

**APLIKASI PENERJEMAH BAHASA ISYARAT MENGGUNAKAN  
CONVOLUTIONAL NEURAL NETWORK BERBASIS WEB  
(Studi Kasus: Bangkit Academy)**

**LAPORAN KERJA PRAKTIK**

**Disusun oleh:  
John Alessandro Jong  
21013040**



**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNIK  
UNIVERSITAS KATOLIK DE LA SALLE MANADO  
MANADO  
2024**

**APLIKASI PENERJEMAH BAHASA ISYARAT MENGGUNAKAN  
CONVOLUTIONAL NEURAL NETWORK BERBASIS WEB**

**(Studi Kasus: Bangkit Academy)**

**LAPORAN KERJA PRAKTIK**

Diajukan untuk memenuhi persyaratan Mata Kuliah Kerja Praktik

**Disusun oleh:**

**John Alessandro Jong**

**21013040**



**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNIK  
UNIVERSITAS KATOLIK DE LA SALLE MANADO  
MANADO  
2024**

**LEMBAR PENGESAHAN  
LAPORAN KERJA PRAKTIK**

**Judul:**

**APLIKASI PENERJEMAH BAHASA ISYARAT MENGGUNAKAN  
*CONVOLUTIONAL NEURAL NETWORK* BERBASIS WEB  
(Studi Kasus: Bangkit Academy)**

**Telah disetujui dan disahkan pada tanggal: 6 Januari 2025**

**Bangkit Academy 2024**

**Oleh:**

**Google, GoTo Traveloka – Machine Learning Path**

## LEMBAR PERNYATAAN

Yang bertanda tangan di bawah ini:

Nama : John Alessandro Jong  
NIM : 21013040  
Tempat/Tanggal Lahir : Manado/26 November 2003  
Fakultas/Program Studi : Teknik/Teknik Informatika

Dengan ini, saya menyatakan bahwa laporan Kerja Praktik dan Aplikasi yang berjudul "**Aplikasi Penerjemah Bahasa Isyarat Menggunakan Convolutional Neural Network Berbasis Web (Studi Kasus: Bangkit Academy)**" adalah hasil karya saya sendiri. Seluruh isi laporan ini, kecuali bagian yang mengacu pada teori atau referensi yang telah disebutkan sumbernya, tidak mengambil atau menyalin dari karya orang lain, baik sebagian maupun keseluruhan.

Pernyataan ini saya buat dengan penuh kesadaran dan kejujuran. Apabila di kemudian hari terbukti bahwa pernyataan ini tidak sesuai dengan kenyataan, saya siap menerima sanksi akademis yang diberlakukan oleh pihak Fakultas.

Manado, 20 Desember 2024

Yang Membuat Pernyataan



**John Alessandro Jong**

Menyetujui,

Dosen Pembimbing I

**Indah Yessi Kairupan, S.T., M.Sc.**

Dosen Pembimbing II

**Vivie Deyby Kumenap, S.T., M.Cs.**

Ketua Program Studi

**Vivie Deyby Kumenap, S.T., M.Cs.**

Mengetahui,



**Ronaki A. Hachmadi, S.T., M.T.**



**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNIK  
UNIVERSITAS KATOLIK DE LA SALLE  
MANADO**

**FORM KP - 003**

**FORMULIR DATA UMUM PERUSAHAAN**


NAMA MAHASISWA : John Alessandro Jong  
NIM : 21013040

NAMA PERUSAHAAN : Bangkit Academy 2024 By Google, GoTo,  
Traveloka  
ALAMAT PERUSAHAAN : Dicoding Space, Jalan Batik Kumeli No 50,  
Kecamatan: Cibeunying Kaler, Kelurahan: Sukulayu,  
RT: 10, RW: 07, Kota Bandung, Jawa

DIDIRIKAN TAHUN : 2018  
BIDANG BISNIS : NIB (9120304611285)  
JUMLAH KARYAWAN : Jasa Pendidikan Komputer (Teknologi Informasi)  
PEMILIK : Narenda Wicaksono dan Kevin Kurniawan  
DEWAN DIREKTUR : Dimas Catur Wibowo , Nur Rohman, Ahmad  
Imadduin

WAKIL PERUSAHAAN

Tanggal : 6 Januari 2025  
Nama : Muhammad Amhar Ramyadin  
Jabatan : Mentor

(Tanda tangan dan  
cap perusahaan) : 



**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNIK  
UNIVERSITAS KATOLIK DE LA SALLE  
MANADO**

**FORM KP - 004**

**FORMULIR PENILAIAN KEMAJUAN KERJA PRAKTEK**

**A. UMUM**

Nama Mahasiswa : John Alessandro Jong  
NIM Mahasiswa : 21013040  
Program Studi : Teknik Informatika  
Dosen Pembimbing Akademik : Anthoni Rivai Pulakiang, S.Pd., M.Kom.  
Topik/Rencana Bidang : Aplikasi Penerjemah Bahasa Isyarat  
Menggunakan Convolutional Neural Network Berbasis Web (Studi Kasus: Bangkit Academy)  
Pembimbing 1 : Indah Yessi Kairupan, S.T., M.Sc.  
Pembimbing 2 : Vivie Deyby Kumenap, S.T., M.Cs.  
Terhitung Mulai : 30 Agustus 2024  
Target Selesai : 20 Desember 2024

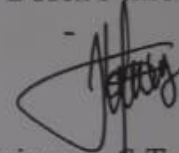
**B. KEGIATAN PELAKSANAAN KERJA PRAKTEK**

No.	Tanggal	Jenis Kegiatan	Paraf Pembimbing
1.	3 September 2024	Konsultasi Proposal KP	
2.	30 September 2024	Konsultasi Proposal KP	
3.	10 Oktober 2024	Revisi Proposal KP	
4.	4 November 2024	Revisi Proposal KP	
5.	6 November 2024	Konsultasi Bab I	
6.	15 November 2024	Konsultasi Bab I dan Bab II	
7.	16 November 2024	Konsultasi Bab I dan Bab II	
8.	16 Desember 2024	Konsultasi Bab I - IV	
9.	17 Desember 2024	Konsultasi Bab I - IV	

10.	17 Desember 2024	Konsultasi Bab I – IV dan Aplikasi	Utah.
11.	18 Desember 2024	Konsultasi Bab III, Bab IV, dan Aplikasi	Utah.
12.	18 Desember 2024	Konsultasi Bab III, Bab IV, dan Aplikasi	Utah.
13.	7 Januari 2025	Konsultasi Bab III, Bab IV, dan Aplikasi	Utah.
14.	8 Januari 2025	Konsultasi Bab III, Bab IV, dan Aplikasi	Utah.
15.	9 Januari 2025	Konsultasi Bab IV dan Aplikasi	Utah.
16.	9 Januari 2025	Konsultasi Bab IV dan Aplikasi	Utah.

Manado, 9 Januari 2025

Dosen Pembimbing



Indah Yessi Kairupan, S.T., M.Sc.



