library* ini terintegrasi dengan *AR Core* sebagai Android SDK dan *Google Filament* sebagai *3D engine*. Hal ini membuat proses *rendering* objek digital pada aplikasi Android menjadi lebih mudah dan cepat.

4) *ML Kit version 17.0.0*

Library ini digunakan untuk memudahkan proses integrasi model yang sudah dibangun sebelumnya dalam bentuk TF Lite ke dalam aplikasi Android.

e) *Software Development Kit (SDK):*

1) *AR Core version 1.37.0*

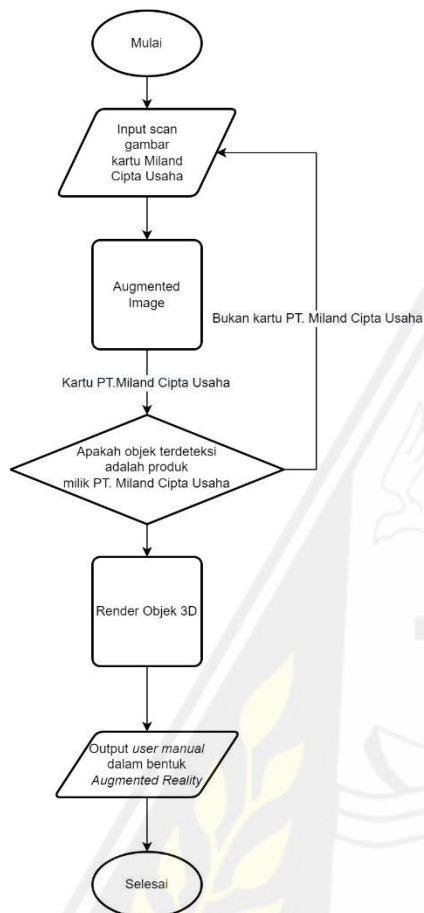
SDK ini terintegrasi dengan *library* Sceneform dan berfungsi untuk meningkatkan pengalaman AR bagi pengguna aplikasi agar lebih realistis.

f) *3D Engine*

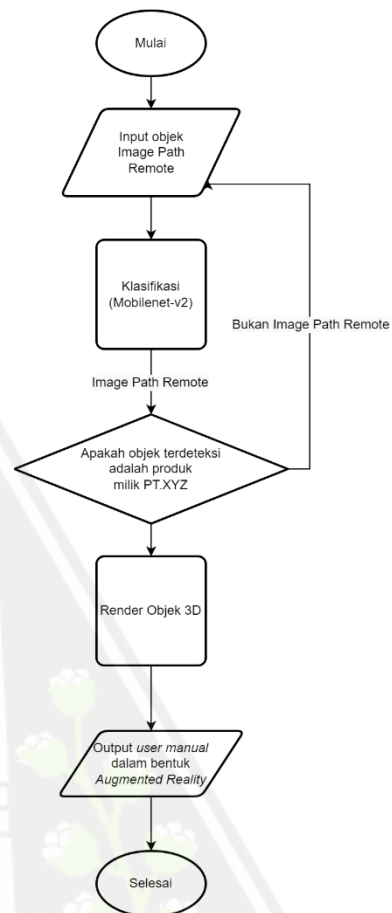
1) *Google Filament*

3D engine ini terintegrasi dengan *library* Sceneform dan digunakan untuk proses *render* objek digital.

3.1.2. Diagram Alir Aplikasi



Gambar 3.1: *Flowchart Menu Image Path Monitor*



Gambar 3.2: *Flowchart Menu Image Path Remote*

Aplikasi dibagi ke dalam dua menu utama sehingga ada dua *flowchart* yang dihasilkan pada penelitian ini. Gambar 3.1 merupakan *flowchart* untuk menu *Image Path Monitor*, sedangkan Gambar 3.2 merupakan *flowchart* untuk menu *Image Path Remote*. Masing-masing *flowchart* memiliki alur yang kurang lebih sama. Perbedaan pada alur tersebut hanya pada proses pemindaian *marker*.

Pertama, aplikasi akan menerima input berupa gambar kartu PT. Miland Cipta Usaha atau objek *Image Path Remote*. Selanjutnya, input tersebut dikenali dengan menggunakan teknologi *Augmented Image* untuk *Image Path Monitor*, sedangkan deteksi objek digunakan untuk mengenali input pada *Image Path Remote*. Apabila proses input yang dikenali sudah sesuai, maka aplikasi akan

melakukan *render* terhadap objek digital dari *Image Path Monitor* maupun *Image Path Remote*.

3.1.3. Brainstorming

Pada tahap *brainstorming*, diskusi dilakukan dengan PT. Miland Cipta Usaha beserta dengan dosen pembimbing untuk mendapatkan gambaran terkait aplikasi *user manual* yang akan dibuat dengan memanfaatkan teknologi AR. Catatan detail terkait hasil dari tahap *brainstorming* dapat dilihat pada Tabel 3.1.

Tabel 3.1: Catatan *Brainstorming*

Narasumber	:	PT. Miland Cipta Usaha
Tanggal	:	Kamis, 26 Januari 2023
Catatan		
Masalah	:	PT. Miland Cipta Usaha sebagai perusahaan yang bergerak di bidang <i>wall display</i> sering sekali mengalami kendala pada saat ingin mendemonstrasikan produk mereka kepada <i>client</i> . Kendala ini terjadi karena produk yang dihasilkan rata-rata memiliki ukuran yang besar dan berat sehingga sulit apabila ingin dibawa ke mana-mana.
Tujuan Pembuatan Aplikasi	:	Aplikasi <i>user manual</i> dengan teknologi AR yang dibuat oleh peneliti bertujuan untuk memudahkan PT. Miland Cipta Usaha dalam melakukan demonstrasi tanpa perlu membawa produk tersebut.
Sasaran	:	Pihak promotor dari PT. Miland Cipta Usaha dan masyarakat umum
Interaksi AR	:	PT. Miland Cipta Usaha ingin aplikasi dapat menampilkan menu yang ada di <i>Image Path Monitor</i> , serta <i>list port</i> dan tombol dalam bentuk AR. Selain itu, PT. Miland Cipta Usaha ingin adanya interaksi yang terjadi pada saat tombol tertentu ditekan oleh pengguna.

3.1.4. Interview Tahap 1

Interview tahap 1 ini dilakukan untuk mengumpulkan informasi terkait cara PT. Miland Cipta Usaha dalam mendemonstrasikan Image Path Monitor kepada *client*. Selain itu, *interview* ini juga dilakukan untuk mendapat gambaran dari bentuk AR yang diinginkan oleh PT. Miland Cipta Usaha. Catatan detail terkait hasil dari *interview* tahap 1 tersebut dapat dilihat pada Tabel 3.2.

Tabel 3.2: Catatan *Interview* Tahap 1

Narasumber	:	PT. Miland Cipta Usaha
Tanggal	:	Kamis, 16 Februari 2023
Catatan		
Proses Demonstrasi Produk	:	<ul style="list-style-type: none">• Demonstrasi diawali dengan pengenalan terkait port dan tombol yang ada pada <i>monitor</i>, kemudian dilanjutkan dengan pengenalan terkait fitur-fitur <i>Image Path Monitor</i>, seperti S-Write, <i>File Sharing</i>, dll• AR akan menampilkan bentuk virtual dari <i>monitor</i> lengkap dengan port dan tombol yang ada• AR akan menampilkan teks berisi informasi terkait port dan tombol yang ada• AR akan menampilkan video terkait petunjuk penggunaan fitur yang dipilih oleh pengguna. Video tersebut akan disediakan oleh PT. Miland Cipta Usaha

3.1.5. Interview Tahap 2

Interview tahap 2 ini dilakukan untuk mengetahui gambaran dari aplikasi yang diperlukan oleh PT. Miland Cipta Usaha. Gambaran aplikasi yang dimaksud meliputi, tema warna, alur dan referensi desain aplikasi, serta informasi terkait *user manual* produk. Catatan detail terkait hasil dari *interview* tahap 2 tersebut dapat dilihat pada Tabel 3.3.

Tabel 3.3: Catatan *Interview* Tahap 2

Narasumber	:	PT. Miland Cipta Usaha
Tanggal	:	Kamis, 23 Februari 2023
Catatan		
Tema Warna Aplikasi	:	Tema warna aplikasi yaitu biru
Alur Aplikasi	:	Aplikasi akan menampilkan <i>splash screen</i> , kemudian muncul petunjuk penggunaan aplikasi. Setelah itu, aplikasi akan menampilkan kamera yang berfungsi untuk pemindaian gambar/objek kemudian menampilkan AR dari monitor
Interaksi AR	:	Aplikasi akan menampilkan tombol yang menggambarkan port maupun fitur dari monitor. Pada saat tombol port ditekan akan muncul deskripsi port, sedangkan pada saat tombol fitur ditekan akan muncul video yang menjelaskan fitur tersebut
<i>User Manual Image Path Monitor</i>	:	PT. Miland Cipta Usaha akan mengunggah dokumen berisi rincian deskripsi terkait masing-masing port dan fitur yang ada pada Image Path Monitor dalam bentuk PDF di Google Drive

3.1.6. *Interview* Tahap 3

Interview tahap 3 ini dilakuakn untuk mendapatkan informasi lebih detail terkait *user manual* dari remote. Catatan detail terkait hasil dari *interview* tahap 3 tersebut dapat dilihat pada Tabel 3.4.

Tabel 3.4: Catatan *Interview* Tahap 3

Peneliti	:	Albertus Adrian Susanto
Narasumber	:	PT. Miland Cipta Usaha
Tanggal	:	Senin, 24 Maret 2023
Catatan		

<i>User Manual Image Path Remote</i>	:	PT. Miland Cipta Usaha mengirim dokumen berisi rincian deskripsi terkait masing-masing tombol yang ada pada remote dalam bentuk PDF melalui Whatsapp
--------------------------------------	---	--

3.1.7. Structured Workshop Tahap 1

Structured workshop tahap 1 ini bertujuan untuk memberikan gambaran dari aplikasi yang akan dibuat nantinya kepada PT. Miland Cipta Usaha. *Workshop* dilakukan melalui Whatsapp diawali dengan peneliti mengirimkan video *prototype* aplikasi yang telah dibuat dengan menggunakan Figma. Selain itu, peneliti juga meminta *user manual* berisi keterangan terkait masing-masing port dan tombol kepada PT. Miland Cipta Usaha. Catatan detail terkait hasil dari *structured workshop* tahap 1 tersebut dapat dilihat pada Tabel 3.5.

Tabel 3.5: Catatan *Structured Workshop* Tahap 1

Peserta	:	PT. Miland Cipta Usaha
Tanggal	:	Selasa, 28 Februari 2023
Catatan		
<ul style="list-style-type: none"> • PT. Miland Cipta Usaha telah sepakat dengan alur dari aplikasi yang ditawarkan oleh peneliti. Berikut adalah alur jalannya aplikasi yang ada pada video <i>prototype</i> tersebut: <i>Splash screen</i> → <i>Onboarding</i> → Petunjuk Penggunaan → <i>Scan Gambar</i> → Interaksi dengan objek digital • PT. Miland Cipta Usaha sepakat dengan warna serta <i>asset</i> yang ditawarkan oleh peneliti dalam <i>prototype</i> aplikasi tersebut • PT. Miland Cipta Usaha mengirimkan <i>user manual Image Path Monitor</i> kepada peneliti melalui Whatsapp 		

3.1.8. Structured Workshop Tahap 2

Structured workshop tahap 2 ini bertujuan untuk melakukan demonstrasi terkait aplikasi AR yang sudah dibuat sampai dengan tanggal 16 Maret 2023. Demo aplikasi dilakukan secara online melalui Zoom. Dalam workshop tersebut, peneliti

fokus untuk mendemonstrasikan proses pemindaian gambar dan interaksi AR yang terjadi pada objek digital. Detail terkait interaksi AR yang ada pada video tersebut dapat dilihat pada Tabel 3.6.

Tabel 3.6: Rincian Interaksi AR

Bentuk AR	Deskripsi
<i>3D Monitor</i>	Muncul <i>3D monitor</i> yang menyerupai <i>Image Path Monitor</i> , mulai dari layar, <i>frame</i> , tombol, hingga port
Tombol <i>Power</i>	Muncul tombol power. Apabila tombol ditekan akan muncul layar beranda dari ICT
Tombol Port	Muncul tombol di masing-masing port. Apabila tombol ditekan akan muncul deskripsi terkait port tersebut
Tombol <i>Keyfeatures</i>	Muncul tombol <i>keyfeatures</i> . Apabila tombol ditekan akan muncul deskripsi fitur utama dari ICT
Tombol <i>More</i>	Muncul tombol <i>more</i> . Apabila tombol ditekan belum terjadi interaksi.
Tombol Whiteboard	Muncul tombol <i>whiteboard</i> . Apabila tombol ditekan akan tampil video berisi penjelasan terkait fitur S-Write
Tombol <i>Folder</i>	Muncul tombol <i>folder</i> . Apabila tombol ditekan belum terjadi interaksi.
Tombol <i>Sharing File</i>	Muncul tombol <i>sharing file</i> . Apabila tombol ditekan akan tampil video berisi penjelasan terkait fitur <i>sharing file</i> tanpa kabel.

Catatan detail terkait hasil dari *structured workshop* tahap 2 tersebut dapat dilihat pada Tabel 3.7.

Tabel 3.7: Catatan *Structured Workshop* Tahap 2

Narasumber	:	PT. Miland Cipta Usaha
Tanggal	:	Kamis, 16 Maret 2023
Catatan		

- Pada tahap ini, peneliti sudah berhasil merealisasikan prototype Figma yang telah dibuat sebelumnya ke dalam aplikasi.
- PT. Miland Cipta Usaha telah sepakat dengan bentuk dan interaksi dari AR user manual Interactive Collaboration Tool (ICT) yang ditawarkan oleh peneliti.

3.1.9. Structured Workshop Tahap 3

Structured workshop tahap 3 ini bertujuan untuk melakukan demonstrasi terkait aplikasi AR yang sudah dibuat sampai dengan tanggal 30 Maret 2023. Demo aplikasi dilakukan secara online melalui Zoom. Catatan detail terkait hasil dari *structured workshop* tahap 3 tersebut dapat dilihat pada Tabel 3.8.

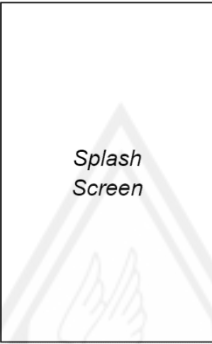
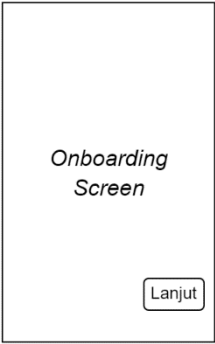
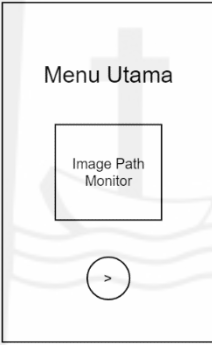
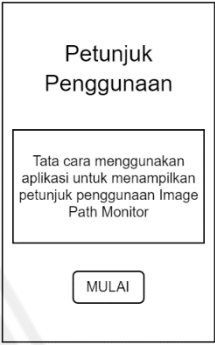


Tabel 3.8. Catatan *Structured Workshop* Tahap 3

Narasumber	:	PT. Miland Cipta Usaha
Tanggal	:	Kamis, 30 Maret 2023
Catatan		
<ul style="list-style-type: none"> • Pada tahap ini, peneliti sudah berhasil merealisasikan perubahan terhadap desain UI dari aplikasi karena adanya penambahan fitur untuk menampilkan user manual remote. Penambahan fitur ini memberikan beberapa perubahan juga terhadap UI dari aplikasi sehingga perlu didemonstrasikan kepada PT. Miland Cipta Usaha. • Selain itu, peneliti juga mendemonstrasikan proses scan objek secara <i>real time</i> dengan memanfaatkan model yang telah dibuat sebelumnya. Apabila objek sudah sesuai, maka aplikasi akan menampilkan objek AR. Pada tahap ini, model dan objek AR masih berupa data <i>dummy</i>. 		

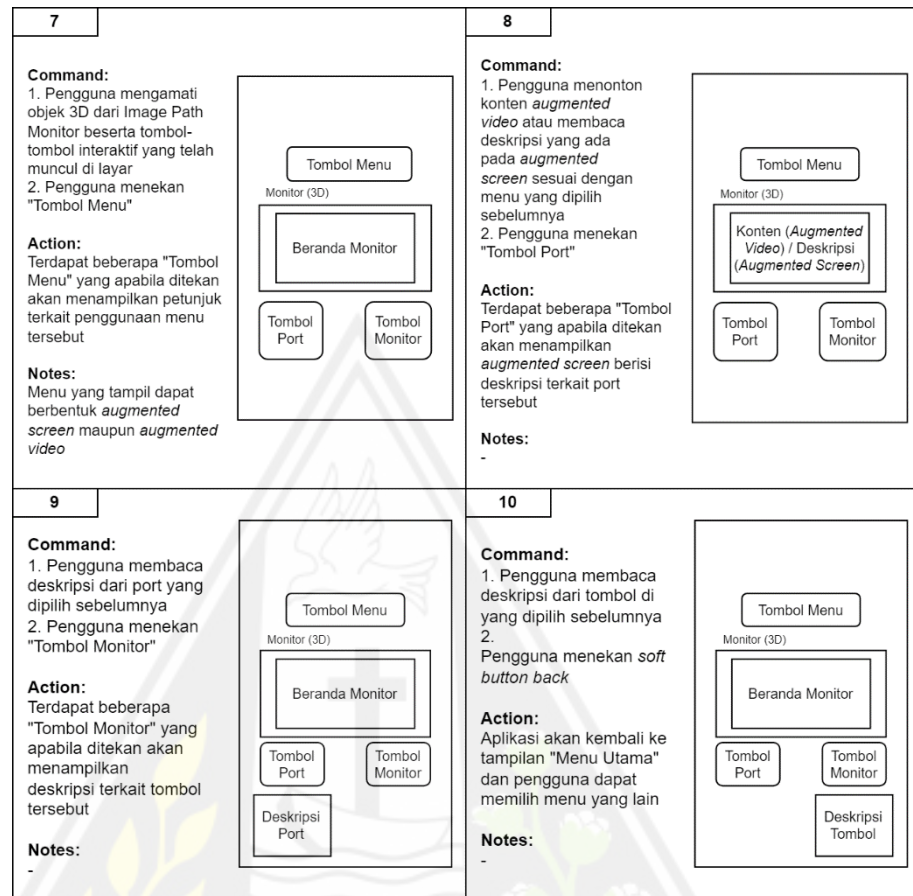
3.2. Design

3.2.1. Storyboarding

1. Storyboarding Image Path Monitor

EXPERIMENTAL STORYBOARD - IMAGE PATH MONITOR	
Title	Miland AR Detail
Author	Albertus Adrian Susanto
1	2
<p>Command: -</p> <p>Action: Pegguna menunggu selama 3 detik</p> <p>Notes: Splash Screen melakukan verifikasi semua <i>resource</i> yang digunakan aplikasi</p> 	<p>Command: Pegguna menekan tombol "Lanjut"</p> <p>Action: Aplikasi mengarah ke halaman <i>onboarding screen</i> berikutnya</p> <p>Notes: <i>Onboarding screen</i> hanya akan muncul saat aplikasi dibuka pertama kali. Halaman ini akan menampilkan deskripsi singkat terkait aplikasi.</p> 
3	4
<p>Command: 1. Pengguna memilih menu "Image Path Monitor" 2. Pengguna menekan tombol ">"</p> <p>Action: Aplikasi mengarah ke halaman berisi petunjuk penggunaan Image Path Monitor</p> <p>Notes: Tombol ">" untuk lanjut ke halaman berikutnya</p> 	<p>Command: Pegguna menekan tombol "MULAI"</p> <p>Action: Aplikasi mengarah ke halaman selanjutnya untuk memulai proses pemindaian</p> <p>Notes: Proses inialisasi AR dan memuat <i>resource</i></p> 
5	6
<p>Command: Pegguna mengarahkan kamera ke kartu PT. Miland Cipta Usaha</p> <p>Action: Aplikasi melakukan pemindaian terhadap input gambar yang diterima kamera. Pengguna menunggu proses deteksi gambar bekerja untuk menempatkan objek digital dari Image Path Monitor</p> <p>Notes: Deteksi hanya dilakukan sekali dan kamera akan terus melakukan <i>tracking</i> terhadap gambar yang berhasil dikenali oleh aplikasi</p> 	<p>Command: Pegguna menekan <i>pop up/prompt</i> berisi petunjuk</p> <p>Action: Aplikasi akan mengarah ke <i>prompt</i> berisi petunjuk berikutnya. Apabila seluruh petunjuk sudah muncul, maka pengguna dapat mulai berinteraksi dengan objek digital yang telah tersedia</p> <p>Notes: -</p> 


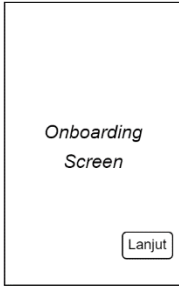
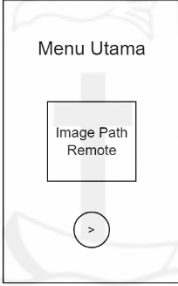
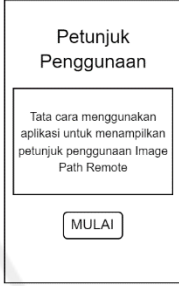


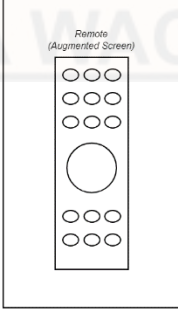
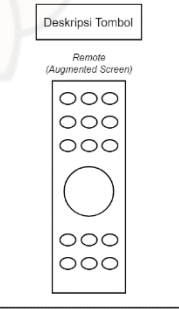
Gambar 3.3: Storyboarding Image Path Monitor



Gambar 3.4: Lanjutan *Storyboarding Image Path Monitor*

Gambar 3.3 dan Gambar 3.4 merupakan *storyboarding* dari menu *Image Path Monitor* yang akan dikembangkan dalam penelitian ini. *Storyboarding* ini memberikan gambaran terkait alur penggunaan aplikasi untuk menampilkan *user manual* dari *Image Path Monitor*.

2. Storyboarding Image Path Remote

EXPERIMENTAL STORYBOARD - IMAGE PATH REMOTE	
Title	Miland AR Detail
Author	Albertus Adrian Susanto
1	2
<p>Command: -</p> <p>Action: Pengguna menunggu selama 3 detik</p> <p>Notes: Splash Screen melakukan verifikasi semua resource yang digunakan aplikasi</p> 	<p>Command: Pengguna menekan tombol "Lanjut"</p> <p>Action: Aplikasi mengarah ke halaman <i>onboarding screen</i> berikutnya</p> <p>Notes: <i>Onboarding screen</i> hanya akan muncul saat aplikasi dibuka pertama kali. Halaman ini akan menampilkan deskripsi singkat terkait aplikasi.</p> 
3	4
<p>Command: Pengguna memilih menu "Image Path Remote" 2. Pengguna menekan tombol ">"</p> <p>Action: Aplikasi membuka tampilan petunjuk penggunaan Image Path Remote</p> <p>Notes: Tombol ">" untuk lanjut ke halaman berikutnya</p> 	<p>Command: Pengguna menekan tombol "MULAI"</p> <p>Action: Aplikasi mengarah ke halaman selanjutnya untuk memulai proses pemindaian</p> <p>Notes: Proses inisialisasi AR dan memuat <i>resource</i></p> 
5	6
<p>Command: Pengguna mengarahkan kamera ke objek Image Path Remote</p> <p>Action: Aplikasi melakukan pemindaian terhadap input objek yang diterima kamera. Pengguna menunggu proses deteksi objek bekerja untuk menempatkan objek digital dari Image Path Remote</p> <p>Notes: Deteksi hanya dilakukan sekali dan melakukan tracking terhadap objek yang berhasil dikenali oleh aplikasi</p> 	<p>Command: Pengguna menekan <i>pop up/prompt</i> petunjuk</p> <p>Action: Aplikasi akan mengarahkan pengguna ke <i>prompt</i> petunjuk berikutnya. Apabila seluruh petunjuk sudah muncul, maka pengguna dapat mulai berinteraksi dengan objek digital yang telah tersedia.</p> <p>Notes: -</p> 
7	8
<p>Command: 1. Pengguna mengamati <i>augmented screen</i> berbentuk Image Path Remote beserta dengan tombol-tombol interaktif di dalamnya 2. Pengguna berinteraksi dengan menekan tombol-tombol tersebut</p> <p>Action: Terdapat beberapa tombol pada <i>remote</i> yang apabila ditekan akan menampilkan deskripsi rinci terkait tombol tersebut</p> <p>Notes: -</p> 	<p>Command: 1. Pengguna membaca deskripsi rinci terkait tombol yang dipilih sebelumnya 2. Pengguna menekan <i>soft button back</i></p> <p>Action: Aplikasi akan kembali ke tampilan "Menu Utama" dan pengguna dapat memilih menu yang lain</p> <p>Notes: -</p> 

Gambar 3.5: Storyboarding Image Path Remote

Gambar 3.5 merupakan *storyboarding* dari menu *Image Path Remote* yang akan dikembangkan dalam penelitian ini. *Storyboarding* ini memberikan gambaran terkait alur penggunaan aplikasi untuk menampilkan *user manual* dari *Image Path Remote*.

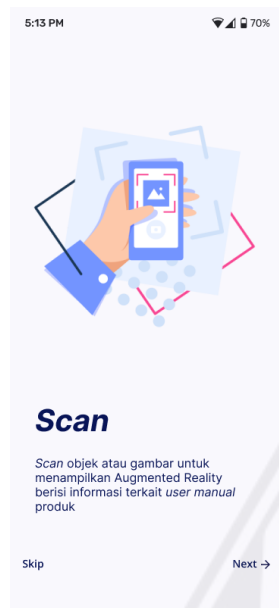
3.2.2. Prototyping

Sebelum peneliti membangun aplikasi, *prototype* dari aplikasi perlu dibuat dengan menggunakan *software* Figma. *Prototype* ini dibuat agar PT. Miland Cipta Usaha mendapat gambaran dari aplikasi yang akan dibuat dan interaksi-interaksi yang mungkin terjadi.

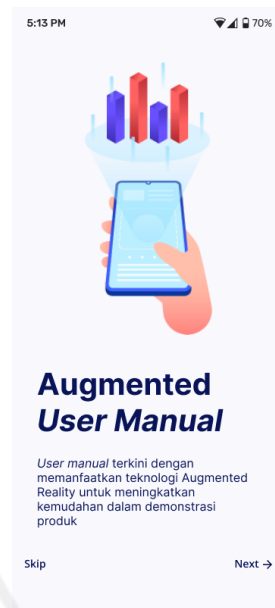


Gambar 3.6: *Prototype* Halaman *Splash Screen*

Gambar 3.6 merupakan *prototype* dari halaman *splash screen*. Halaman ini akan muncul pada saat pengguna membuka aplikasi.



Gambar 3.7: *Prototype Halaman Onboarding 1*



Gambar 3.8: *Prototype Halaman Onboarding 2*



Gambar 3.9: *Prototype Halaman Onboarding 3*

Gambar 3.7, Gambar 3.8, dan Gambar 3.9 merupakan *prototype* dari halaman *onboarding*. Halaman ini berisi deskripsi singkat terkait aplikasi.



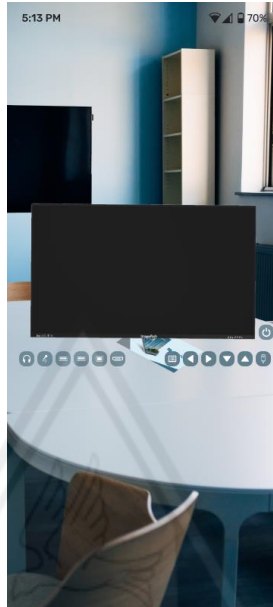
Gambar 3.10: *Prototype* Halaman Utama

Gambar 3.10 merupakan *prototype* dari halaman utama. Pada halaman ini pengguna disediakan dua opsi menu yang dapat dipilih sesuai kebutuhan.



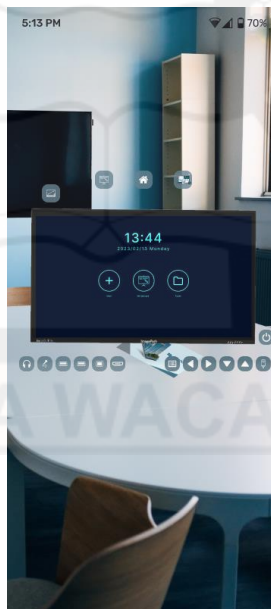
Gambar 3.11: *Prototype* Halaman *Scan Marker* untuk Menu *Image Path Monitor*

Gambar 3.11 merupakan *prototype* dari halaman *scan marker* untuk menu *Image Path Monitor*. *Prototype* ini memberikan gambaran terkait proses pemindaian gambar yang perlu dilakukan dengan menggunakan kamera untuk menampilkan *user manual* dari *Image Path Monitor*.



Gambar 3.12: *Prototype* Objek Digital *Image Path Monitor*

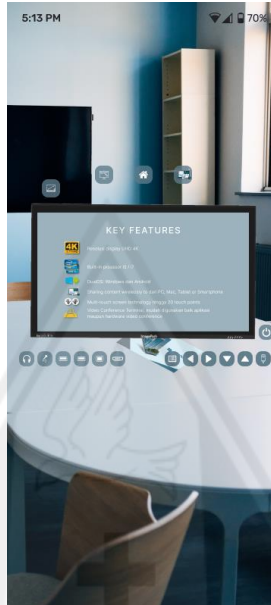
Gambar 3.12 merupakan *prototype* dari objek digital *Image Path Monitor* yang akan muncul setelah proses pemindaian berhasil. Pada *prototype* tersebut, ada beberapa objek digital yang akan tampil, seperti 3D monitor, tombol-tombol *port*, dan tombol *power*.



Gambar 3.13: *Prototype* Objek Digital Menu pada *Image Path Monitor*

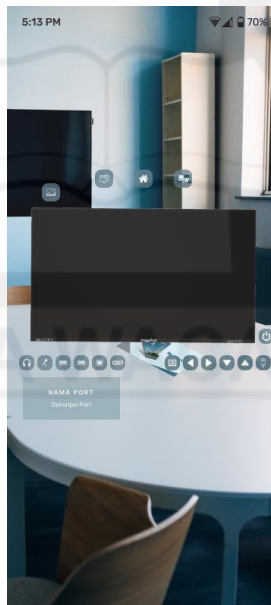
Gambar 3.13 merupakan *prototype* dari objek digital yang akan muncul setelah pengguna menekan tombol *power*. Pada *prototype* tersebut, ada beberapa

objek digital yang akan tampil, seperti halaman beranda di 3D monitor dan tombol-tombol menu.



Gambar 3.14: *Prototype* Objek Digital Fitur Utama pada *Image Path Monitor*

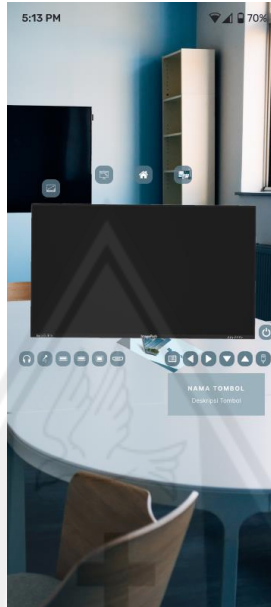
Gambar 3.14 merupakan *prototype* dari objek digital yang akan muncul setelah pengguna menekan tombol *keyfeatures*. Pada *prototype* tersebut, ada objek digital yang menampilkan deskripsi terkait fitur utama dari *Image Path Monitor*.



Gambar 3.15: *Prototype* Objek Digital Papan Petunjuk Port pada *Image Path Monitor*

Gambar 3.15 merupakan *prototype* dari objek digital yang akan muncul setelah pengguna menekan tombol yang ada di sisi kiri bawah monitor. Pada

prototype tersebut, ada objek digital yang menampilkan informasi terkait nama dan deskripsi port yang ada di *Image Path Monitor*.