**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNIK  
UNIVERSITAS KATOLIK DE LA SALLE  
MANADO**

**FORM KP - 005**

**FORMULIR PENILAIAN PELAKSANAAN KERJA PRAKTEK**

Mohon diisi dan dicek seperlunya,

NAMA MAHASISWA : John Alessandro Jong  
NIM : 21013040  
NAMA PERUSAHAAN : Bangkit Academy 2024 By Google, GoTo,  
Traveloka  
ALAMAT PERUSAHAAN : Jakarta, Indonesia  
TGL KERJA PRAKTEK : 10 September 2024  
TOPIK YANG DIBAHAS : Aplikasi Penerjemah Bahasa Isyarat  
Menggunakan Convolutional Neural Network  
Berbasis Web (Studi Kasus: Bangkit Academy)

Nilai	=	50	60	70	80	90	100
Sikap							
Kerajinan	=	50	60	70	80	90	100
Prestasi	=	50	60	70	80	90	100

**KOMENTAR/SARAN**

NILAI RATA-RATA : 84,4  
TANGGAL : 6 Januari 2025  
NAMA PENILAI : Muhammad Amhar Ramyadin  
JABATAN : Mentor

(Tanda tangan dan  
cap perusahaan)

:  **bangk!t**

## KATA PENGANTAR

Segala puji dan syukur penulis panjatkan kepada Tuhan Yang Maha Esa atas rahmat dan kasih karunia-Nya, sehingga penulis dapat menyelesaikan Laporan Kerja Praktik di Bangkit Academy dengan judul “Aplikasi Klasifikasi Sampah Menggunakan *Convolutional Neural Network* Berbasis Web (Studi Kasus: Bangkit Academy)” dengan baik dan sesuai waktu yang ditentukan.

Penulis menyadari bahwa terselesaikannya laporan ini tidak lepas dari bantuan, bimbingan, serta dukungan dari berbagai pihak. Oleh karena itu, penulis ingin menyampaikan rasa terima kasih yang sebesar-besarnya kepada:

1. Bapak Prof. Dr. Johanis Ohoitumur selaku Rektor Universitas Katolik De La Salle Manado.
2. Bapak Ronald Albert Rachmadi, S.T., M.T. selaku Dekan Fakultas Teknik.
3. Ibu Vivie Deyby Kumenap, S.T., M.Cs. selaku Ketua Program Studi Teknik Informatika dan Dosen Pembimbing II yang menuntun proses penulisan laporan ini.
4. Bapak Anthoni Rivai Pulakiang, S.Pd., M.Kom. selaku Dosen Pembimbing Akademik yang membantu dalam proses Kerja Praktik.
5. Ibu Indah Yesi Kairupan, S.T., M.Sc. selaku Dosen Pembimbing I, yang telah memberikan masukan dan menuntun dalam proses penulisan laporan ini.
6. Orang tua dan keluarga saya yang senantiasa membantu saya dalam hidup saya.
7. Rekan-rekan Teknik Angkatan 2021 yang telah memberikan dukungan moral dan inspirasi selama proses laporan ini.
8. Rekan-rekan Bangkit yang sangat mendukung proses penyelesaian program Bangkit, kepada Vina, Rama, Kevin, Seta, Kak Kevin, Kak Andre, Tian, Hilal, dan Ghefira.

Penulis menyadari bahwa laporan ini masih jauh dari kesempurnaan. Oleh karena itu, kritik, saran, dan masukan yang membangun dari pembaca sangat diharapkan demi perbaikan di masa yang akan datang. Akhir kata, penulis berharap laporan ini dapat memberikan manfaat bagi pembaca serta menjadi kontribusi yang berarti di bidang teknologi informasi.

Manado, Januari 2025

Penulis

## DAFTAR ISI

HALAMAN JUDUL.....	i
LEMBAR PENGESAHAN .....	ii
LEMBAR PERNYATAAN.....	iii
FORMULIR DATA UMUM PERUSAHAAN (FORM KP – 003).....	iv
FORMULIR PENILAIAN KEMAJUAN PRAKTIK (FORM KP – 004) .....	v
FORMULIR PENILAIAN PELAKSANAAN KERJA PRAKTIK (FORM KP – 005) .....	vii
KATA PENGANTAR .....	viii
DAFTAR ISI.....	ix
DAFTAR TABEL.....	xi
DAFTAR GAMBAR .....	xii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	2
1.3 Tujuan Kerja Praktik .....	2
1.4 Manfaat Kerja Praktik .....	2
1.5 Batasan Masalah.....	3
1.6 Sistematika Penulisan.....	4
BAB II DATA UMUM PERUSAHAAN.....	5
2.1 Sejarah Umum Perusahaan.....	5
2.2 Lingkup Pekerjaan Perusahaan .....	5
2.2.1 Visi .....	5
2.2.2 Misi .....	6
2.2.3 Logo .....	6
2.2.4 Tugas Pokok dan Fungsi Bidang .....	7
2.3 Lingkup Pekerjaan Yang Dilakukan .....	8
BAB III LANDASAN TEORI.....	9
3.1 Teori Pendukung .....	9
3.1.1 Aplikasi .....	9
3.1.2 Web .....	9
3.1.3 Bahasa Isyarat (SIBI dan BISINDO) .....	9
3.1.4 <i>Machine Learning</i> dan <i>Deep Learning</i> .....	10
3.1.5 <i>Convolutional Neural Network</i> .....	11
3.2 Teknologi yang Digunakan .....	15
3.2.1 Python .....	15
3.2.2 TensorFlow .....	17
3.2.3 Google Colab .....	19
3.2.4 Kaggle .....	19
3.3 Kakas Pemodelan .....	20
3.3.1 <i>Flowchart</i> .....	20

3.3.2 <i>Data Flow Diagram</i> .....	21
3.4 Metodologi Pengembangan Sistem .....	22
3.5 Prosedur Pengumpulan Data .....	24
BAB IV PEMBAHASAN.....	25
4.1 Analisis .....	25
4.1.1 Pengumpulan dan Pengolahan Data.....	25
4.1.2 Analisis Hasil Pengumpulan dan Pengolahan Data .....	26
4.1.3 Identifikasi Masalah dan Kesempatan .....	26
4.1.4 Identifikasi Target Pengguna .....	27
4.1.5 Identifikasi Sumber Daya.....	28
4.1.6 Daftar Spesifikasi Persyaratan .....	29
4.2 Desain .....	30
4.2.1 <i>Flowchart</i> .....	30
4.2.2 <i>Data Flow Diagram</i> .....	31
4.2.3 <i>Storyboard</i> .....	32
4.3 Implementasi .....	35
4.3.1 Implementasi Tampilan Antarmuka.....	36
4.3.2 Implementasi Pemrograman.....	40
4.4 Pengujian .....	46
4.4.1 Tujuan Pengujian .....	46
4.4.2 Kriteria Pengujian .....	47
4.4.3 Kasus Pengujian .....	47
4.4.4 Pelaksanaan Pengujian .....	47
BAB V KESIMPULAN DAN SARAN.....	59
5.1 Kesimpulan.....	59
5.2 Saran.....	59
DAFTAR PUSTAKA .....	60
LAMPIRAN	