Gambar 3.16: *Prototype* Objek Digital Papan Petunjuk Tombol pada *Image Path Monitor*

Gambar 3.16 merupakan *prototype* dari objek digital yang akan muncul setelah pengguna menekan tombol yang ada di sisi kanan bawah monitor. Pada *prototype* tersebut, ada objek digital yang menampilkan informasi terkait nama dan deskripsi tombol yang ada di *Image Path Monitor*.



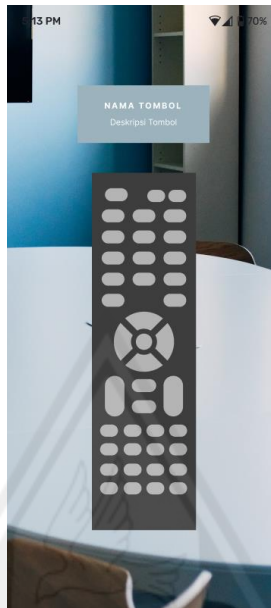
Gambar 3.17: *Prototype* Halaman *Scan Marker* untuk Menu *Image Path Monitor*

Gambar 3.17 merupakan *prototype* dari halaman *scan marker* pada menu *Image Path Remote*. *Prototype* ini memberikan gambaran terkait proses pemindaian gambar yang perlu dilakukan dengan menggunakan kamera untuk menampilkan *user manual* dari *Image Path Remote*.



Gambar 3.18: *Prototype* Objek Digital *Image Path Remote*

Gambar 3.18 merupakan *prototype* dari objek digital *Image Path Remote* yang akan muncul setelah proses pemindaian berhasil. Pada *prototype* tersebut, objek digital yang muncul berupa layout 2D dari monitor beserta dengan tombol-tombol di dalamnya.



Gambar 3.19: *Prototype* Objek Digital Papan Petunjuk Tombol pada *Image Path Remote*

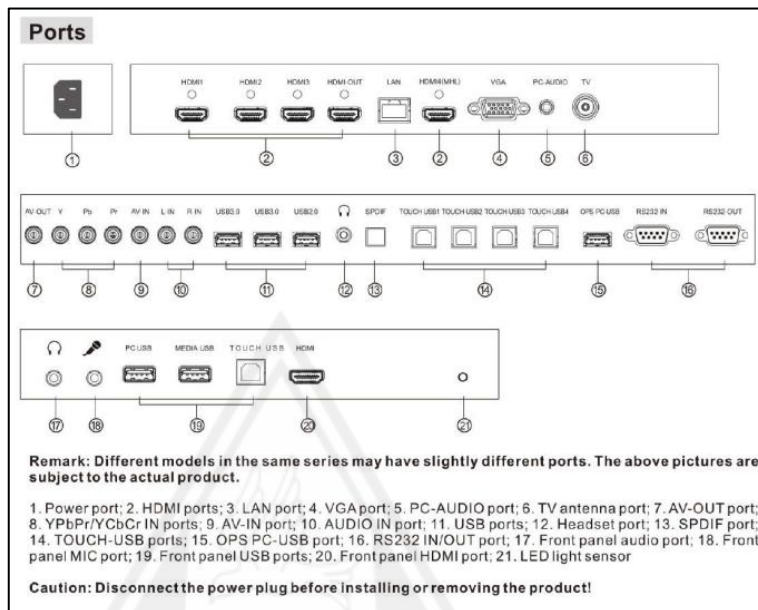
Gambar 3.19 merupakan *prototype* dari objek digital yang akan muncul setelah pengguna menekan tombol pada remote. Pada *prototype* tersebut, ada objek digital yang menampilkan informasi terkait nama dan deskripsi tombol yang ada di *Image Path Remote*.

3.3. Material Collecting

3.3.1. User Manual berbasis Textbook

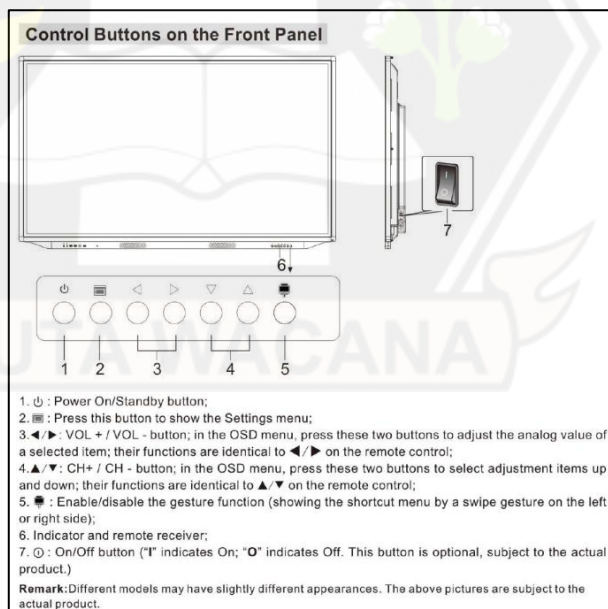
Sebelum membangun aplikasi, petunjuk terkait penggunaan *Interactive Collaboration Tool* yang ada pada buku manual perlu dipelajari terlebih dahulu. Dalam buku manual tersebut, ada penjelasan lengkap terkait *Image Path Monitor* dan *Image Path Remote*.

1. *User Manual Image Path Monitor*



Gambar 3.20: Petunjuk Port *Image Path Monitor*

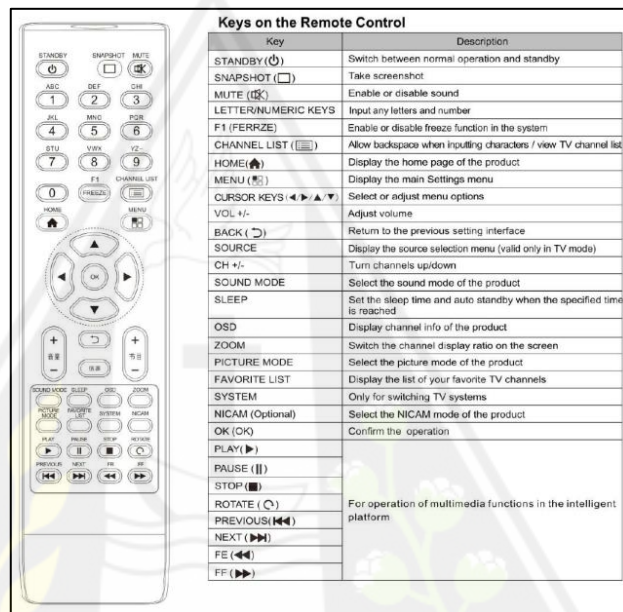
Gambar 3.20 merupakan petunjuk terkait port-port yang ada di *Image Path Monitor*. Dalam penelitian ini, aplikasi yang dibangun hanya akan menampilkan petunjuk terkait *front panel Image Path Monitor*. Oleh karena itu, penelitian ini hanya akan mengambil petunjuk yang dengan nomor 17, 18, 19, dan 20.



Gambar 3.21. Petunjuk Tombol *Image Path Remote*

Gambar 3.21 merupakan petunjuk terkait tombol-tombol yang ada di *Image Path Monitor*. Dalam penelitian ini, aplikasi yang dibangun hanya akan menampilkan petunjuk terkait *front panel Image Path Monitor*. Oleh karena itu, penelitian ini hanya akan mengambil petunjuk yang dengan nomor 1, 2, 3, 4, dan 5.

2. User Manual Image Path Remote



Gambar 3.22: Petunjuk *Image Path Remote*

Gambar 3.22 merupakan petunjuk terkait tombol-tombol yang ada di *Image Path Remote*. Dalam penelitian ini, aplikasi yang dibangun akan menampilkan seluruh petunjuk terkait masing-masing tombol yang ada pada *remote* tersebut.

3.3.2. Material untuk Objek Digital

Dalam membangun objek digital untuk *Image Path Monitor* dan *Image Path Remote*, ikon-ikon perlu dikumpulkan untuk membantu pengguna dalam mengenali tombol maupun port yang ada pada objek digital tersebut. Ikon-ikon tersebut kemudian diolah dengan menggunakan *software* Adobe Photoshop untuk mengubah ukuran, rotasi, maupun warna sesuai kebutuhan. Ikon tersebut diperoleh dari berbagai macam sumber, seperti flaticon.com dan feathericons.com.

3.3.3. Video Demonstrasi

Video demonstrasi diperlukan untuk memudahkan pengguna dalam memahami petunjuk penggunaan produk yang memerlukan banyak interaksi, salah satunya yaitu menu untuk *whiteboard* dan *sharing file* tanpa kabel. Oleh karena itu, PT. Miland Cipta Usaha memberikan alternatif untuk mengambil video demonstrasi dari *channel* youtube MFI Review yang dikelola perusahaan tersebut.

Tabel 3.9: Sumber Referensi Video

Nama <i>Channel</i>	: MFI Review
Judul Video	: ImagePathICT (Interactive Collaboration Tool)
Link	: https://youtu.be/8oEujVnDd7k

Tabel 3.9 merupakan detail terkait referensi video yang digunakan peneliti dalam membangun aplikasi. Peneliti mengunduh video tersebut, kemudian mengeditnya menggunakan *software* Adobe Premiere Pro sesuai kebutuhan.

3.3.4. Sampel *Remote Control*

Dalam membangun model untuk mengenali objek *Image Path Remote*, sampel data perlu dikumpulkan dan diolah menjadi dataset. Sampel yang digunakan dalam penelitian ini berupa video dengan format MP4 dan gambar dengan format JPG. Selanjutnya, sampel tersebut diolah menjadi dataset yang terbagi dalam dua kategori, yaitu dataset untuk *Image Path Remote* dan dataset untuk *remote* dari produk lain.

Tabel 3.10: Referensi Program untuk Mengolah Sampel

<i>Author</i>	: theAIGuysCode
<i>Repository</i>	: yolov4-custom-functions
Link	: https://github.com/theAIGuysCode/yolov4-custom-functions

Program berbasis console yang dibuat oleh theAIGuysCode tersebut digunakan untuk mengolah sampel video dan gambar menjadi dataset. Detail terkait referensi program yang digunakan untuk mengolah sampel dapat dilihat pada Tabel 3.10.

1. Sampel *Image Path Remote*

Sampel *Image Path Remote* dikumpulkan dengan mengambil gambar dan video menggunakan *smartphone*.

a) *Video Image Path Remote*

Video *remote* diambil dengan memutar *smartphone* sejauh 360° untuk mendapatkan objek *remote* dari berbagai sudut rotasi. Selain itu, perlu adanya video dengan *background* dan posisi yang berbeda untuk mendapatkan sampel yang variatif dari *remote*.



Gambar 3.23: *Video Image Path Remote*

Pada Gambar 3.3, ada tiga video *remote* dari *Image Path Remote* yang sudah dikumpulkan dalam satu folder untuk diolah lebih lanjut. Video tersebut diolah dengan mengekstrak *frame* dari video menjadi gambar dan melakukan *crop* terhadap objek *remote* yang ada pada *frame* tersebut menggunakan kode program yang peneliti peroleh dari Github theAIGuysCode.



Gambar 3.24. Hasil Pengolahan Sampel Video menjadi Gambar

Gambar 3.24 merupakan contoh hasil *frame* yang berhasil diekstrak dan di-*crop* oleh program dari theAIGuysCode. Berdasarkan tiga video tersebut, peneliti berhasil memperoleh 56 sampel gambar.

b) *Gambar Remote*

Selain itu, gambar *remote* dari berbagai sudut rotasi juga perlu diambil untuk mendapatkan sampel yang variatif dari *remote*.



Gambar 3.25: Contoh sampel *Image Path Remote* sebelum Proses *Cropping*

Gambar 3.25 merupakan contoh sampel *Image Path Remote* sebelum diolah menjadi dataset. Pada sampel tersebut, gambar masih disertai dengan *background* dari meja. Selanjutnya, peneliti mengolah sampel tersebut dengan menggunakan kode program yang peneliti peroleh dari github theAIGuysCode.



Gambar 3.26: Contoh sampel *Image Path Remote* setelah Proses *Cropping*

Gambar 3.26 merupakan contoh sampel *Image Path Remote* setelah diolah menjadi dataset. Pada sampel tersebut, gambar hanya berfokus pada objek *remote* dan siap untuk digunakan sebagai dataset.

2. Sampel *Remote* dari Produk Lain

Tabel 3.11: Referensi Sampel Data Sekunder

Nama Website	: images.cv
Link	: https://images.cv/dataset/remote-control-image-classification-dataset

Sampel remote dari produk lain dikumpulkan dengan mengunduh dataset *remote control* yang ada di internet. Tabel 3.11 merupakan detail referensi terkait dataset *remote control* tersebut.



Gambar 3.27: Contoh Sampel Remote dari Data Sekunder 1



Gambar 3.28: Contoh Sampel Remote dari Data Sekunder 2



Gambar 3.29: Contoh Sampel Remote dari Data Sekunder 3

Gambar 3.27, Gambar 3.28, dan Gambar 3.29 merupakan salah satu contoh sampel remote dari website *images.cv*. Dataset tersebut terdiri atas 1.849 gambar dan tergolong dalam data sekunder. Dataset ini akan digunakan sebagai sampel remote dari produk lain. Selain itu, sampel tersebut akan ditambah dengan data primer yang diambil dari berbagai sudut rotasi.

BAB IV

IMPLEMENTASI DAN PEMBAHASAN

4.1. Assembly

Tahap ini akan memberikan gambaran terkait proses pembuatan model menggunakan arsitektur MobileNetV2. Selain itu, tahap ini juga memberikan gambaran mengenai proses pembuatan aplikasi, mulai dari tahap pembuatan objek digital hingga implementasi nyata dari *prototype* yang telah dibuat sebelumnya.

4.1.1. Pembuatan Model untuk Detektor Remote

Model untuk deteksi *Image Path Remote* dibuat dengan menggunakan Jupyter Notebook dan Google Colab sebagai IDE. Secara garis besar, berikut tahap yang perlu dilakukan untuk membangun model tersebut:

1. Augmentasi Sampel

Berdasarkan sampel yang ada, telah terjadi ketimpangan antara jumlah sampel untuk *Image Path Remote* dengan jumlah sampel untuk *remote* dari produk lain. Oleh karena itu, augmentasi terhadap sampel *Image Path Remote* perlu dilakukan untuk mengatasi masalah tersebut.

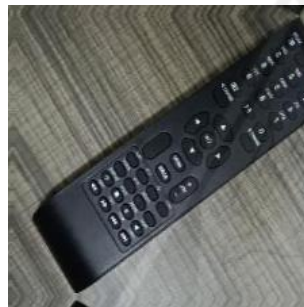
Program untuk melakukan proses augmentasi tersebut dibuat menggunakan Google Colab dengan memanfaatkan sebuah API yaitu Keras. API ini berfungsi untuk mempermudah proses augmentasi gambar. *Image Data Generator* merupakan salah satu *class* yang digunakan dalam membangun program tersebut.

Tabel 4.1: Parameter untuk Augmentasi Gambar

<i>Parameter</i>	<i>Value</i>
<i>rotation_range</i>	45
<i>width_shift_range</i>	0.2
<i>height_shift_range</i>	0.2
<i>shear_range</i>	0.2
<i>zoom_range</i>	0.2

<i>horizontal_flip</i>	True
<i>fill_mode</i>	Wrap
<i>cval</i>	125

Tabel 4.1 merupakan parameter yang digunakan untuk melakukan augmentasi sampel gambar *Image Path Remote* dalam penelitian ini.



Gambar 4.1: Contoh Hasil Augmentasi 1



Gambar 4.2: Contoh Hasil Augmentasi 2

1



Gambar 4.3: Contoh Hasil Augmentasi 3

Gambar 4.1, Gambar 4.2, dan Gambar 4.3 merupakan contoh hasil dari augmentasi sampel gambar remote dari produk PT. Miland Cipta Usaha. Proses augmentasi tersebut menghasilkan 1.556 sampel baru.

2. Pembuatan Dataset

Sampel yang sudah dikumpulkan sebelumnya pada tahap *material collecting* perlu diolah dan dikelompokkan untuk menghasilkan dataset remote. Pengolahan sampel diawali dengan memanipulasi ukuran gambar menggunakan program Python yang telah dibuat menggunakan Jupyter Notebook dengan memanfaatkan library PIL dari Python. Manipulasi ini perlu dilakukan agar ukuran gambar menyerupai

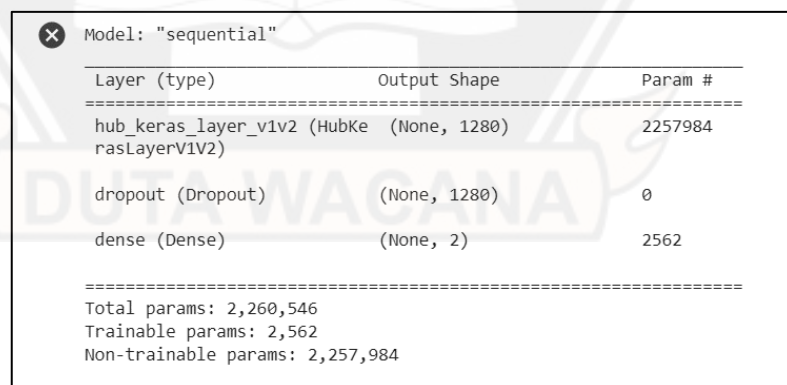
ukuran input yang dapat diterima oleh MobileNetV2 dan meminimalisir ukuran file dari dataset.

Selanjutnya, setiap sampel tersebut dikelompokkan dengan menggunakan program Python yang telah dibuat menggunakan Jupyter Notebook. Dalam hal ini, beberapa *module* dari Python, seperti Shutil, OS, dan Glob digunakan untuk memudahkan proses pengelompokan sampel ke dalam folder yang ingin dijadikan sebagai dataset.

Berdasarkan hasil pengolahan dan pengelompokan sampel-sampel tersebut. Dataset untuk *Image Path Remote* terdiri atas 1.679 file, sedangkan dataset untuk *remote* dari produk lain terdiri atas 1.886 file. Kedua dataset tersebut memiliki ukuran file yang relatif kecil yaitu 15 hingga 20 MB.

3. Pelatihan Model (*Training*)

Setelah dataset sudah jadi, maka langkah selanjutnya yaitu membagi dataset menjadi *train set*, *validation set*, dan *test set*. Proses *training* untuk menghasilkan model melibatkan *train set* dan *validation set*. Dalam penelitian ini, library Tensorflow Lite Model Maker digunakan agar proses *training* model menjadi lebih sederhana. Selain itu, library ini mempermudah proses ekspor model ke dalam bentuk TFLite agar bisa diintegrasikan pada aplikasi *mobile*.



```
Model: "sequential"
-----
Layer (type)                Output Shape              Param #
-----
hub_keras_layer_v1v2 (HubKerasLayerV1V2)  (None, 1280)             2257984
dropout (Dropout)            (None, 1280)             0
dense (Dense)                 (None, 2)                 2562
-----
Total params: 2,260,546
Trainable params: 2,562
Non-trainable params: 2,257,984
```

Gambar 4.4: Layer pada Model

Gambar 4.4 merupakan layer-layer yang digunakan selama proses *transfer learning* untuk model menggunakan arsitektur MobileNetV2. Total parameter yang digunakan selama proses pelatihan

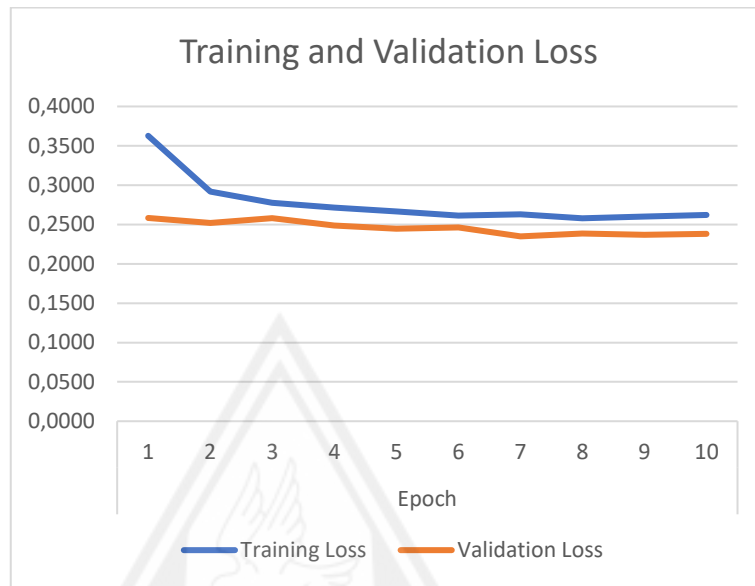
sebanyak 2.260.546. Total parameter tersebut diperoleh berdasarkan total bobot masing-masing layer yang ada pada arsitektur MobileNetV2. Layer pada arsitektur MobileNetV2 dapat dilihat pada Tabel x.

Layer *hub_keras_layer_v1v2* merupakan layer pada *pre-trained* model yang sudah dilatih sebelumnya dengan arsitektur MobileNetV2 menggunakan ImageNet sehingga perlu dibekukan pada saat proses *transfer learning* berlangsung. Layer yang termasuk dalam *hub_keras_layer_v1v2* tersebut, yaitu:

- a) Conv1 (*Convolutional Layer*)
- b) Block 1 hingga Block 13. Block ini terdiri atas convolutional layer (bottleneck 0), depthwise convolutional layer (bottleneck 1), dan pointwise convolutional layer (bottleneck 1).

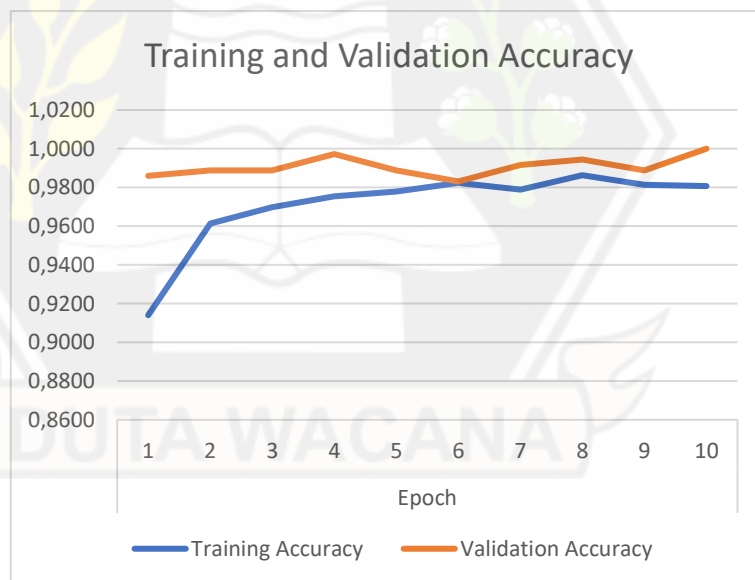
Layer *dense* merupakan layer pada model yang diperbarui pada saat proses *transfer learning* berlangsung. Layer tersebut menghasilkan output sebanyak 2 *class* untuk mengklasifikasikan remote ke dalam *Image Path Remote* dan *remote* dari produk lain.

Total parameter yang diperoleh pada saat proses pelatihan terbagi menjadi 2 kategori, yaitu *trainable params* dan *non-trainable params*. *Trainable params* adalah jumlah parameter yang bobotnya dapat diperbarui, sedangkan *non-trainable params* adalah jumlah parameter yang bobotnya tidak dapat diperbarui selama proses pelatihan *transfer learning* berlangsung. Pada penelitian ini, total bobot yang diperbarui sebanyak 2.562 parameter, sedangkan total bobot yang dibekukan sebesar 2.257.984 parameter.



Gambar 4.5: Grafik *Training* dan *Validation Loss*

Gambar 4.5 merupakan visualisasi grafik dari *training* dan *validation loss* yang terjadi sepanjang proses pelatihan. Berdasarkan grafik tersebut, *loss* untuk *training set* cenderung menurun didukung dengan penurunan *loss* pada *validation set*.



Gambar 4.6: Grafik *Training* dan *Validation Accuracy*

Gambar 4.6 merupakan visualisasi grafik dari *training* dan *validation accuracy* yang terjadi sepanjang proses *training*. Berdasarkan grafik tersebut, *accuracy* untuk *training set* cenderung meningkat diikuti dengan peningkatan *accuracy* untuk *validation set*.

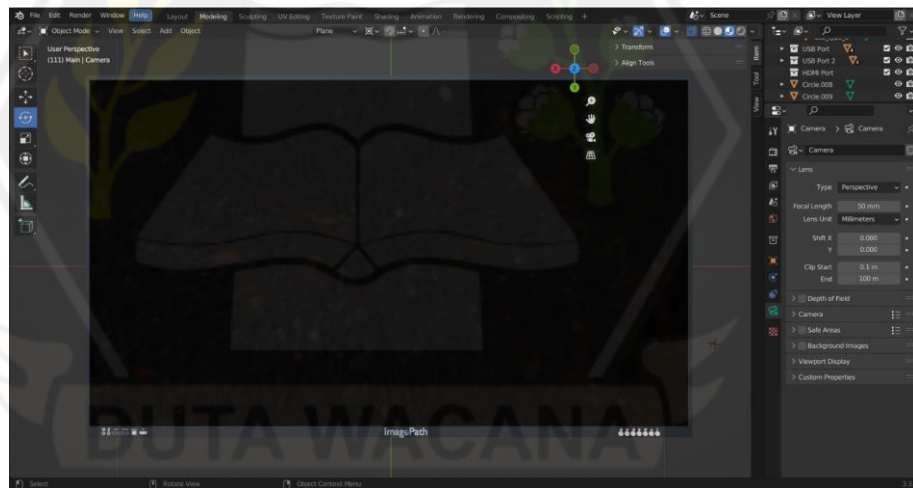
Berdasarkan kedua grafik tersebut, maka dapat disimpulkan jika model yang dihasilkan sudah baik (*goodfit*). Masing-masing grafik menunjukkan tidak adanya ciri-ciri *overfit* maupun *underfit* karena hasil *loss* dan *accuracy* dari proses pelatihan cenderung diikuti dengan hasil yang hampir sama dari proses validasi.

4.1.2. Pembuatan Aplikasi

Dalam penelitian ini, aplikasi dan model untuk deteksi *remote* dibuat secara paralel. Pembuatan aplikasi dalam penelitian ini menggunakan *software* Android Studio. Secara garis besar, berikut tahap yang peneliti lakukan dalam membuat aplikasi:

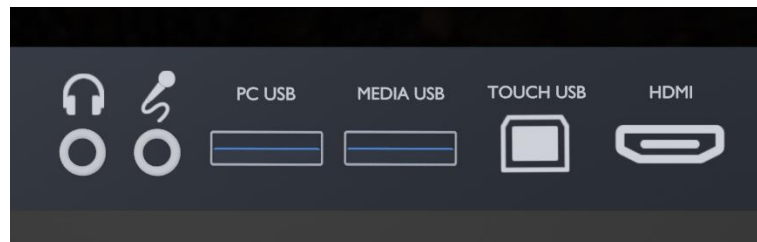
1. Pembuatan 3D Monitor

Peneliti membuat 3D dari Image Path Monitor dengan menggunakan *software* blender. Pembuatan 3D diawali dengan menganalisis setiap komponen yang ada pada monitor, seperti letak port dan tombol.



Gambar 4.7: Pembuatan 3D Image Path Monitor

Gambar 4.7 merupakan tampilan dari 3D dari Image Path Monitor dari sisi depan. Pada 3D tersebut ada beberapa bagian yang menjadi fokus utama dalam penelitian ini, yaitu panel layer, port-port, dan tombol-tombol yang ada pada monitor.



Gambar 4.8: Port Monitor pada Objek 3D



Gambar 4.9: Tombol Monitor pada Objek 3D

Gambar 4.8 merupakan tampilan dari port, sedangkan Gambar 4.9 merupakan tampilan dari tombol yang ada di 3D monitor. Port dan tombol tersebut dibuat menyerupai monitor di objek nyata, mulai dari urutan hingga ikon-ikon.

2. Menambahkan *dependency*

a) *Sceneform*

Sceneform merupakan library yang menggabungkan ARCore SDK dengan Google Filament dan berfungsi untuk memudahkan proses rendering objek virtual.

```
Implementation "com.gorisse.thomas.sceneform:sceneform:1.21.0"
```

Tambahkan kode di atas ke dalam file build.gradle, kemudian klik “Sync Now” untuk memulai proses instalasi package sceneform ke dalam projek.

b) ML Kit

ML Kit merupakan SDK yang memungkinkan *developer* untuk mengimplementasikan *Machine Learning* dalam aplikasi Android dan iOS.

```
implementation 'com.google.mlkit:object-detection-custom:17.0.0'
```

Tambahkan kode di atas ke dalam file `build.gradle`, kemudian klik “*Sync Now*” untuk memulai proses instalasi package ML Kit ke dalam projek.

3. Pembuatan Tombol Interaktif

a) Port Monitor



Gambar 4.10: *Layout Port Monitor*

Gambar 4.10 merupakan layout yang peneliti buat untuk menampilkan port monitor pada objek digital. Berikut merupakan keterangan untuk masing-masing port monitor yang ada pada layout tersebut:



Gambar 4.11: *Port Audio Out*

Gambar 4.11 merupakan tombol dengan ikon *headphone* dan berfungsi untuk menampilkan petunjuk terkait port *audio-out* pada monitor.



Gambar 4.12: *Port Mic Out*

Gambar 4.12 merupakan tombol dengan ikon *microphone* dan berfungsi untuk menampilkan petunjuk terkait port *mic-out* pada monitor.



Gambar 4.13: Port PC USB dan Media USB

Gambar 4.13 merupakan tombol dengan ikon USB tipe A dan berfungsi untuk menampilkan petunjuk terkait port *PC USB* dan *media USB* pada monitor.



Gambar 4.14: *Port Touch USB*

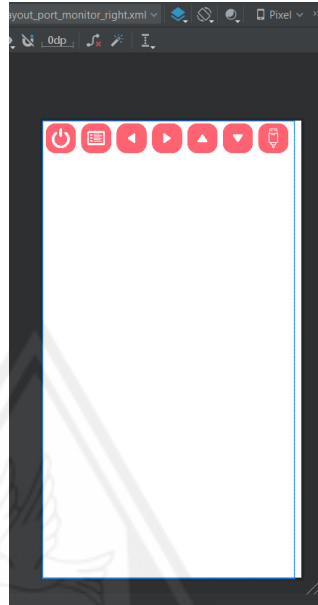
Gambar 4.14 merupakan tombol dengan ikon USB tipe B dan berfungsi untuk menampilkan petunjuk terkait port *touch USB* pada monitor.



Gambar 4.15: Port HDMI

Gambar 4.15 merupakan tombol dengan ikon HDMI dan berfungsi untuk menampilkan petunjuk terkait port *HDMI* pada monitor.

b) Tombol Monitor



Gambar 4.16: Layout Tombol Monitor

Gambar 4.16 merupakan layout yang peneliti buat untuk menampilkan tombol monitor pada objek digital. Berikut merupakan keterangan untuk masing-masing tombol monitor yang ada pada layout tersebut:



Gambar 4.17: Tombol Power

Gambar 4.17 merupakan tombol dengan ikon *power* dan berfungsi untuk menampilkan petunjuk terkait tombol power pada monitor.



Gambar 4.18: Tombol Pengaturan

Gambar 4.18 merupakan tombol dengan ikon *list* dan berfungsi untuk menampilkan petunjuk terkait tombol pengaturan pada monitor.



Gambar 4.19: Tombol *Volume Down*



Gambar 4.20: Tombol *Volume Up*

Gambar 4.19 merupakan tombol dengan ikon *play*, sedangkan Gambar 4.20 merupakan tombol dengan ikon sama yang diputar sejauh 180°. Masing-masing tombol berfungsi untuk menampilkan petunjuk terkait tombol pengaturan *volume* pada monitor.



Gambar 4.21. Tombol *Gesture*

Gambar 4.21 merupakan tombol dengan ikon kabel USB dan berfungsi untuk menampilkan petunjuk terkait tombol *gesture* pada monitor.

4. Pembuatan Remote Interaktif



Gambar 4.22. *Layout Remote*

Gambar 4.22 merupakan *layout* dari *remote* yang akan dirender sebagai objek digital. Setiap tombol pada *layout* tersebut apabila ditekan akan menampilkan informasi terkait masing-masing tombol pada *remote*.