## DAFTAR TABEL

Tabel 3.1 Daftar <i>Library</i> Python.....	16
Tabel 3.2 Daftar <i>Library</i> TensorFlow .....	18
Tabel 3.3 Simbol <i>Flowchart</i> .....	20
Tabel 3.4 Simbol <i>Data Flow Diagram</i> .....	21
Tabel 4.1 Daftar Masalah dan Kesempatan .....	27
Tabel 4.2 Daftar Pengguna.....	28
Tabel 4.3 Daftar Pengembang Aplikasi .....	28
Tabel 4.4 Daftar Perangkat Keras .....	28
Tabel 4.5 Daftar Perangkat Lunak .....	29
Tabel 4.6 Kode Model SIBI.....	40
Tabel 4.7 Kode Model BISINDO .....	43
Tabel 4.8 Hasil Pengujian Model BISINDO .....	56

## DAFTAR GAMBAR

Gambar 2.1	Logo Kemendikbud [13] .....	6
Gambar 3.1	Arsitektur VGG16 [25] .....	12
Gambar 3.2	Metodologi <i>Waterfall</i> [42] .....	22
Gambar 4.1	<i>Flowchart</i> Autentikasi.....	30
Gambar 4.2	<i>Flowchart</i> Terjemah Bahasa Isyarat .....	31
Gambar 4.3	DFD Level 0.....	31
Gambar 4.4	DFD Level 1 .....	32
Gambar 4.5	<i>Storyboard</i> Beranda.....	33
Gambar 4.6	<i>Storyboard</i> Register.....	33
Gambar 4.7	<i>Storyboard</i> Login.....	34
Gambar 4.8	<i>Storyboard</i> Penerjemahan .....	35
Gambar 4.9	Halaman <i>Login</i> .....	36
Gambar 4.10	Halaman Register .....	36
Gambar 4.11	Halaman Beranda .....	37
Gambar 4.12	Halaman Terjemah SIBI.....	37
Gambar 4.13	Halaman Terjemah BISINDO .....	38
Gambar 4.14	Halaman <i>Loading</i> Terjemahan .....	38
Gambar 4.15	Tampilan Hasil Terjemahan .....	39
Gambar 4.16	Halaman <i>About Us</i> .....	39
Gambar 4.17	Halaman Beranda pada Microsoft Edge.....	48
Gambar 4.18	Hasil Terjemahan pada Microsoft Edge.....	48
Gambar 4.19	Halaman Beranda pada Mozilla Firefox .....	49
Gambar 4.20	Tampilan Hasil Terjemahan pada Mozilla Firefox .....	49
Gambar 4.21	Halaman Beranda pada Google Chrome.....	50
Gambar 4.22	Tampilan Hasil Terjemahan pada Google Chrome.....	50
Gambar 4.23	Mengakses Halaman <i>Login</i> .....	51
Gambar 4.24	Mengisi <i>Field</i> Untuk Register .....	51
Gambar 4.25	Berhasil Register Akun .....	51
Gambar 4.26	Mengisi <i>Field</i> Login.....	52
Gambar 4.27	Tampilan Setelah Berhasil <i>Login</i> .....	52
Gambar 4.28	Akurasi Model SIBI .....	53
Gambar 4.29	<i>Loss</i> Model SIBI .....	53
Gambar 4.30	Akurasi Model BISINDO .....	54
Gambar 4.31	<i>Loss</i> Model BISINDO .....	54
Gambar 4.32	Hasil Pengujian Model SIBI .....	55

## DAFTAR LAMPIRAN

Lampiran A <i>User Acceptance Test</i> .....	A-1
--	-----

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Tuli dan hilang pendengaran adalah kondisi yang mempengaruhi kemampuan seseorang dalam mendengar. Kondisi ini tersebar di seluruh dunia dan bisa ditemukan pada setiap wilayah dan populasi. Menurut *World Health Organization* (WHO), pada tahun 2024 sekitar 20% populasi global atau sekitar 1,5 miliar orang mengalami masalah hilang pendengaran dan 430 juta di antaranya mengalami gangguan yang mengharuskan mereka memakai alat bantu dengar dan berkomunikasi dengan bahasa isyarat [1]. Menurut BPS, di Indonesia, terdapat hampir 5 juta orang yang mengalami kesulitan mendengar pada berbagai level [2]. Orang dengan kondisi ini di Indonesia, memilih dikenal dengan istilah Orang Tuli sebagai identitas diri [3].

Bahasa isyarat merupakan alat komunikasi utama bagi sebagian besar orang dengan gangguan pendengaran. Di banyak negara, bahasa isyarat digunakan sebagai bahasa ibu oleh komunitas Tuli, dan menjadi media penting dalam kehidupan sosial mereka [4]. Berbeda dengan teknologi seperti alat bantu dengar, teks tertulis, implan koklea, dan perangkat lainnya, bahasa isyarat lebih mudah diakses oleh masyarakat dari kalangan mana pun [5]. Meskipun demikian, masih ada kesenjangan komunikasi antara individu dengan gangguan pendengaran dan masyarakat umum yang tidak familier dengan bahasa isyarat. Hal ini dapat menghambat interaksi sosial, pendidikan, hingga akses terhadap layanan kesehatan. Terutama dengan kurang dan tidak adanya kesadaran akan budaya dan gaya hidup komunitas Tuli di Indonesia. Salah satu solusi yang paling memungkinkan untuk diimplementasikan adalah penggunaan teknologi, khususnya dalam penggunaan teknologi berbasis pembelajaran mesin atau *machine learning*. Teknologi ini memungkinkan pengembangan sistem yang dapat mengenali bahasa isyarat, sehingga membantu individu non-Tuli untuk lebih mudah berkomunikasi dengan komunitas Tuli.

Pembelajaran mesin atau dikenal dengan *machine learning* merupakan cabang dari kecerdasan buatan yang memungkinkan komputer untuk belajar

menghasilkan aturan melalui data *input* dan data luaran. Kemajuan teknologi memungkinkan integrasi teknologi seperti kamera dengan model *image recognition*. Dalam konteks deteksi bahasa isyarat, algoritma pembelajaran mesin dilatih menggunakan data gambar atau video dalam jumlah besar untuk huruf isyarat [6]. Dengan mempelajari pola dan karakteristik gambar tersebut, model pembelajaran mesin dapat mengenali gambar baru dan menerjemahkan gambar tersebut dengan makna tertentu [7]. Pendekatan yang umum digunakan untuk *image recognition* seperti ini adalah *deep learning*, terutama dengan memanfaatkan *Convolutional Neural Networks* atau CNNs. CNNs sangat efektif untuk mengidentifikasi dan mengklasifikasikan objek visual, sehingga cocok untuk mengenali bentuk dan posisi tangan dalam berbagai kondisi [8].

Berdasarkan penjelasan di atas, maka akan dibangun aplikasi penerjemah bahasa isyarat berbasis web yang bisa menerjemahkan 24 alfabet SIBI (Sistem Isyarat Bahasa Indonesia) dan 26 alfabet BISINDO (Bahasa Isyarat Indonesia) dari *input* gambar dan video. Proses kerja umumnya dimulai dengan mengumpulkan *dataset* gambar alfabet bahasa isyarat dalam berbagai posisi dan situasi. Data ini kemudian akan diproses dan digunakan dalam pelatihan model pembelajaran mesin sehingga dapat mengenali pola yang beragam dalam gambar alfabet bahasa isyarat.