5. Pembuatan Menu *Image Path Monitor*

Tabel 4.2. *Class*, *Variable*, dan *Method* untuk Menu *Image Path Monitor*

Nama Class	Deskripsi
ImagePathMonitorActivity	<i>Class</i> ini berfungsi untuk mengatur alur kerja dari menu untuk menampilkan <i>user manual</i> dari <i>Image Path Monitor</i> .
AugmentedImageDatabase	<i>Class</i> dari AR Core ini berfungsi agar kamera dari AR Core dapat mengenali gambar yang sudah ditetapkan sebagai <i>marker</i> .
Nama Variable	Deskripsi
database	Objek variabel yang diinisialisasi dari <i>class</i> <i>AugmentedImageDatabase</i> . Objek variabel ini berfungsi untuk menyimpan gambar yang akan dikenali sebagai <i>marker</i> untuk menampilkan objek digital dengan memanggil method <code>addImage()</code> .
Nama Method	Deskripsi

setAugmentedImageDatabase ()	<i>Method</i> dari AR Core ini berfungsi untuk mengatur basis data gambar yang akan digunakan sebagai <i>marker</i> .
detachVideoNode ()	<i>Method</i> ini berfungsi untuk menghapus objek digital berbentuk video.
detachPortScreenNode ()	<i>Method</i> ini berfungsi untuk menghapus objek digital berupa papan petunjuk terkait port.
detachKeyFeaturesScreenNode ()	<i>Method</i> ini berfungsi untuk menghapus objek digital berupa papan petunjuk berisi informasi terkait fitur utama dari monitor.
detachMainScreenNode ()	<i>Method</i> ini berfungsi untuk menghapus objek digital berupa layar utama (beranda) dari monitor.
onAugmentedImageTrackingUpdate ()	<i>Method</i> ini berfungsi untuk mengatur proses tracking dan pengenalan marker serta menampilkan objek-objek digital apabila proses pengenalan marker berhasil.
loadModels ()	<i>Method</i> ini berfungsi untuk memuat model 3D monitor

	yang sudah dibangun sebelumnya dalam bentuk .glb.
loadScreen ()	<i>Method</i> ini berfungsi untuk menampilkan objek digital yang berkaitan dengan layar dan menu dari <i>Image Path Monitor</i> .
loadPortButtonLeft ()	<i>Method</i> ini berfungsi untuk menampilkan objek digital berupa port-port ada di sebelah kiri <i>front panel Image Path Monitor</i> .
loadPortButtonRight ()	<i>Method</i> ini berfungsi untuk menampilkan objek digital berupa tombol-tombol yang ada di sebelah kanan <i>front panel Image Path Monitor</i> .
loadViewRenderableDesc ()	<i>Method</i> ini berfungsi untuk menampilkan objek digital berupa papan petunjuk berisi informasi terkait port maupun tombol pada monitor.
loadButtonScreen ()	<i>Method</i> ini berfungsi untuk menampilkan tombol-tombol untuk menu di <i>Image Path Monitor</i> .

loadKeyFeaturesButton()	<i>Method</i> ini berfungsi untuk menampilkan tombol untuk menampilkan informasi terkait fitur utama dari monitor.
loadKeyFeaturesScreen()	<i>Method</i> ini berfungsi untuk menampilkan papan petunjuk berisi informasi terkait fitur utama dari monitor.
loadMainScreen()	<i>Method</i> ini berfungsi untuk menampilkan layar utama (beranda) dari monitor.
loadVideoMiland()	<i>Method</i> ini berfungsi untuk menampilkan video berisi petunjuk terkait penggunaan menu-menu yang ada di <i>Image Path Monitor</i> .

Tabel 4.2 merupakan detail teknis terkait *class*, *variable*, dan *method* utama yang digunakan untuk membuat menu *Image Path Monitor*. Proses pemindaian marker untuk menampilkan *user manual Image Path Monitor* dilakukan dengan menggunakan teknologi *augmented image*.

6. Pembuatan Menu *Image Path Remote*

Tabel 4.3. *Class*, *Variable*, dan *Method* untuk Menu *Image Path Remote*

Nama Class	Deskripsi
ImagePathRemoteActivity	<i>Class</i> ini berfungsi untuk mengatur alur kerja dari menu untuk menampilkan <i>user manual</i> dari <i>Image Path Remote</i>

MLKitObjectDetector	<i>Class</i> ini berfungsi untuk memuat model TFLite yang telah dibuat sebelumnya dan memproses hasil deteksi objek.
YuvToRgbConverter	<i>Class</i> ini berfungsi untuk mengkonversi input gambar yang diterima oleh kamera dari AR Core yang awalnya berbentuk Yuv menjadi RGB untuk dikenali oleh model.
DetectedObjectResult	<i>Data class</i> ini berfungsi untuk menyimpan hasil deteksi objek yang menjadi output dari ML Kit. Hasil tersebut terdiri atas nilai <i>confidence</i> , nama objek (<i>label</i>), dan nilai <i>center coordinate</i> .
Nama Variable	Deskripsi
mlKitAnalyzer	Objek variabel yang diinisialisasi dari <i>class</i> MLKitObjectDetector
objectResults	Variabel ini berfungsi untuk menampung hasil analisis yang dihasilkan oleh objek mlKitAnalyzer. Proses untuk analisis tersebut terjadi pada <i>method</i> <code>analyze()</code> . Hasil dari <i>method</i> ini berupa list yang berisi objek dari <i>data class</i> DetectedObjectResult.
Nama Method	Deskripsi
createModels()	Method ini berfungsi untuk inisialisasi objek digital remote yang

	<p>akan dirender dengan menggunakan sceneform. Objek digital tersebut berupa layout XML yang dikonversi ke dalam bentuk objek ViewRenderable.</p>
onScannedObject ()	<p>Method ini berfungsi untuk menjalankan proses deteksi objek dan melakukan render terhadap objek digital apabila hasil deteksi objek sudah sesuai.</p>
loadModel ()	<p><i>Method</i> ini berfungsi untuk menampilkan objek digital sesuai dengan koordinat titik tengah yang diperoleh pada saat proses deteksi objek berlangsung.</p>
transformCoordinates2d ()	<p><i>Center coordinate</i> yang dihasilkan oleh ML Kit merupakan titik tengah objek yang dari gambar yang ditangkap kamera (<i>CPU Image</i>). Ada perbedaan ukuran antara gambar yang ditangkap kamera dengan gambar yang ada di layar <i>smartphone</i>. Hal ini menyebabkan koordinat tersebut menjadi tidak sesuai karena ukuran gambar yang ditangkap kamera lebih besar dibanding ukuran gambar pada layar <i>smartphone</i>. Oleh karena itu, <i>method</i> ini digunakan untuk mengubah <i>CPU Image</i> tersebut ke</p>

	dalam bentuk <i>screen view</i> sehingga gambar tersebut sesuai dengan ukuran layar.
<code>screenPointToRay()</code>	Titik tengah koordinat objek yang sudah dikonversi dalam bentuk <i>screen view</i> akan dikonversi menjadi sinar dalam dunia virtual dengan menggunakan <i>method</i> ini. Sinar ini membantu aplikasi AR untuk mengetahui sudut pandang pengguna dan memungkinkan terjadinya interaksi dengan objek digital ketika pengguna menyentuh objek tersebut di layar <i>smartphone</i> .
<code>loadButtonDesc()</code>	Method ini berfungsi untuk inisialisasi objek digital berupa papan petunjuk terkait informasi tombol yang ada pada remote dan akan dirender dengan menggunakan <i>sceneform</i> . Objek digital tersebut berupa layout XML yang dikonversi ke dalam bentuk objek <i>ViewRenderable</i> .
<code>removeRenderableButton()</code>	<i>Method</i> ini berfungsi untuk menghapus objek papan petunjuk tombol dan dipanggil pada saat pengguna ingin melihat informasi terkait tombol yang lain atau pengguna menekan kembali tombol tersebut.

Tabel 4.3 merupakan detail teknis terkait *class*, *variable*, dan *method* utama yang digunakan untuk membuat menu Image Path Remote. Proses pemindaian marker berupa objek remote untuk menampilkan *user manual Image Path Remote* dilakukan oleh model TFLite yang telah dihasilkan sebelumnya. Model ini diintegrasikan pada aplikasi dengan menggunakan ML Kit.

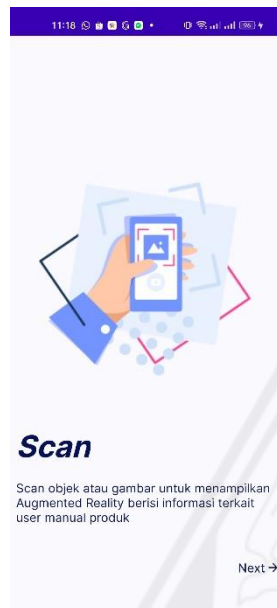
7. Implementasi *Prototype*

Berikut adalah implementasi nyata dari masing-masing *prototype* yang telah dibuat sebelumnya. Proses implementasi *prototype* menjadi aplikasi nyata melibatkan bahan-bahan yang telah dikumpulkan sebelumnya.

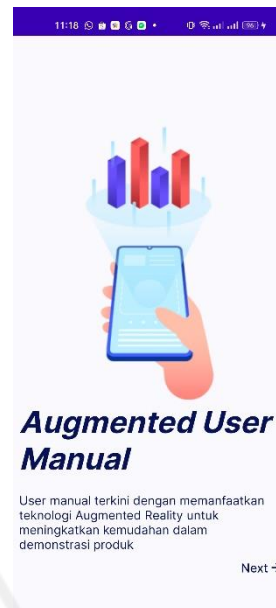


Gambar 4.23: *Screenshot* Halaman *Splash Screen* pada Aplikasi

Gambar 4.23 merupakan *screenshot* dari *splash screen* aplikasi. *Splash screen* ini akan muncul selama 3 detik sebelum pengguna dapat mengakses menu utama di dalam aplikasi.



Gambar 4.24: Screenshot Onboarding Screen 1 pada Aplikasi

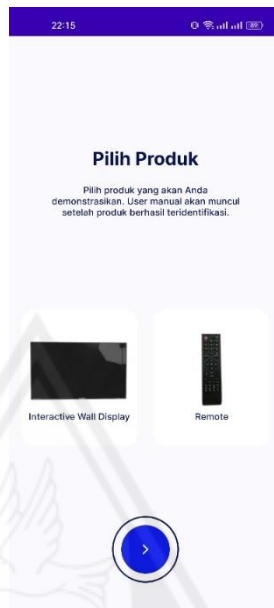


Gambar 4.25: Screenshot Onboarding Screen 2 pada Aplikasi



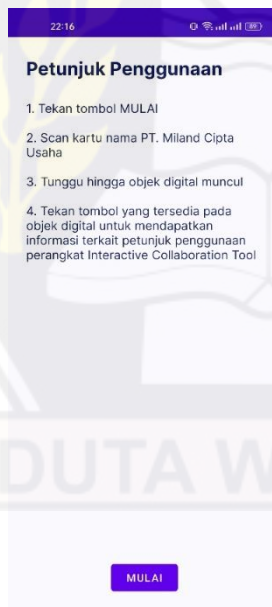
Gambar 4.26: Screenshot Onboarding Screen 3 pada Aplikasi

Gambar 4.24, Gambar 4.25, dan Gambar 4.26 merupakan *screenshot* dari halaman *onboarding* pada aplikasi. Halaman ini hanya akan muncul pada saat pengguna menggunakan aplikasi untuk pertama kali.

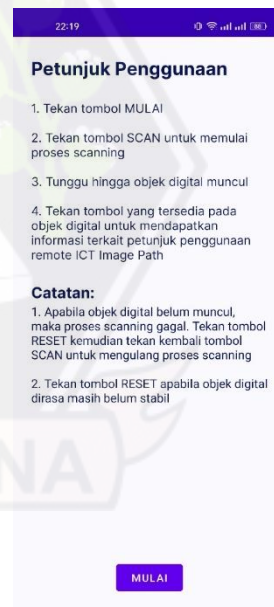


Gambar 4.27: *Screenshot* Menu Utama pada Aplikasi

Gambar 4.27 merupakan *screenshot* dari menu utama aplikasi. Pada menu ini pengguna akan disediakan opsi untuk mengakses *user manual* dari *Image Path Monitor* maupun *Image Path Remote*.



Gambar 4.28: *Screenshot* Halaman Petunjuk Penggunaan Menu *Image Path Monitor* pada Aplikasi



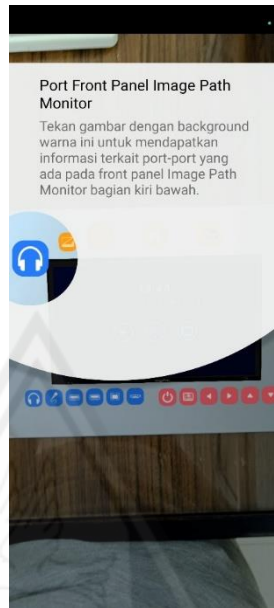
Gambar 4.29: *Screenshot* Halaman Petunjuk Penggunaan Menu *Image Path Remote* pada Aplikasi

Gambar 4.28 dan Gambar 4.29 merupakan *screenshot* dari halaman yang berisi petunjuk untuk menggunakan menu *Image Path Monitor* dan *Image Path Remote*. Apabila pengguna sudah paham, maka pengguna dapat mengakses menu tersebut dengan menekan tombol “MULAI”.



Gambar 4.30: *Screenshot* Tampilan Pemindaian *Marker* untuk Akses *User Manual Image Path Monitor* pada Aplikasi

Gambar 4.30 merupakan *screenshot* pada saat pengguna perlu memindai *marker* berupa gambar kartu PT. Miland Cipta Usaha. Pengguna akan diminta untuk mengarahkan kamera ke *marker*.



Gambar 4.31: *Screenshot Prompt* Petunjuk Interaksi *Image Path Monitor* pada Aplikasi

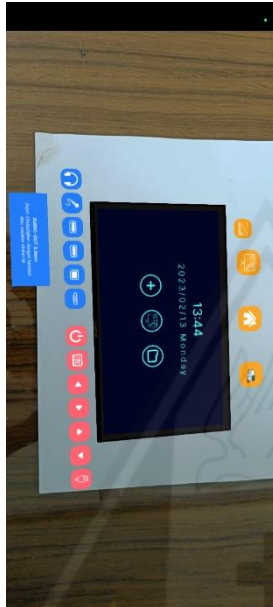
Gambar 4.31 merupakan *screenshot* dari *prompt* berisi petunjuk yang muncul setelah proses pemindaian berhasil. *Prompt* tersebut berisi petunjuk untuk berinteraksi dengan objek digital *Image Path Monitor*.



Gambar 4.32: *Screenshot* Objek Digital *Image Path Monitor* pada Aplikasi

Gambar 4.32 merupakan *screenshot* dari objek digital monitor yang muncul pada aplikasi. Pada *screenshot* tersebut terlihat ada 3D

monitor beserta dengan tombol-tombol interaktif untuk memberikan petunjuk penggunaan monitor.



Gambar 4.33: *Screenshot* Papan



Gambar 4.34: *Screenshot* Papan

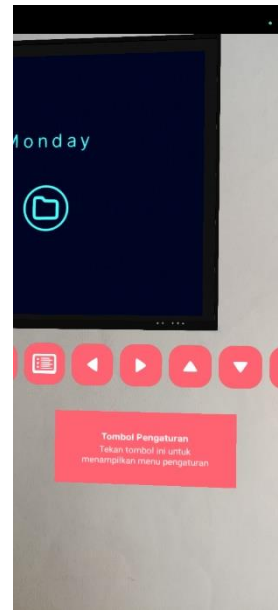
Petunjuk Port Monitor pada Aplikasi 1

Petunjuk Port Monitor pada Aplikasi 2

Gambar 4.33 dan Gambar 4.34 merupakan *screenshot* dari papan petunjuk yang muncul setelah pengguna menekan tombol yang ada di kiri bawah monitor. Papan tersebut berisi informasi terkait port yang ada di monitor.



Gambar 4.35: *Screenshot* Papan Petunjuk Tombol Monitor pada Aplikasi 1



Gambar 4.36: *Screenshot* Papan Petunjuk Tombol Monitor pada Aplikasi 2

Gambar 4.35 dan Gambar 4.36 merupakan *screenshot* dari papan petunjuk yang muncul setelah pengguna menekan tombol yang ada di kanan bawah monitor. Papan tersebut berisi informasi terkait tombol yang ada di monitor.

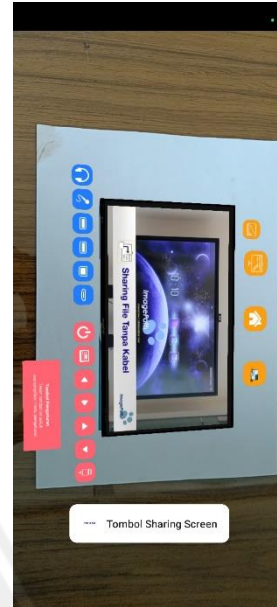


Gambar 4.37: *Screenshot* Papan berisi Fitur Utama Monitor pada Aplikasi

Gambar 4.37 merupakan *screenshot* dari papan yang menampilkan informasi terkait fitur utama dari monitor.



Gambar 4.38: *Screenshot* dari Papan yang Menampilkan Video S-Write



Gambar 4.39: *Screenshot* dari Papan yang Menampilkan Video *Sharing File* tanpa Kabel

Gambar 4.38 dan Gambar 4.39 merupakan *screenshot* dari papan yang menampilkan video terkait penggunaan fitur *whiteboard* maupun fitur *sharing file* tanpa kabel.



Gambar 4.40: *Screenshot* Tampilan Pemindaian *Marker* untuk Akses *User Manual Image Path Remote* pada Aplikasi

Gambar 4.40 merupakan *screenshot* pada saat pengguna perlu memindai *marker* berupa objek *remote*. Pengguna akan diminta untuk mengarahkan kamera ke objek dan menekan tombol “*Scan*” untuk memulai proses pemindaian.



Gambar 4.41: *Screenshot Prompt* Petunjuk Interaksi *Image Path Remote* pada Aplikasi

Gambar 4.41 merupakan *screenshot* dari *prompt* berisi petunjuk yang muncul setelah proses pemindaian berhasil. *Prompt* tersebut berisi petunjuk untuk berinteraksi dengan objek digital *Image Path Remote*.



Gambar 4.42: *Screenshot* Objek Digital *Image Path Remote* pada Aplikasi

Gambar 4.42 merupakan *screenshot* dari objek digital *remote* yang muncul pada aplikasi. Pada objek digital tersebut terdapat tombol-tombol interaktif yang apabila ditekan akan menampilkan papan berisi petunjuk terkait nama dan deskripsi dari tombol tersebut.

4.2. Testing

Pada tahap ini, pengujian dilakukan dengan dua metode, yaitu *confusion matrix* dan *cognitive walkthrough*. Metode *confusion matrix* digunakan untuk menguji akurasi dari model yang dihasilkan oleh arsitektur MobileNetV2 untuk deteksi remote. Di sisi lain, metode *Cognitive Walkthrough* digunakan untuk menguji *usability* dari AR sebagai *user manual*.

4.2.1. Hasil Pengujian Model MobileNetV2 dengan Metode *Confusion Matrix*

Dalam penelitian ini, pengujian untuk model dilakukan dengan melakukan proses pemindaian objek remote secara langsung menggunakan aplikasi. Pengujian ini perlu dilakukan untuk mengetahui nilai akurasi, presisi, dan *recall* yang

dihasilkan oleh model. Sebelum melakukan pengujian, kode pada aplikasi perlu dimodifikasi agar dapat menampilkan nilai *confidence* setelah menekan tombol “SCAN”. Nilai ini menunjukkan akurasi yang dihasilkan oleh model pada saat mendeteksi objek *remote*.

Pengujian dilakukan sebanyak 80 kali terhadap berbagai macam objek remote. Hasil dari pengujian tersebut kemudian dimasukkan ke dalam tabel *confusion matrix* untuk dilakukan perhitungan *accuracy*, *precision*, dan *recall*.



Gambar 4.43: Screenshot Pengujian Model untuk Klasifikasi *Image Path Remote*



Gambar 4.44: Screenshot Pengujian Model untuk Klasifikasi Remote dari Produk Lain

Gambar 4.43 dan Gambar 4.44 merupakan contoh proses pengujian model untuk mendeteksi *remote*. Pada Gambar 4.43, model berhasil mengklasifikasikan objek ke dalam kategori *Image Path Remote* dengan nilai tingkat kepercayaan (*confidence*) sebesar 0,8258433, sedangkan pada Gambar 4.44, model berhasil mengklasifikasikan objek ke dalam kategori bukan *Image Path Remote* dengan nilai tingkat kepercayaan sebesar 0,8830517.

Tabel 4.4: Tingkat Kepercayaan Model

Pengujian	Objek	Hasil Klasifikasi	<i>Confidence</i>	Keterangan
1	<i>Image Path Remote</i>	<i>Image Path Remote</i>	0,6497704	Tempat dengan kondisi cukup cahaya

2	<i>Image Path Remote</i>	<i>Image Path Remote</i>	0,6107450	Tempat dengan kondisi cukup cahaya
3	<i>Image Path Remote</i>	<i>Image Path Remote</i>	0,6096864	Tempat dengan kondisi cukup cahaya
4	<i>Image Path Remote</i>	<i>Image Path Remote</i>	0,6809787	Tempat dengan kondisi cukup cahaya
5	<i>Image Path Remote</i>	<i>Image Path Remote</i>	0,6709595	Tempat dengan kondisi cukup cahaya
6	<i>Image Path Remote</i>	<i>Image Path Remote</i>	0,6581712	Tempat dengan kondisi cukup cahaya
7	<i>Image Path Remote</i>	<i>Image Path Remote</i>	0,6845605	Tempat dengan kondisi cukup cahaya
8	<i>Image Path Remote</i>	<i>Image Path Remote</i>	0,5530394	Tempat dengan kondisi cukup cahaya
9	<i>Image Path Remote</i>	<i>Image Path Remote</i>	0,6058785	Tempat dengan kondisi cukup cahaya
10	<i>Image Path Remote</i>	<i>Image Path Remote</i>	0,5873234	Tempat dengan kondisi cukup cahaya
11	<i>Image Path Remote</i>	Bukan <i>Image Path Remote</i>	0,6545667	Tempat dengan <i>background</i> hitam
12	<i>Image Path Remote</i>	<i>Image Path Remote</i>	0,5841098	Tempat dengan <i>background</i> hitam
13	<i>Image Path Remote</i>	Bukan <i>Image Path Remote</i>	0,5773235	Tempat dengan <i>background</i> hitam
14	<i>Image Path Remote</i>	Bukan <i>Image Path Remote</i>	0,5694278	Tempat dengan <i>background</i> hitam
15	<i>Image Path Remote</i>	Bukan <i>Image Path Remote</i>	0,5023428	Tempat dengan <i>background</i> hitam
16	<i>Image Path Remote</i>	Bukan <i>Image Path Remote</i>	0,5051277	Tempat dengan <i>background</i> hitam
17	<i>Image Path Remote</i>	<i>Image Path Remote</i>	0,5465323	Tempat dengan <i>background</i> hitam
18	<i>Image Path Remote</i>	Bukan <i>Image Path Remote</i>	0,5229352	Tempat dengan <i>background</i> hitam

19	<i>Image Path Remote</i>	Bukan <i>Image Path Remote</i>	0,5271602	Tempat dengan <i>background</i> hitam
20	<i>Image Path Remote</i>	<i>Image Path Remote</i>	0,5202070	Tempat dengan <i>background</i> hitam
21	<i>Image Path Remote</i>	<i>Image Path Remote</i>	0,8258433	Tempat dengan <i>background</i> coklat
22	<i>Image Path Remote</i>	<i>Image Path Remote</i>	0,7585927	Tempat dengan <i>background</i> coklat
23	<i>Image Path Remote</i>	<i>Image Path Remote</i>	0,7955681	Tempat dengan <i>background</i> coklat
24	<i>Image Path Remote</i>	<i>Image Path Remote</i>	0,7683187	Tempat dengan <i>background</i> coklat
25	<i>Image Path Remote</i>	<i>Image Path Remote</i>	0,6978399	Tempat dengan <i>background</i> coklat
26	<i>Image Path Remote</i>	<i>Image Path Remote</i>	0,7311341	Tempat dengan <i>background</i> coklat
27	<i>Image Path Remote</i>	<i>Image Path Remote</i>	0,6206701	Tempat dengan <i>background</i> coklat
28	<i>Image Path Remote</i>	<i>Image Path Remote</i>	0,7163995	Tempat dengan <i>background</i> coklat
29	<i>Image Path Remote</i>	<i>Image Path Remote</i>	0,6849332	Tempat dengan <i>background</i> coklat
30	<i>Image Path Remote</i>	<i>Image Path Remote</i>	0,6968881	Tempat dengan <i>background</i> coklat
31	<i>Image Path Remote</i>	Bukan <i>Image Path Remote</i>	0,7911708	Tempat dengan kondisi kurang cahaya
32	<i>Image Path Remote</i>	Bukan <i>Image Path Remote</i>	0,6892681	Tempat dengan kondisi kurang cahaya
33	<i>Image Path Remote</i>	Bukan <i>Image Path Remote</i>	0,7460851	Tempat dengan kondisi kurang cahaya
34	<i>Image Path Remote</i>	Bukan <i>Image Path Remote</i>	0,7764080	Tempat dengan kondisi kurang cahaya
35	<i>Image Path Remote</i>	Bukan <i>Image Path Remote</i>	0,7128415	Tempat dengan kondisi kurang cahaya
36	<i>Image Path Remote</i>	Bukan <i>Image Path Remote</i>	0,6413772	Tempat dengan kondisi kurang cahaya

37	<i>Image Path Remote</i>	Bukan <i>Image Path Remote</i>	0,7431845	Tempat dengan kondisi kurang cahaya
38	<i>Image Path Remote</i>	Bukan <i>Image Path Remote</i>	0,6856444	Tempat dengan kondisi kurang cahaya
39	<i>Image Path Remote</i>	Bukan <i>Image Path Remote</i>	0,5763407	Tempat dengan kondisi kurang cahaya
40	<i>Image Path Remote</i>	Bukan <i>Image Path Remote</i>	0,7566534	Tempat dengan kondisi kurang cahaya
41	Remote TV Sony Tipe 1	Bukan <i>Image Path Remote</i>	0,9548121	Tempat dengan kondisi cukup cahaya
42	Remote TV Sony Tipe 1	Bukan <i>Image Path Remote</i>	0,9713618	Tempat dengan kondisi cukup cahaya
43	Remote TV Sony Tipe 1	Bukan <i>Image Path Remote</i>	0,9142512	Tempat dengan <i>background</i> hitam
44	Remote TV Sony Tipe 1	Bukan <i>Image Path Remote</i>	0,9130182	Tempat dengan <i>background</i> hitam
45	Remote TV Sony Tipe 1	Bukan <i>Image Path Remote</i>	0,8677617	Tempat dengan <i>background</i> coklat
46	Remote TV Sony Tipe 1	Bukan <i>Image Path Remote</i>	0,8808038	Tempat dengan <i>background</i> coklat
47	Remote TV Sony Tipe 1	Bukan <i>Image Path Remote</i>	0,9802246	Tempat dengan kondisi kurang cahaya
48	Remote TV Sony Tipe 1	Bukan <i>Image Path Remote</i>	0,9843286	Tempat dengan kondisi kurang cahaya
49	Remote TV Sony Tipe 2	Bukan <i>Image Path Remote</i>	0,9190183	Tempat dengan kondisi cukup cahaya
50	Remote TV Sony Tipe 2	Bukan <i>Image Path Remote</i>	0,8942744	Tempat dengan kondisi cukup cahaya
51	Remote TV Sony Tipe 2	Bukan <i>Image Path Remote</i>	0,9037845	Tempat dengan <i>background</i> hitam
52	Remote TV Sony Tipe 2	Bukan <i>Image Path Remote</i>	0,9120752	Tempat dengan <i>background</i> hitam
53	Remote TV Sony Tipe 2	Bukan <i>Image Path Remote</i>	0,7380040	Tempat dengan <i>background</i> coklat

54	Remote TV Sony Tipe 2	Bukan <i>Image Path Remote</i>	0,8830517	Tempat dengan <i>background</i> coklat
55	Remote TV Sony Tipe 2	Bukan <i>Image Path Remote</i>	0,9271585	Tempat dengan kondisi kurang cahaya
56	Remote TV Sony Tipe 2	Bukan <i>Image Path Remote</i>	0,9311366	Tempat dengan kondisi kurang cahaya
57	Remote Proyektor ViewSonic	Bukan <i>Image Path Remote</i>	0,7843319	Tempat dengan kondisi cukup cahaya
58	Remote Proyektor ViewSonic	Bukan <i>Image Path Remote</i>	0,8801696	Tempat dengan kondisi cukup cahaya
59	Remote Proyektor ViewSonic	Bukan <i>Image Path Remote</i>	0,6910058	Tempat dengan <i>background</i> hitam
60	Remote Proyektor ViewSonic	Bukan <i>Image Path Remote</i>	0,7175266	Tempat dengan <i>background</i> hitam
61	Remote Proyektor ViewSonic	Bukan <i>Image Path Remote</i>	0,5951863	Tempat dengan <i>background</i> coklat
62	Remote Proyektor ViewSonic	Bukan <i>Image Path Remote</i>	0,7679997	Tempat dengan <i>background</i> coklat
63	Remote Proyektor ViewSonic	Bukan <i>Image Path Remote</i>	0,8798243	Tempat dengan kondisi kurang cahaya
64	Remote Proyektor ViewSonic	Bukan <i>Image Path Remote</i>	0,8829676	Tempat dengan kondisi kurang cahaya
65	Remote AC TCL	Bukan <i>Image Path Remote</i>	0,9734463	Tempat dengan kondisi cukup cahaya
66	Remote AC TCL	Bukan <i>Image Path Remote</i>	0,9527918	Tempat dengan kondisi cukup cahaya
67	Remote AC TCL	Bukan <i>Image Path Remote</i>	0,9760083	Tempat dengan <i>background</i> hitam
68	Remote AC TCL	Bukan <i>Image Path Remote</i>	0,9726495	Tempat dengan <i>background</i> hitam
69	Remote AC TCL	Bukan <i>Image Path Remote</i>	0,9887224	Tempat dengan <i>background</i> coklat

70	Remote AC TCL	Bukan <i>Image Path Remote</i>	0,9900755	Tempat dengan <i>background</i> coklat
71	Remote AC TCL	Bukan <i>Image Path Remote</i>	0,9786774	Tempat dengan kondisi kurang cahaya
72	Remote AC TCL	Bukan <i>Image Path Remote</i>	0,9639479	Tempat dengan kondisi kurang cahaya
73	Remote Mi TV Stick	Bukan <i>Image Path Remote</i>	0,9349139	Tempat dengan kondisi cukup cahaya
74	Remote Mi TV Stick	Bukan <i>Image Path Remote</i>	0,8996553	Tempat dengan kondisi cukup cahaya
75	Remote Mi TV Stick	Bukan <i>Image Path Remote</i>	0,8993185	Tempat dengan <i>background</i> hitam
76	Remote Mi TV Stick	Bukan <i>Image Path Remote</i>	0,8560094	Tempat dengan <i>background</i> hitam
77	Remote Mi TV Stick	Bukan <i>Image Path Remote</i>	0,9159276	Tempat dengan <i>background</i> coklat
78	Remote Mi TV Stick	Bukan <i>Image Path Remote</i>	0,9322605	Tempat dengan <i>background</i> coklat
79	Remote Mi TV Stick	Bukan <i>Image Path Remote</i>	0,8633377	Tempat dengan kondisi kurang cahaya
80	Remote Mi TV Stick	Bukan <i>Image Path Remote</i>	0,8897490	Tempat dengan kondisi kurang cahaya
Rata-Rata Tingkat Kepercayaan (Confidence)			0,774969685	

Pengujian model dilakukan sebanyak 40 kali untuk *Image Path Remote* dan 40 kali untuk remote dari produk lain. Berdasarkan Tabel 4.4, hasil pengujian menunjukkan rata-rata persentase nilai tingkat kepercayaan sebesar $0,774969685 \approx 0,775$.