## **1.2 Rumusan Masalah**

Bagaimana merancang dan membangun aplikasi penerjemah bahasa isyarat berbasis web menggunakan *Convolutional Neural Network* yang bisa menjadi media komunikasi dan edukasi bahasa isyarat?

## **1.3 Tujuan Kerja Praktik**

Merancang dan membangun aplikasi penerjemah bahasa isyarat berbasis web menggunakan *Convolutional Neural Network* yang bisa menjadi media komunikasi dan edukasi bahasa isyarat.

## **1.4 Manfaat Kerja Praktik**

A. Manfaat untuk masyarakat

1. Aplikasi penerjemah bahasa isyarat bisa membantu komunitas Tuli untuk berkomunikasi dengan non-Tuli, memberikan kemudahan dan pemahaman antara kedua pihak.
2. Aplikasi ini berpotensi menjadi alternatif media komunikasi antara kaum Tuli dan non-Tuli.
3. Aplikasi ini bisa membantu non-Tuli belajar bahasa isyarat dengan mengetahui isyarat yang tepat melalui hasil terjemahan.
4. Aplikasi penerjemah bahasa isyarat dapat memperluas kemampuan komunikasi bagi orang yang berinteraksi dengan orang Tuli, terutama bagi mereka yang belum familiar dengan bahasa isyarat.
5. Aplikasi ini juga dapat meningkatkan pemahaman non-Tuli terhadap budaya dan cara hidup komunitas tunarungu, menciptakan rasa saling menghargai dan empati antar kedua belah pihak.

#### B. Manfaat untuk penulis/mahasiswa

1. Pembangunan aplikasi ini dapat menjadi nilai yang baik untuk mahasiswa dalam memenuhi Kerja Praktik (KP).
2. Mahasiswa mendapatkan pengalaman dalam mengimplementasikan model *machine learning* dalam aplikasi pengenalan bahasa isyarat.
3. Mahasiswa mendapatkan pengalaman berkolaborasi dengan pengembang dari *path-path* berbeda, yaitu *cloud computing*, dan *mobile development* untuk menyusun aplikasi kompleks.

### 1.5 Batasan Masalah

1. Pembangunan model menggunakan TensorFlow dan memanfaatkan CNN untuk klasifikasi gambar alfabet.
2. Pembangunan model *machine learning* untuk menerjemahkan 24 alfabet SIBI, mengecualikan alfabet J dan Z yang merupakan alfabet bergerak (gestur).
3. Pembangunan model *machine learning* untuk menerjemahkan 26 alfabet BISINDO A sampai Z.
4. Aplikasi ini memungkinkan pengguna untuk mengunggah gambar atau mengunggah video untuk diterjemahkan oleh model *machine learning*.

## **1.6 Sistematika Penulisan**

Berikut adalah susunan penulisan laporan kerja praktik ini.

### **BAB I: Pendahuluan**

Bab ini menguraikan latar belakang pengembangan aplikasi yang dibuat. Di dalamnya terdapat rumusan masalah, tujuan, serta manfaat dari pengembangan aplikasi. Selanjutnya, dijelaskan mengenai batasan masalah dan juga sistematika penulisan laporan terkait pembangunan aplikasi ini.

### **BAB II: Data Umum Perusahaan**

Bab ini memaparkan informasi umum mengenai perusahaan yang menjadi lokasi pengembangan aplikasi. Di dalamnya terdapat sejarah perusahaan, ruang lingkup pekerjaan yang dijalankan oleh perusahaan, serta ruang lingkup pekerjaan yang dilakukan oleh perusahaan tersebut.

### **BAB III: Landasan Teori**

Bab ini menguraikan teori-teori yang mendasari pengembangan aplikasi yang dibuat. Selanjutnya, dibahas mengenai teknologi yang digunakan dalam pengembangan perangkat lunak, metodologi pengembangan, serta prosedur pengumpulan dan pengolahan data.

### **BAB IV: Pembahasan**

Bab ini menjelaskan empat bagian pembahasan dari pembangunan aplikasi yang dibangun. Bagian pertama menjelaskan analisis, bagian kedua adalah bagian desain, bagian ketiga adalah bagian implementasi, dan bagian keempat dan terakhir dari pembahasan adalah pengujian.

### **BAB V: Kesimpulan dan Saran**

Bab ini menjelaskan kesimpulan dari seluruh proses pembangunan aplikasi dan saran untuk pembangunan atau penelitian kedepannya.

## **BAB II**

### **DATA UMUM PERUSAHAAN**

#### **2.1 Sejarah Umum Perusahaan**

Kampus Merdeka merupakan salah satu kebijakan Merdeka Belajar oleh Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi (Kemendikbudristek) [9]. Kemendikbudristek bermula sebagai Kementerian Pengajaran yang didirikan karena semangat membangun identitas bangsa pada awal kemerdekaan. Menteri Pengajaran yang pertama adalah Bapak Pendidikan Nasional, Ki Hadjar Dewantara. Sejak awal, badan ini sudah ada kesadaran untuk menyiapkan kurikulum dan mencerdaskan kehidupan bangsa. Terdapat beberapa iterasi perubahan organisasi dengan penggabungan Kementerian Riset dan Teknologi ke Kementerian Pendidikan dan Kebudayaan pada tahun 2021, membentuk Kemendikbudristek. Menteri Nadiem Anwar Makarim memimpin Kemendikbudristek pada 28 April 2021 [10].

Kampus Merdeka memiliki program Magang dan Studi Independen Bersertifikat (MSIB). Program MSIB merupakan program persiapan karier komprehensif dalam bentuk magang dan studi independen dengan jaminan konversi SKS. Program MSIB melakukan kerja sama dengan banyak mitra yang siap mendidik dan membentuk kemampuan mahasiswa. Mitra-mitra tersebut merupakan organisasi atau perusahaan dari bidang sektor usaha yang terdapat pada IDX Industrial Classification [11]. Salah satu mitra program MSIB adalah Bangkit Academy. Bangkit Academy merupakan program persiapan karir yang dipimpin oleh Google dan didukung oleh mitra industri seperti Gojek, Tokopedia, dan Traveloka. Bangkit menawarkan tiga pilihan jalur pembelajaran atau *learning path*, yaitu *Machine Learning*, *Cloud Computing*, dan *Mobile Development*.

#### **2.2 Lingkup Pekerjaan Perusahaan**

Lingkup pekerjaan dari Kemendikbudristek adalah sebagai berikut.

##### **2.2.1 Visi**

Kemendikbudristek mendukung Visi dan Misi Presiden untuk mewujudkan Indonesia Maju yang berdaulat, mandiri, dan berkepribadian melalui terciptanya

Pelajar Pancasila yang bernalar kritis, kreatif, mandiri, beriman, bertakwa kepada Tuhan Yang Maha Esa, dan berakhlak mulia, bergotong royong, dan berkebinekaan global [12].