Tabel 4.5: Tabel *Confusion Matrix* Hasil Pengujian

N = 80	<i>Actual Miland Product</i>	<i>Actual Other Product</i>
<i>Predicted Miland Product</i>	23	0

<i>Predicted Other Product</i>	17	40
---------------------------------------	----	----

Tabel 4.5 merupakan tabel *confusion matrix* yang diperoleh setelah pengujian berlangsung. Pada tabel tersebut, model berhasil mendeteksi seluruh objek *remote* dengan benar. Hasil pengujian menggunakan *confusion matrix* menunjukkan nilai *true positive* sebesar 23, *false positive* sebesar 0, *false negative* sebesar 17, dan *true negative* sebesar 40.

$$accuracy = \frac{23 + 40}{23 + 40 + 0 + 17} = 0,7875 \quad (4.1)$$

Pada penelitian ini dilakukan perhitungan menggunakan rumus yang ada persamaan 2.1 untuk mengetahui tingkat akurasi yang dihasilkan oleh model. Berdasarkan hasil perhitungan pada persamaan 4.1, model yang dilatih dengan menggunakan arsitektur MobileNetV2 berhasil memperoleh nilai akurasi sebesar 0,7875.

$$precision = \frac{23}{23 + 0} = 1 \quad (4.2)$$

Di sisi lain, pada penelitian dilakukan perhitungan menggunakan rumus yang ada pada persamaan 2.2 untuk mengetahui nilai presisi yang dihasilkan oleh model. Perhitungan ini dilakukan untuk mengetahui perbandingan antara jumlah prediksi benar dengan hasil yang dideteksi oleh sistem. Berdasarkan hasil yang diperoleh pada persamaan 4.2 diperoleh nilai presisi sebesar 1 (satu).

$$recall = \frac{23}{23 + 17} = 0,575 \quad (4.3)$$

Selain itu, pada penelitian dilakukan perhitungan menggunakan rumus yang ada pada persamaan 2.3 untuk mengetahui nilai *recall* yang dihasilkan oleh model. Perhitungan ini dilakukan untuk mengetahui perbandingan antara jumlah prediksi benar dengan hasil yang sebenarnya. Berdasarkan hasil yang diperoleh pada persamaan 4.3 diperoleh hasil sebesar 0,575.

4.2.2. Hasil Pengujian Usability dengan Metode Cognitive Walkthrough

Dalam penelitian ini, pengujian aplikasi dilakukan sebanyak dua iterasi dan masing-masing iterasi terdiri atas tujuh orang responden. Iterasi pertama dilakukan pada hari Jumat, 26 Mei 2023, sedangkan iterasi kedua dilakukan pada hari Selasa,

30 Mei 2023. Pengujian *usability* ini perlu dilakukan untuk mengetahui efektivitas teknologi AR dalam menampilkan *user manual* bagi pengguna.

1. Tingkat Penyelesaian *Task*

Dalam pengujian ini, perhitungan tingkat penyelesaian *task* perlu dilakukan berdasarkan persentase *task* yang berhasil diselesaikan tanpa bantuan dan *task* yang berhasil diselesaikan dengan bantuan oleh responden.

Tabel 4.6: Tingkat Penyelesaian *Task* Iterasi 1

Responden	Tingkat Penyelesaian <i>Task</i>	
	Sukses tanpa bantuan	Sukses dengan bantuan
1	1 (<i>Task 2</i>)	2 (<i>Task 1</i> dan <i>Task 3</i>)
2	1 (<i>Task 1</i>)	2 (<i>Task 2</i> dan <i>Task 3</i>)
3	2 (<i>Task 1</i> dan <i>Task 2</i>)	1 (<i>Task 3</i>)
4	2 (<i>Task 1</i> dan <i>Task 2</i>)	1 (<i>Task 3</i>)
5	2 (<i>Task 1</i> dan <i>Task 2</i>)	1 (<i>Task 3</i>)
6	1 (<i>Task 1</i>)	2 (<i>Task 2</i> dan <i>Task 3</i>)
7	1 (<i>Task 2</i>)	2 (<i>Task 1</i> dan <i>Task 3</i>)

Tabel 4.7: Tingkat Penyelesaian *Task* Iterasi 2

Responden	Tingkat Penyelesaian <i>Task</i>	
	Sukses tanpa bantuan	Sukses dengan bantuan
1	3 (<i>Task 1, Task 2, Task 3</i>)	0
2	3 (<i>Task 1, Task 2, Task 3</i>)	0
3	3 (<i>Task 1, Task 2, Task 3</i>)	0
4	3 (<i>Task 1, Task 2, Task 3</i>)	0
5	3 (<i>Task 1, Task 2, Task 3</i>)	0
6	3 (<i>Task 1, Task 2, Task 3</i>)	0
7	3 (<i>Task 1, Task 2, Task 3</i>)	0

Tabel 4.6 memberikan informasi terkait jumlah *task* yang berhasil diselesaikan oleh masing-masing responden pada iterasi 1, sedangkan Tabel 4.7 memberikan informasi terkait jumlah *task* yang berhasil diselesaikan oleh masing-masing responden pada iterasi 2. Berdasarkan tabel tersebut, peneliti merangkum jumlah *task* yang sukses tanpa bantuan dan jumlah *task* yang sukses dengan bantuan ke dalam Tabel 4.8 dan Tabel 4.9.

Tabel 4.8: Rangkuman Jumlah *Task* Sukses Iterasi 1

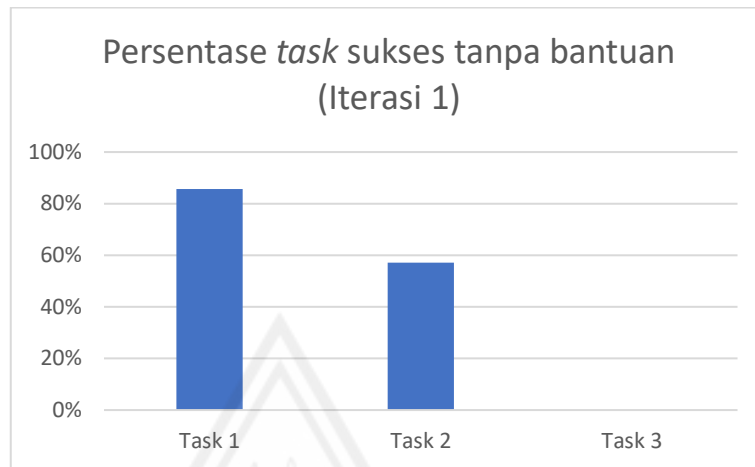
	Sukses tanpa bantuan	Sukses dengan bantuan
<i>Task 1</i>	6	1
<i>Task 2</i>	4	3
<i>Task 3</i>	0	7

Tabel 4.9: Rangkuman Jumlah *Task* Sukses Iterasi 2

	Sukses tanpa bantuan	Sukses dengan bantuan
<i>Task 1</i>	7	0
<i>Task 2</i>	7	0
<i>Task 3</i>	7	0

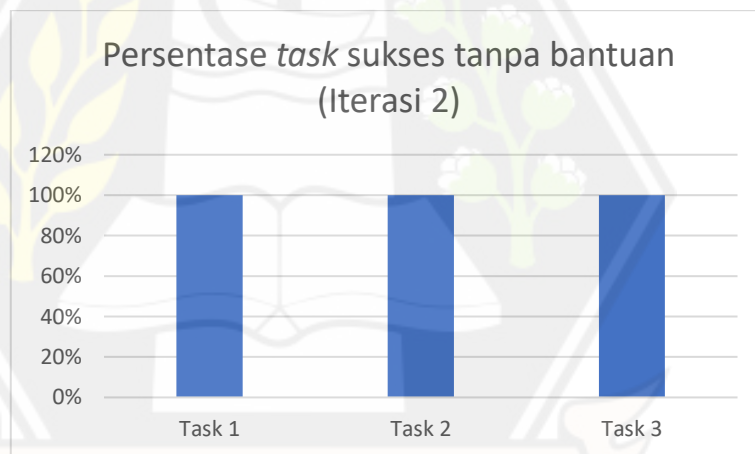
Selanjutnya, perhitungan persentase perlu dilakukan untuk masing-masing *task* yang berhasil tanpa bantuan untuk mengetahui efektivitas AR sebagai *user manual* dengan menggunakan rumus statistika pada persamaan 4.4.

$$\frac{\text{Jumlah responden sukses menyelesaikan task tanpa bantuan}}{\text{Total responden}} \times 100\% \quad (4.4)$$



Gambar 4.45: Grafik Persentase *Task* Sukses Tanpa Bantuan (Iterasi 1)

Gambar 4.45 menunjukkan persentase *task* yang sukses tanpa bantuan untuk iterasi 1. Berdasarkan grafik tersebut, *task 1* berhasil diselesaikan oleh 86% responden, *task 2* berhasil diselesaikan oleh 57% responden, dan tidak ada responden yang berhasil menyelesaikan *task 3* tanpa bantuan.



Gambar 4.46: Grafik Persentase *Task* Sukses Tanpa Bantuan (Iterasi 2)

Gambar 4.46 menunjukkan persentase *task* yang sukses tanpa bantuan untuk iterasi 2. Berdasarkan grafik tersebut, *task 1*, *task 2*, dan *task 3* dapat diselesaikan oleh seluruh responden tanpa bantuan.

Selanjutnya dilakukan perhitungan rata-rata persentase *task* yang sukses tanpa bantuan dengan menggunakan rumus statistika pada persamaan 4.5.

$$\frac{\text{Total persentase tingkat task yang sukses tanpa bantuan}}{\text{Jumlah task}} \times 100\% \text{ (4.5)}$$

Berdasarkan perhitungan diperoleh rata-rata persentase *task* yang sukses tanpa bantuan sebesar 47,67% untuk iterasi 1, sedangkan iterasi 2 menghasilkan rata-rata persentase sebesar 100%.

2. Tingkat Kesalahan

Selain itu, peneliti juga menghitung jumlah kesalahan yang timbul dan mengkategorikan setiap masalah tersebut untuk mengetahui efektivitas AR sebagai *user manual*. Jumlah kesalahan adalah banyaknya kesalahan yang dilakukan responden saat melakukan *task* yang tidak memenuhi tujuan dari *task* yang telah ditentukan sebelumnya (Putri dkk., 2021). Berdasarkan jumlah kesalahan tersebut, peneliti dapat mengetahui tingkat kesulitan yang dialami pengguna dalam menggunakan teknologi AR untuk menampilkan *user manual*.

Tabel 4.10. Jumlah Kesalahan Responden Iterasi 1

Responden	Jumlah Kesalahan			Total
	Task 1	Task 2	Task 3	
1	2	2	2	6
2	1	1	3	5
3	1	2	2	5
4	1	2	3	6
5	1	0	2	3
6	1	1	2	4
7	2	3	2	7
Total	9	11	16	36

Tabel 4.10 merupakan rekap jumlah kesalahan yang dilakukan oleh responden untuk masing-masing *task* pada iterasi satu. Berdasarkan tabel tersebut, responden menghasilkan 9 kesalahan pada saat melaksanakan *task* 1, 11 kesalahan pada saat melaksanakan *task* 2, dan 16 kesalahan pada saat melaksanakan *task* 3. Total kesalahan yang telah terjadi selama proses pengujian berlangsung yaitu 36. Selain itu, peneliti

juga mengkategorikan masing-masing kesalahan yang dilakukan responden berdasarkan masing-masing *task*.

Tabel 4.11. Jumlah Kesalahan Responden Iterasi 2

Responden	Jumlah Kesalahan			Total
	Task 1	Task 2	Task 3	
1	1	1	1	3
2	0	1	2	3
3	0	0	2	2
4	1	1	1	3
5	0	1	2	3
6	0	0	1	1
7	0	0	1	1
Total	2	4	10	16

Tabel 4.11 merupakan rekap jumlah kesalahan yang dilakukan oleh responden untuk masing-masing *task* pada iterasi dua. Berdasarkan tabel tersebut, responden menghasilkan 2 kesalahan pada saat melaksanakan *task 1*, 4 kesalahan pada saat melaksanakan *task 2*, dan 10 kesalahan pada saat melaksanakan *task 3*. Total kesalahan yang telah terjadi selama proses pengujian berlangsung yaitu 16. Selain itu, peneliti juga mengkategorikan masing-masing kesalahan yang dilakukan responden berdasarkan masing-masing *task*.

Tabel 4.12. Kategori Kesalahan Iterasi 1

	Kategori	Jumlah Kesalahan	Total
Task 1	Tidak tahu kalau tombol-tombol bisa ditekan untuk mendapatkan informasi	2	9
	Tombol tidak urut sehingga kadang salah tekan	7	
Task 2	Tidak tahu kalau harus tekan tombol power lebih dahulu	6	11

	Menekan tombol <i>whiteboard</i> yang ada di <i>home screen</i>	3	
	Tidak tahu kalau ada tombol menu yang terpisah dari <i>home screen</i>	1	
	Tombol tidak urut sehingga kadang salah tekan	1	
Task 3	Tidak tahu kalau tombol-tombol bisa ditekan untuk mendapatkan informasi	2	16
	Tidak tahu kalau untuk mendapatkan petunjuk harus sedikit memiringkan <i>smartphone</i> ke bawah	7	
	Menekan tombol selain <i>snapshot</i>	7	
Total			36

Berdasarkan Tabel 4.12, kesalahan yang terjadi pada iterasi 1 saat menjalankan *task* pertama dapat dibagi ke dalam 2 kategori, kesalahan untuk *task* kedua dapat dibagi ke dalam 4 kategori, sedangkan kesalahan untuk *task* 3 dapat dibagi ke dalam 3 kategori.

Tabel 4.13. Kategori Kesalahan Iterasi 2

	Kategori	Jumlah Kesalahan	Total
Task 1	Tombol terlalu kecil	2	2
Task 2	Ikon kurang merepresentasikan menu di monitor	3	4
	Menekan tombol <i>whiteboard</i> yang ada di <i>home screen</i>	1	
Task 3	Deteksi objek gagal	3	10
	Menekan tombol selain <i>snapshot</i>	7	
Total			16

Berdasarkan Tabel 4.13, kesalahan yang terjadi pada iterasi 2 saat menjalankan *task* pertama dapat dibagi ke dalam 1 kategori, kesalahan

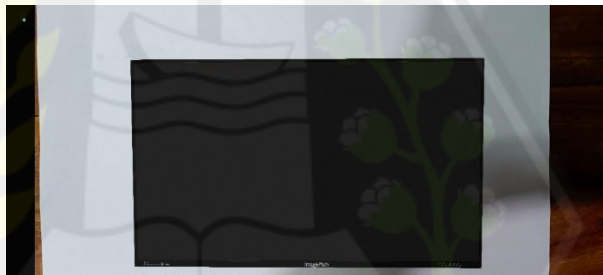
untuk *task* kedua dapat dibagi ke dalam 2 kategori, dan kesalahan untuk *task* 3 dapat dibagi ke dalam 2 kategori.

3. Cerita Pengalaman Pengguna

Setelah menyelesaikan setiap *task*, responden diminta untuk menceritakan pengalamannya selama menggunakan aplikasi berdasarkan pada keempat pertanyaan yang menjadi dasar dalam melakukan pengujian dengan metode *cognitive walkthrough*. Selain itu, responden juga diminta untuk memberikan pendapat atau saran agar aplikasi bisa lebih baik lagi.

4. Perbaikan AR

Berdasarkan pengujian pada iterasi 1, perlu adanya perbaikan untuk tampilan AR untuk menyelesaikan *task* 2 dan *task* 3. Perbaikan ini dilakukan berdasarkan cerita, pendapat, dan saran yang disampaikan oleh responden.



Gambar 4.47. Objek Digital *Image Path Monitor* sebelum Perbaikan

Pada iterasi 1, pengguna mengalami kesulitan untuk menyelesaikan *task* 2 karena pada saat objek digital muncul untuk pertama kali, tidak ada tombol yang memberikan petunjuk untuk menampilkan informasi terkait fitur *whiteboard*. Hal ini terjadi karena pengguna harus menekan tombol power terlebih dahulu untuk mengakses menu tersebut. Gambar 4.47 merupakan tampilan dari objek digital sebelum pengguna menekan tombol power. Oleh karena itu, perubahan dilakukan untuk menghilangkan tahap tersebut sehingga pengguna dapat langsung mendapatkan informasi terkait fitur *whiteboard* tanpa harus menekan tombol power terlebih dahulu. Perubahan tersebut dapat dilihat pada Gambar 4.32.



Gambar 4.48. Objek Digital *Image Path Remote* sebelum Perbaikan

Pada iterasi 1, pengguna mengalami kesulitan pada saat ingin mendapatkan informasi tombol di remote. Hal ini terjadi karena papan petunjuk tersebut terlalu jauh dari objek remote dan tegak lurus. Gambar 4.48 merupakan tampilan dari objek digital sebelum dilakukan perbaikan. Oleh karena itu, perubahan dilakukan dengan mendekatkan jarak papan petunjuk dengan objek remote dan membuat papan petunjuk menjadi sejajar dengan remote. Perubahan tersebut dapat dilihat pada Gambar 4.42.