### 2.2.2 Misi

Berikut adalah misi-misi dari Kemendikbudristek [12]:

1. Mewujudkan pendidikan yang relevan dan berkualitas tinggi, merata dan berkelanjutan, didukung oleh infrastruktur dan teknologi.
2. Mewujudkan pelestarian dan pemajuan kebudayaan serta pengembangan bahasa dan sastra.
3. Mengoptimalkan peran serta seluruh pemangku kepentingan untuk mendukung transformasi dan reformasi pengelolaan pendidikan dan kebudayaan.

### 2.2.3 Logo



Gambar 2.1 Logo Kemendikbud [13]

Uraian lambang [13]:

1. Bidang Segi Lima (Biru Muda) menggambarkan alam kehidupan Pancasila.
2. Semboyan Tut Wuri Handayani digunakan oleh Ki Hajar Dewantara dalam melaksanakan sistem pendidikannya. Pencantuman semboyan ini berarti melengkapi penghargaan dan penghormatan kita terhadap almarhum Ki Hajar Dewantara yang hari lahirnya telah dijadikan Hari Pendidikan Nasional.
3. Belencong Menyala Bermotif Garuda Belencong (menyala) merupakan lampu yang khusus dipergunakan pada pertunjukan wayang kulit. Cahaya belencong membuat pertunjukan menjadi hidup.
4. Burung Garuda (yang menjadi motif belencong) memberikan gambaran sifat dinamis, gagah perkasa, mampu dan berani mandiri mengarungi angkasa luas.

Ekor dan sayap garuda digambarkan masing-masing lima, yang berarti: ‘satu kata dengan perbuatan Pancasila’.

5. Buku merupakan sumber bagi segala ilmu yang dapat bermanfaat bagi kehidupan manusia.
6. Warna putih pada ekor dan sayap garuda dan buku berarti suci, bersih tanpa pamrih. Warna kuning emas pada nyala api berarti keagungan dan keluhuran pengabdian. Warna biru muda pada bidang segi lima berarti pengabdian yang tak kunjung putus dengan memiliki pandangan hidup yang mendalam (pandangan hidup Pancasila)

#### **2.2.4 Tugas Pokok dan Fungsi Bidang**

Kemendikbudristek mempunyai tugas menyelenggarakan urusan pemerintahan di bidang pendidikan, kebudayaan, ilmu pengetahuan, dan teknologi untuk membantu Presiden dalam menyelenggarakan pemerintahan negara. Berikut merupakan fungsi dari Kemendikbudristek [14]:

1. Perumusan dan penetapan kebijakan di bidang pendidikan anak usia dini, pendidikan dasar, pendidikan menengah, dan pendidikan masyarakat, serta pengelolaan kebudayaan;
2. Pelaksanaan fasilitasi penyelenggaraan pendidikan anak usia dini, pendidikan dasar, pendidikan menengah, dan pendidikan masyarakat, serta pengelolaan kebudayaan;
3. Pelaksanaan kebijakan di bidang peningkatan mutu dan kesejahteraan guru dan pendidik lainnya, serta tenaga kependidikan;
4. Koordinasi pelaksanaan tugas, pembinaan, dan pemberian dukungan administrasi kepada seluruh unsur organisasi di lingkungan Kementerian Pendidikan dan Kebudayaan;
5. Pengelolaan barang milik/kekayaan negara yang menjadi tanggung jawab Kementerian Pendidikan dan Kebudayaan;
6. Pengawasan atas pelaksanaan tugas di lingkungan Kementerian Pendidikan dan Kebudayaan;
7. Pelaksanaan bimbingan teknis dan supervisi atas pelaksanaan urusan Kementerian Pendidikan dan Kebudayaan di daerah;

8. Pelaksanaan pengembangan, pembinaan, dan perlindungan bahasa dan sastra;
9. Pelaksanaan penelitian dan pengembangan di bidang pendidikan anak usia dini, pendidikan dasar, pendidikan menengah, dan pendidikan masyarakat, serta kebudayaan; dan
10. Pelaksanaan dukungan substantif kepada seluruh unsur organisasi di lingkungan Kementerian Pendidikan dan Kebudayaan.

### 2.3 Lingkup Pekerjaan Yang Dilakukan

Sebagai salah satu mahasiswa yang mendaftar dan lolos seleksi mitra Bangkit Academy dalam *path Machine Learning*, maka mahasiswa menjadi *Machine Learning Cohort*. Tugas dan tanggung jawab *Machine Learning Cohort* adalah mengikuti kursus Dicoding dan Coursera, mengikuti pembelajaran *Instructor Led Training (ILT) Machine Learning* dan *Soft Skills*, dan mengikuti semua kegiatan wajib lainnya dari Bangkit Academy. Pada puncaknya, *Machine Learning Cohort* harus menuntaskan proyek *Capstone* kolaborasi tiga *learning path*. Proyek *Capstone* berlangsung selama satu bulan lebih, dari 4 November sampai 13 Desember. Penyelenggaraan keseluruhan program Bangkit sendiri berjalan dari bulan September sampai dengan bulan Desember (4 bulan).

Pekerjaan seorang *Machine Learning Cohort* pada *Capstone* ini adalah melakukan pencarian dan pembersihan *dataset* yang relevan, melakukan optimasi pada *dataset*, kemudian melakukan pelatihan dan *tuning* model *machine learning* sehingga mendapat model dengan akurasi *training* dan validasi yang cukup untuk melakukan klasifikasi. Setelah model yang baik dibentuk maka akan dilakukan pengujian model, dan pada akhirnya model yang sudah dilatih akan diberikan kepada *Cloud Computing Cohort* dan *Mobile Development Cohort*. Penulis juga mendapat tugas tambahan menjadi *team lead* yang bertanggung jawab memimpin dan melakukan *project management* berupa dokumentasi proyek.

## **BAB III**

### **LANDASAN TEORI**

#### **3.1 Teori Pendukung**

Teori pendukung merujuk pada dasar atau landasan teoritis yang menjelaskan konsep, teknologi, atau metodologi yang diterapkan dalam pembangunan aplikasi ini. Teori ini berfungsi untuk memberikan konteks dan dasar yang kuat bagi penerapan solusi teknis, serta membantu dalam memahami relevansi dan penerapan berbagai konsep yang digunakan.

##### **3.1.1 Aplikasi**

Definisi aplikasi dalam konteks teknologi informasi, adalah perangkat lunak yang dirancang untuk menjalankan tugas tertentu sesuai kebutuhan pengguna. Jenis aplikasi ini sangat beragam, mencakup aplikasi *mobile*, aplikasi berbasis web, aplikasi desktop, hingga aplikasi berbasis *cloud*. Secara umum, aplikasi dapat diartikan sebagai program yang siap digunakan untuk menjalankan perintah dari pengguna, dengan tujuan memberikan hasil yang lebih efisien dan akurat sesuai kebutuhan spesifik [15].

##### **3.1.2 Web**

Web adalah halaman-halaman berisi data dan informasi yang merupakan bagian dari internet. Web merupakan nama umum dari *World Wide Web* dan pada dasarnya, web adalah sistem server internet yang mendukung dokumen terformat khusus. HTML (*Hypertext Markup Language*) merupakan *markup language* yang mendukung tautan dokumen. Untuk mengakses halaman atau dokumen web digunakan *web browser* seperti Internet Explorer, Google Chrome, Mozilla Firefox, dan lainnya [16].

##### **3.1.3 Bahasa Isyarat (SIBI dan BISINDO)**

Bahasa isyarat merupakan bahasa visual yang menggunakan tangan, gerakan tubuh, dan ekspresi untuk berkomunikasi. Bahasa isyarat digunakan oleh orang-orang Tuli dan orang yang memiliki gangguan pendengaran.

Di Indonesia sendiri terdapat dua bahasa isyarat, yaitu Sistem Isyarat Bahasa Indonesia (SIBI) dan Bahasa Isyarat Indonesia (BISINDO). SIBI merupakan bahasa isyarat terstruktur dan formal yang disusun oleh pemerintah dan memiliki tata aturan bahasa yang mendekati Bahasa Indonesia. SIBI mengikuti aturan seperti subjek, predikat, dan objek sehingga lebih kaku dibandingkan dengan BISINDO. BISINDO merupakan bahasa isyarat alami yang berkembang dengan sendirinya di kalangan komunitas Tuli di Indonesia. BISINDO sangat fleksibel dan mengikuti pola komunikasi spontan. Setiap daerah di Indonesia memiliki variasi yang berbeda, namun strukturnya serupa sehingga bisa dipahami [17].

### **3.1.4 Machine Learning dan Deep Learning**

*Machine Learning* (ML) adalah konsep pembelajaran mesin dimana mesin yang dikembangkan belajar untuk melakukan tugas sendiri tanpa perintah manusia. ML merupakan gabungan konsep seperti matematika, statistika, dan *data mining*. Berbeda dengan pemrograman tradisional dimana manusia memberikan *input* berupa data dan aturan program sehingga program memberikan *output*, pemrograman ML adalah dimana manusia memberikan *input* data dan *output* sehingga program menghasilkan aturan. Aturan ini lebih dikenal dengan model [18].

Di sisi lain, *Deep Learning* (DL) adalah sub-bidang dari ML yang mengandalkan jaringan saraf tiruan dengan banyak lapisan (*deep neural networks*). Keistimewaan dari *deep learning* adalah kemampuannya untuk memproses dan memahami data yang lebih kompleks dan tidak terstruktur, seperti gambar atau suara, dengan melakukan ekstraksi fitur otomatis. Meskipun *deep learning* membutuhkan lebih banyak data dan daya komputasi, *deep learning* dapat menangani masalah yang jauh lebih kompleks dan menghasilkan performa yang lebih baik dalam berbagai aplikasi, seperti pengenalan gambar, suara, dan pemrosesan bahasa alami [19].

#### **3.1.4.1 Data**

Menurut Kamus Besar Bahasa Indonesia (KBBI), data adalah kumpulan fakta atau informasi yang diperoleh melalui berbagai cara, berbentuk teks, angka,

atau gambar. Data dapat diolah lebih lanjut sehingga menghasilkan suatu informasi. Berdasarkan cara mendapatkannya, data dibagi menjadi primer, yaitu data yang diperoleh secara langsung dan data sekunder, yaitu data yang diperoleh dari sumber yang sudah ada sebelumnya. Berdasarkan sifat data, data terbagi menjadi dua, yaitu kualitatif dan kuantitatif. Data kualitatif bersifat dapat dihitung dan berbentuk angka seperti data umur, data berat dan tinggi badan, sedangkan data kuantitatif bersifat deskriptif dan tidak berbentuk angka seperti kepuasan pelanggan dan kualitas suatu layanan [20].

#### **3.1.4.2 Model**

Model *machine learning* atau ML adalah objek yang sudah dilatih untuk mengenal berbagai tipe pola. Model ML dilatih melalui kumpulan data yang sudah dipilih dan dibersihkan kemudian dilatih untuk mengenal apa yang harus dilakukan dengan data tersebut. Setelah melatih model ML, model tersebut disimpan dan dapat digunakan untuk melakukan prediksi [21].

#### **3.1.4.3 Transfer Learning**

*Transfer learning* merupakan metode dalam *machine learning* di mana pengetahuan model diperoleh dari pelatihan yang sudah dilakukan sebelumnya dengan suatu data sehingga modelnya sudah memiliki pengetahuan yang bisa dipakai untuk meningkatkan performa model. Secara teknik, *transfer learning* menggunakan model dengan bobot di satu proyek yang sudah pernah dilatih ke proyek lain. Teknik *transfer learning* paling banyak ditemukan pada proyek *deep learning* yang memerlukan *deep neural networks* [22] [23].

#### **3.1.5 Convolutional Neural Network**

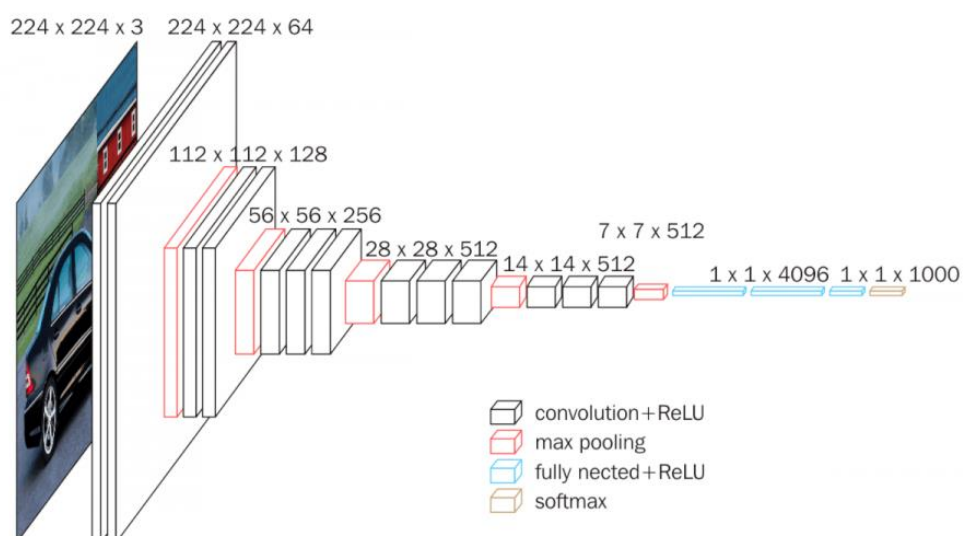
*Neural networks* atau jaringan saraf adalah sub konsep dari *machine learning* dan merupakan jantung algoritma *deep learning*. *Neural networks* terdiri dari lapisan *node* yang terdiri dari lapisan *input*, satu atau lebih lapisan *hidden*, dan lapisan *output*. Setiap *node* terhubung satu sama lain seperti jaringan dan mempunyai bobot dan ambang batas (*threshold*). Jika *output* pada *node* melebihi

batas nilai ambang batas yang ditentukan, *node* tersebut akan mengirim data ke lapisan berikutnya dalam jaringan.