4.2.3. Pembahasan Hasil Pengujian Model MobileNetV2

Berdasarkan hasil pengujian, model yang dihasilkan dengan menggunakan arsitektur MobileNetV2 berhasil memperoleh nilai rata-rata tingkat kepercayaan sebesar 0,775. Selain itu, hasil perhitungan *confusion matrix* menunjukkan nilai *accuracy* mencapai nilai 0,7875, *precision* mencapai nilai 1 (satu), dan *recall* mencapai nilai 0,575. Keberhasilan model dalam mengklasifikasikan remote dipengaruhi oleh beberapa faktor, seperti warna dari latar belakang objek dan kondisi cahaya dalam suatu ruangan. Berdasarkan Tabel 4.4, model masih belum maksimal dalam mengenali objek *Image Path Remote* dengan benar apabila latar belakang dari objek remote berwarna hitam dan ruangan dalam kondisi kurang

cahaya. Meskipun demikian, hasil rata-rata tingkat kepercayaan diikuti dengan akurasi, presisi, dan *recall* dari tabel *confusion matrix* menunjukkan model yang dihasilkan oleh arsitektur MobileNetV2 sudah cukup akurat untuk mengklasifikasikan objek *Image Path Remote* dengan *remote* lainnya.

Selain itu, model yang dihasilkan dari arsitektur tersebut tergolong ringan untuk ditanamkan pada aplikasi Android. Meskipun terjadi penurunan hasil akurasi dari model pada saat dikonversi menjadi TF Lite, tetapi penurunan tersebut tidak cukup signifikan dan model masih dapat berfungsi untuk mengklasifikasikan objek *remote* dengan baik.

4.2.4. Pembahasan Hasil Pengujian Usability

Rata-rata persentase tingkat penyelesaian *task* tanpa bantuan dari iterasi 1 hanya menunjukkan hasil sebesar 47,67% sehingga banyak pengguna masih kesulitan dalam menyelesaikan *task* tanpa bantuan. Selain itu, pada iterasi 1 telah terjadi kesalahan sebanyak 36 kali. Dengan demikian, hasil pengujian pada iterasi 1 menunjukkan apabila AR masih belum efektif untuk menampilkan *user manual*.

Berdasarkan hasil iterasi 1 tersebut, perbaikan terhadap antarmuka dari AR dan penyederhanaan tahapan dalam menyelesaikan *task* perlu dilakukan untuk meningkatkan persentase tingkat penyelesaian *task* dan menurunkan jumlah kesalahan yang terjadi.

Rata-rata persentase tingkat penyelesaian *task* tanpa bantuan dari iterasi 2 berhasil menunjukkan hasil sebesar 100%. Hasil tersebut menunjukkan peningkatan yang cukup signifikan dibandingkan dengan hasil yang diperoleh dari iterasi 1. Selain itu, pada iterasi 2 telah terjadi kesalahan sebanyak 16 kali. Hasil ini menunjukkan penurunan jumlah kesalahan yang cukup signifikan dibanding dengan hasil yang diperoleh dari iterasi 1. Dengan demikian, hasil pengujian pada iterasi 2 berhasil menunjukkan apabila AR sudah efektif untuk menampilkan *user manual*.

Berdasarkan hasil pengujian *usability* pada iterasi 1 dan iterasi 2 tersebut, ada korelasi yang kuat antara rata-rata persentase *task* sukses tanpa bantuan dengan jumlah kesalahan. Semakin sedikit *task* yang sukses tanpa bantuan, maka semakin

besar juga total kesalahan yang terjadi pada saat responden menggunakan aplikasi begitu juga sebaliknya.

4.3. Distribution

Setelah tahap pembuatan model dan aplikasi selesai, maka aplikasi perlu didistribusikan kepada pengguna sebagai pengganti dari *user manual Interactive Collaboration Tool* berbasis *textbook*. Proses pendistribusian aplikasi dilakukan dengan menunggah aplikasi ke Google Drive, kemudian pengguna dapat mengunduhnya dengan memindai kode QR pada Gambar 4.49.



Gambar 4.49. QR Code Aplikasi

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

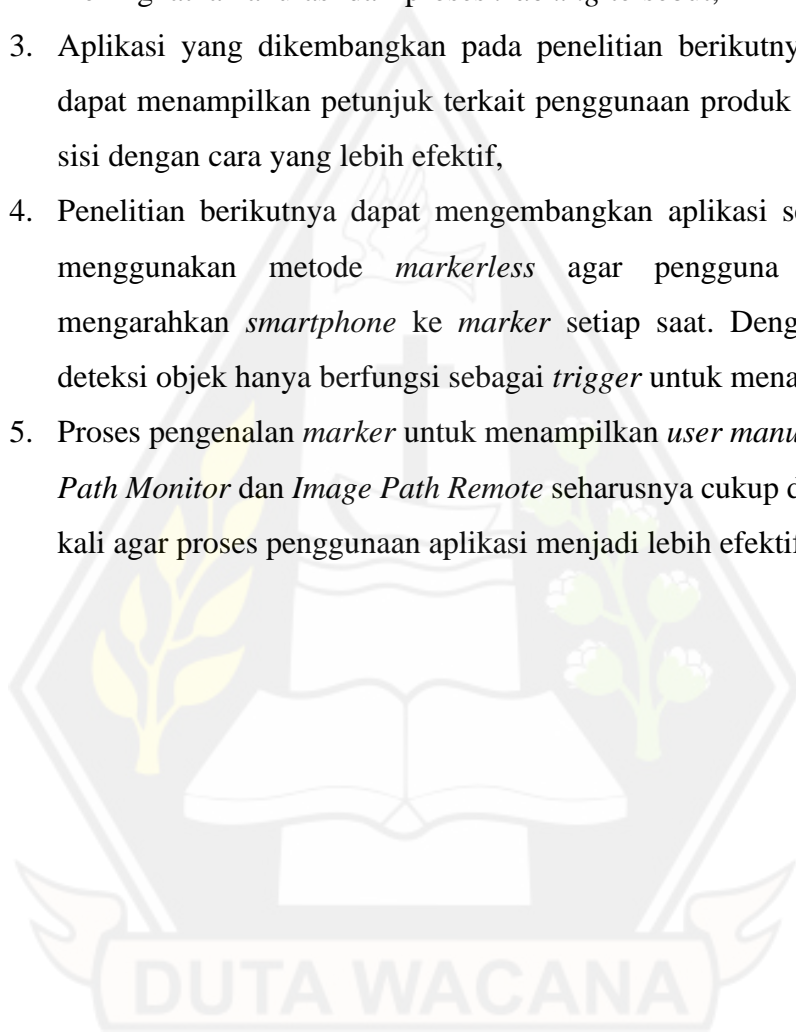
Berdasarkan penelitian ini diperoleh hasil berupa aplikasi yang berfungsi untuk menampilkan *user manual* dengan teknologi AR dan merapkan deteksi objek secara *real-time*. Berbagai pengujian telah dilakukan untuk mengetahui akurasi dari model yang dihasilkan oleh MobileNetV2 dan efektivitas dari teknologi AR sebagai *user manual* produk. Berdasarkan hasil pengujian yang telah dilakukan tersebut, maka dapat diperoleh kesimpulan sebagai berikut:

1. Pengujian dengan menggunakan metode *confusion matrix* menunjukkan jika model yang dibangun menggunakan arsitektur MobileNetV2 berhasil memperoleh akurasi yang cukup baik pada saat diintegrasikan dengan aplikasi Android. Hal ini dapat dilihat berdasarkan rata-rata tingkat kepercayaan yang dihasilkan berhasil memperoleh nilai sebesar 0,775. Hal ini juga didukung dengan nilai akurasi yang dihasilkan mencapai 0,7875, presisi dengan nilai sebesar 1 (satu), dan nilai *recall* sebesar 0,575 berdasarkan tabel *confusion matrix*.
2. Pengujian dengan menggunakan metode *Cognitive Walkthrough* menunjukkan jika teknologi AR dinilai efektif untuk menampilkan *user manual* produk. Hal ini dapat dilihat berdasarkan rata-rata persentase *task* sukses tanpa bantuan pada iterasi 2 berhasil mencapai 100% didukung dengan rendahnya total kesalahan yang terjadi setelah dilakukan perbaikan terhadap AR.

5.2. Saran

Berdasarkan penelitian ini, ada beberapa saran yang dapat dilakukan untuk pengembangan aplikasi lebih lanjut, yaitu:

1. Model untuk deteksi objek dapat dikembangkan menggunakan arsitektur lain kemudian dilakukan perbandingan terhadap arsitektur dari MobileNetV2 dari sisi akurasi maupun performa,
2. Proses *tracking* untuk menampilkan AR menggunakan model masih kurang akurat sehingga perlu adanya penelitian secara khusus untuk meningkatkan akurasi dari proses *tracking* tersebut,
3. Aplikasi yang dikembangkan pada penelitian berikutnya diharapkan dapat menampilkan petunjuk terkait penggunaan produk dari berbagai sisi dengan cara yang lebih efektif,
4. Penelitian berikutnya dapat mengembangkan aplikasi serupa dengan menggunakan metode *markerless* agar pengguna tidak perlu mengarahkan *smartphone* ke *marker* setiap saat. Dengan demikian, deteksi objek hanya berfungsi sebagai *trigger* untuk menampilkan AR,
5. Proses pengenalan *marker* untuk menampilkan *user manual* dari *Image Path Monitor* dan *Image Path Remote* seharusnya cukup dilakukan satu kali agar proses penggunaan aplikasi menjadi lebih efektif.



DAFTAR PUSTAKA

- Alroobaea, R., & Mayhew, P. J. (2014). How many participants are really enough for usability studies? *Proceedings of 2014 Science and Information Conference, SAI 2014*, 48–56. <https://doi.org/10.1109/SAI.2014.6918171>
- Arth, C., Grasset, R., Gruber, L., Langlotz, T., Mulloni, A., & Wagner, D. (2015). *The History of Mobile Augmented Reality*. <http://arxiv.org/abs/1505.01319>
- Berryman, D. R. (2012). Augmented Reality: A Review. Dalam *Medical Reference Services Quarterly* (Vol. 31, Nomor 2, hlm. 212–218). <https://doi.org/10.1080/02763869.2012.670604>
- Cahyaningsih, Y. (2020). Teknologi Augmented Reality pada Promosi Berbasis Android. *Journal of Computer Science and Engineering (JCSE)*, 1(2), 90–115. <https://doi.org/10.36596/jcse.v1i2.60>
- de Paiva Guimarães, M., & Martins, V. F. (2014). A checklist to evaluate augmented reality applications. *Proceedings - 2014 16th Symposium on Virtual and Augmented Reality, SVR 2014*, 45–52. <https://doi.org/10.1109/SVR.2014.17>
- Ilahiyah, S., & Nilogiri, A. (2018). Implementasi Deep Learning Pada Identifikasi Jenis Tumbuhan Berdasarkan Citra Daun Menggunakan Convolutional Neural Network. *JUSTINDO (Jurnal Sistem Dan Teknologi Informasi Indonesia)*, 3(2), 49–56.
- Kolla, S. S. V. K., Sanchez, A., & Plapper, P. (2021). Comparing Effectiveness of Paper Based and Augmented Reality Instructions for Manual Assembly and Training Tasks. *SSRN Electronic Journal*. <https://doi.org/10.2139/ssrn.3859970>
- Manajang, D., Sompie, S. R. U. A., & Jacobus, A. (2021). Implementasi Framework Tensorflow Object Detection API Dalam Mengklasifikasi Jenis Kendaraan Bermotor. *Jurnal Teknik Informatika*, 15(3), 171–178.
- Maria Navin, J. R., & Pankaja, R. (2016). Performance analysis of text classification algorithms using confusion matrix. *International Journal of Engineering and Technical Research (IJETR)*, 6(4), 75–78.

- Müller, L., Aslan, I., & Krüßen, L. (2013). GuideMe: A mobile augmented reality system to display user manuals for home appliances. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8253 LNCS, 152–167. https://doi.org/10.1007/978-3-319-03161-3_11
- Nagrath, P., Jain, R., Madan, A., Arora, R., Kataria, P., & Hemanth, J. (2021). SSDMNV2: A real time DNN-based face mask detection system using single shot multibox detector and MobileNetV2. *Sustainable Cities and Society*, 66, 102692. <https://doi.org/https://doi.org/10.1016/j.scs.2020.102692>
- O’Shea, K., & Nash, R. (2015). *An Introduction to Convolutional Neural Networks*. <http://arxiv.org/abs/1511.08458>
- Paul-Louis Pröve. (2018, April 11). *MobileNetV2: Inverted Residuals and Linear Bottlenecks*. <https://towardsdatascience.com/mobilenetv2-inverted-residuals-and-linear-bottlenecks-8a4362f4ffd5>
- Putri, A. A., Rifky, M., Susanto, N., Azhari, M. A., & Maghfiroh, L. R. (2021). Uji Kegunaan Perangkat Lunak menggunakan Metode Cognitive Walkthrough Studi Kasus Web Forum PKL 60 Politeknik Statistika STIS (Software Usability Testing using Cognitive Walkthrough : A Case Study of PKL 60 Politeknik Statistika STIS Web Forum). Dalam *Seminar Nasional Official Statistics*.
- Qolbiyatul Lina. (2019, Januari 2). *Apa itu Convolutional Neural Network?* <https://medium.com/@16611110/apa-itu-convolutional-neural-network-836f70b193a4#:~:text=Bagaimana%20CNN%20bekerja%3F,tersebut%20dengan%20filter%20yang%20digunakan.>
- Rahmad, F., Suryanto, Y., & Ramli, K. (2020). Performance Comparison of Anti-Spam Technology Using Confusion Matrix Classification. *IOP Conference Series: Materials Science and Engineering*, 879(1). <https://doi.org/10.1088/1757-899X/879/1/012076>
- Rizki, Y. (2012). Markerless Augmented Reality Pada Perangkat Android. *Teknik Elektro. ITS Surabaya*.

- Rizqi Okta Ekoputris. (2018, Mei 9). *MobileNet: Deteksi Objek pada Platform Mobile*. <https://medium.com/nodeflux/mobilenet-deteksi-objek-pada-platform-mobile-bbbf3806e4b3>
- Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). *MobileNetV2: Inverted Residuals and Linear Bottlenecks*.
- Schaeffer, S. E. (2014). *Usability Evaluation for Augmented Reality*.
- Şengel, E. (2013). Usability Level of a University Web Site. *Procedia - Social and Behavioral Sciences*, 106, 3246–3252.
<https://doi.org/10.1016/j.sbspro.2013.12.373>
- Sheikh, S., Heyat, M. B. Bin, AlShorman, O., Masadeh, M., & Alkhatni, F. (2021, Mei 16). A Review of Usability Evaluation Techniques for Augmented Reality Systems in Education. *2021 Innovation and New Trends in Engineering, Technology and Science Education Conference, IETSEC 2021*. <https://doi.org/10.1109/IETSEC51476.2021.9440506>
- Sugiarto, H. (2018). Penerapan Multimedia Development Life Cycle Pada Aplikasi Pengenalan Abjad Dan Angka. *IJCIT (Indonesian Journal on Computer and Information Technology)*, 3(1).
- Wahyudi, N., Harianto, R. A., & Setyati, E. (2019). Augmented Reality Marker Based Tracking Visualisasi Drawing 2D ke dalam Bentuk 3D dengan Metode FAST Corner Detection. *INSYST: Journal of Intelligent System and Computation*, 1(1), 9–18.
- Wharton, C., Rieman, J., Lewis, C., & Polson, P. (1994). The Cognitive Walkthrough Method: A Practitioner's Guide. Dalam *Usability Inspection Methods* (hlm. 105–140). John Wiley & Sons, Inc.
- Widiyanto, A., & Rifa, A. (2014). *User Manual with Augmented Reality to Support Packaging Products*.
- Wu, Q., Liu, Y., Li, Q., Jin, S., & Li, F. (2017). The application of deep learning in computer vision. *2017 Chinese Automation Congress (CAC)*, 6522–6527.
<https://doi.org/10.1109/CAC.2017.8243952>