*Convolutional Neural Network* (CNN) merupakan jenis jaringan saraf yang biasa digunakan pada klasifikasi dan *computer vision*. Pendekatan CNN adalah perkalian matriks dari prinsip aljabar linear untuk mengidentifikasi pola dalam gambar. Sebelumnya, proses ekstraksi fitur dilakukan secara manual sehingga memakan waktu untuk mengidentifikasi pola dalam gambar. CNN memberikan pendekatan yang lebih mudah dan skalabel untuk ekstraksi fitur dalam gambar. Namun, CNN memiliki kekurangan dimana CNN cukup berat dalam segi komputasi dan memerlukan *Graphics Processing Unit* (GPU) untuk dijalankan. CNN memiliki tiga jenis lapisan utama, yaitu *Convolutional layer*, *Pooling layer*, dan *Fully Connected layer* (FC). Lapisan pertama adalah *Convolutional layer* dan lapisan terakhir adalah *Fully Connected layer* [8].

### 3.1.5.1 VGG16

VGG16 adalah salah satu model *pre-trained* CNN yang cukup terkenal dan dikembangkan oleh Visual Geometry Group (VGG) di Universitas Oxford. Model ini memiliki 16 lapisan dan digunakan untuk tugas pengenalan gambar. VGG16 dikenal dengan arsitektur yang relatif sederhana tetapi efektif, menggunakan filter konvolusi 3x3 untuk mengenali pola secara bertahap di berbagai tingkat kompleksitas [24].



Gambar 3.1 Arsitektur VGG16 [25]

### 3.1.5.2 Lapisan CNN

Lapisan CNN adalah komponen kunci yang mengatur bagaimana data gambar diproses dalam jaringan saraf. Setiap lapisan dalam CNN berfungsi untuk mengekstraksi fitur tertentu dari gambar, dimulai dari fitur dasar seperti tepi atau sudut pada lapisan awal, hingga fitur yang lebih kompleks di lapisan yang lebih dalam [26].

*Convolutional Layer* adalah komponen inti dari CNN yang berfungsi untuk mengekstraksi fitur atau pola dari data *input*, terutama dalam pengolahan citra. Pada lapisan ini, sebuah filter atau *kernel* (biasanya berupa matriks kecil) diaplikasikan pada gambar *input* dengan melakukan operasi konvolusi, yaitu perkalian elemen per elemen antara filter dan bagian *input* gambar. Hasilnya adalah peta fitur yang menyoroti pola-pola spesifik, seperti tepi, sudut, atau tekstur yang ada dalam gambar. Proses konvolusi ini memungkinkan model untuk mengenali struktur dasar dalam gambar, yang akan diproses lebih lanjut di lapisan-lapisan berikutnya [27].

*Max Pooling Layer* adalah lapisan dalam CNN yang bertujuan untuk mengurangi dimensi peta fitur yang dihasilkan oleh *Convolutional Layer* sambil mempertahankan informasi yang penting. Proses *pooling* dilakukan dengan membagi peta fitur menjadi wilayah-wilayah kecil dan memilih nilai maksimum dari setiap wilayah tersebut. Hal ini membantu mengurangi ukuran data secara keseluruhan, mengurangi beban komputasi, dan mencegah *overfitting* [27].

*Dropout Layer* adalah teknik regularisasi yang digunakan untuk mencegah *overfitting* dalam model *neural network*, termasuk CNN. Pada lapisan ini, sejumlah neuron secara acak akan 'dijatuhkan' (*drop-out*) selama proses pelatihan dengan probabilitas yang ditentukan, misalnya 20% atau 50%. Hal ini mencegah neuron-neuron tersebut untuk beradaptasi terlalu erat dengan data pelatihan dan membantu model untuk belajar representasi yang lebih umum, bukan hanya menyesuaikan diri dengan pola spesifik dalam data pelatihan. *Dropout* dapat meningkatkan generalisasi model dan memberikan performa yang lebih baik pada data yang belum pernah dilihat sebelumnya [27].

*Flatten Layer* adalah lapisan yang digunakan untuk mengubah data yang berbentuk multidimensi, seperti peta fitur dari lapisan konvolusi atau *pooling*, menjadi vektor satu dimensi yang dapat diteruskan ke lapisan *Dense (Fully*

*Connected Layer*). Tanpa *flattening*, data yang berbentuk matriks atau tensor tidak dapat diproses lebih lanjut dalam lapisan *Dense*, yang mengharuskan *input* berbentuk vektor satu dimensi. *Flattening* mengubah data dari bentuk menjadi satu baris panjang yang memudahkan pemrosesan dan pengolahan lebih lanjut dalam jaringan saraf [27].

*Dense Layer*, atau biasa disebut *Fully Connected Layer (FC)*, adalah lapisan dalam jaringan neural di mana setiap neuron terhubung dengan semua neuron di lapisan sebelumnya. Lapisan ini bertanggung jawab untuk melakukan komputasi akhir berdasarkan pengetahuan yang telah diproses pada *layer-layer* sebelumnya. Neuron dalam *Dense Layer* melakukan operasi linear pada *input* yang diberikan dan menghasilkan *output* yang akan diteruskan ke lapisan berikutnya atau digunakan untuk prediksi akhir. *Dense Layer* umumnya digunakan di bagian akhir dari CNN untuk menghasilkan keputusan akhir atau pengklasifikasian berdasarkan fitur yang telah diekstraksi dan diproses oleh lapisan-lapisan konvolusi dan *pooling* [27].

*Softmax Layer* adalah lapisan yang digunakan pada *output* akhir dari model *neural network* untuk tugas klasifikasi multi-kelas. Fungsi aktivasi softmax adalah mengubah *output* dari lapisan *Dense* menjadi probabilitas, di mana setiap nilai *output* dipetakan ke dalam rentang  $[0, 1]$  dan jumlah seluruh probabilitas adalah 1. Fungsi ini memungkinkan model untuk memberikan prediksi yang lebih mudah dimengerti, yaitu dengan mengidentifikasi kelas yang memiliki probabilitas tertinggi sebagai prediksi akhir. *Softmax* sangat berguna dalam klasifikasi gambar, teks, dan aplikasi lainnya di mana *output* terdiri dari beberapa kategori yang saling eksklusif [27].

### 3.1.5.3 Optimizer

*Optimizer* bertanggung jawab untuk memperbaharui bobot model agar dapat meminimalkan *loss function*. Dalam konteks CNN, *optimizer* seperti SGD (*Stochastic Gradient Descent*), Adam, atau RMSprop digunakan untuk memastikan bahwa pembaruan bobot dilakukan secara efisien dan model dapat belajar lebih cepat dan akurat. Pemilihan *optimizer* yang tepat dapat mempengaruhi kecepatan konvergensi dan kemampuan model untuk menghindari *local minima* [26].

### 3.1.5.4 Loss Function

*Loss Function* merupakan fungsi yang digunakan untuk mengukur seberapa baik atau buruk performa model dalam menghasilkan prediksi. Dalam klasifikasi gambar, *loss function* yang sering digunakan adalah *categorical crossentropy*, yang mengukur perbedaan antara distribusi probabilitas yang diprediksi oleh model dan distribusi target yang benar [26].

$$L = - \sum_{i=0}^N y_i \log(pi) \dots \dots \dots \text{(Rumus 3.1)}$$

### 3.1.5.5 Metrik Akurasi

Akurasi adalah metrik penilaian performa model yang digunakan untuk menilai seberapa baik kinerja model dalam membuat prediksi yang benar. Secara sederhana, akurasi merupakan ukuran yang menggambarkan proporsi prediksi yang benar dibandingkan dengan jumlah total prediksi yang dibuat model [28].

$$\text{Accuracy} = \frac{\text{correct classifications}}{\text{total classifications}} = \frac{TP+TN}{TP+TN+FP+FN} \dots \dots \dots \text{(Rumus 3.2)}$$

## 3.2 Teknologi yang Digunakan

Teknologi yang digunakan dalam membangun model *machine learning* pada aplikasi penerjemah bahasa isyarat adalah Python sebagai bahasa pemrograman, Google Colaboratory sebagai lingkungan pengembangan, Kaggle sebagai sumber *dataset*, dan berbagai *library* yang mendukung pembangunan model, seperti TensorFlow, Keras, Scikit-Learn, Numpy, Matplotlib, OpenCV, TensorFlow.js, dan TensorFlow Lite.

### 3.2.1 Python

Python merupakan bahasa pemrograman *high-level* yang mudah dipelajari dan diinterpretasikan. Dengan struktur data yang kuat, pengetikan yang fleksibel, dan pengikatan yang dinamis, Python cocok untuk pengembangan aplikasi *rapid* (*Rapid Application Development*) dan sebagai bahasa skrip untuk menghubungkan komponen-komponen yang ada. Sintaksisnya yang sederhana memudahkan pembacaan kode, sehingga berpotensi untuk mengurangi *cost maintenance*. Python

juga mendukung penggunaan modul dan paket, yang memfasilitasi modularisasi dan *reusability* kode. Interpreter Python dan *library* standar yang lengkap tersedia secara gratis dalam bentuk *source* atau *binary* untuk semua platform utama dan dapat didistribusikan secara bebas [29].

**Tabel 3.1 Daftar *Library* Python**

<b><i>Library</i> Python</b>	<b>Penjelasan</b>
Scikit-Learn	Scikit-Learn adalah <i>library</i> Python yang memudahkan penerapan algoritma <i>machine learning</i> , baik <i>supervised</i> maupun <i>unsupervised</i> . Scikit-Learn dibangun di atas <i>library</i> lain seperti NumPy, pandas, dan Matplotlib. Scikit-Learn menyediakan berbagai fungsionalitas, termasuk regresi, klasifikasi serta kluster. Selain itu, <i>library</i> ini juga menawarkan alat untuk pemilihan model dan <i>pre-processing</i> data, seperti normalisasi min-maks, yang berguna dalam persiapan data untuk analisis lebih lanjut [30].
Numpy	NumPy adalah <i>library</i> dasar untuk komputasi <i>scientific</i> di Python. <i>Library</i> ini menyediakan objek <i>array</i> multidimensi, berbagai objek turunan, serta berbagai rutinitas untuk operasi cepat pada <i>array</i> . Fitur-fitur yang disediakan termasuk operasi matematika, logika, manipulasi bentuk, penyortiran, pemilihan, <i>input</i> dan <i>output</i> , transformasi Fourier diskrit, aljabar linier dasar, operasi statistik dasar, simulasi acak, dan masih banyak lagi. NumPy menjadi alat penting dalam pengolahan data numerik dan komputasi ilmiah [31].
Matplotlib	Matplotlib adalah <i>library</i> lengkap untuk keperluan visualisasi statis, animasi, dan interaktif di Python. Matplotlib dapat dengan mudah menghasilkan <i>plot</i> berkualitas tinggi untuk publikasi, serta menciptakan figur interaktif yang memungkinkan <i>zoom</i> , <i>pan</i> , dan

<b>Library Python</b>	<b>Penjelasan</b>
	<p>pembaruan secara <i>real-time</i>. <i>Library</i> ini digunakan untuk menyesuaikan gaya visual dan tata letak grafik sesuai kebutuhan, serta mengekspor hasilnya ke berbagai format <i>file</i>. Selain itu, Matplotlib dapat disematkan dalam JupyterLab dan <i>Graphical User Interface</i> (GUI), dan mendukung integrasi dengan berbagai paket pihak ketiga yang memperluas fungsionalitasnya, menjadikannya alat yang sangat fleksibel untuk berbagai keperluan visualisasi data [32].</p>
OpenCV	<p>OpenCV atau <i>Open Source Computer Vision</i> adalah <i>library open-source</i> yang gratis dan menyediakan berbagai alat untuk aplikasi <i>computer vision</i> dan <i>machine learning</i>. <i>Library</i> ini memiliki berbagai fitur yang berguna dalam pemrosesan gambar dan video secara <i>real-time</i>, memungkinkan mesin untuk memahami konten dari gambar dan video. Dalam penglihatan komputer, OpenCV digunakan untuk berbagai tugas seperti deteksi dan pengenalan wajah, identifikasi objek, klasifikasi aksi manusia dalam video, pelacakan gerakan kamera, pelacakan objek bergerak, serta ekstraksi model 3D objek. OpenCV juga mendukung berbagai algoritma pembelajaran mesin dengan lebih dari 2.500 algoritma yang telah dioptimalkan [33].</p>

### 3.2.2 TensorFlow

TensorFlow adalah *framework open-source* yang dikembangkan oleh Google Brain untuk memfasilitasi pengembangan model *machine learning* dan kecerdasan buatan. *Framework* ini sangat populer dalam pengembangan aplikasi berbasis *neural networks* dan *deep learning*, serta mendukung *deployment* model di berbagai platform. TensorFlow bekerja dengan grafik aliran data yang terdiri dari

*node* dan *edges*, yang memproses data dalam bentuk *array* multidimensi yang disebut *tensor*. Selain itu, TensorFlow mendukung berbagai bahasa pemrograman seperti Python, JavaScript, C++, dan Java, sehingga dapat digunakan di berbagai lingkungan pengembangan. TensorFlow digunakan dalam berbagai aplikasi seperti pengenalan gambar, pemrosesan bahasa alami, otomatisasi data, pelacakan model, pemantauan kinerja, dan pelatihan ulang model. Dengan fitur-fitur tersebut, TensorFlow menjadi alat yang sangat fleksibel dan efisien untuk membangun dan mengelola model *machine learning* yang kompleks [34].

**Tabel 3.2 Daftar *Library* TensorFlow**

<b><i>Library</i></b> <b>TensorFlow</b>	<b>Penjelasan</b>
TensorFlow.js	TensorFlow.js adalah <i>library</i> yang dibangun di atas <i>deeplearn.js</i> untuk memungkinkan pengembangan model <i>deep learning</i> langsung di dalam <i>web browser</i> . <i>Deep learning</i> , yang merupakan cabang dari <i>machine learning</i> dan kecerdasan buatan, dapat diimplementasikan dengan mudah menggunakan TensorFlow.js. Dengan <i>library</i> ini, pengguna dapat membangun berbagai model, seperti <i>Convolutional Neural Network</i> (CNN), <i>Recurrent Neural Network</i> (RNN), dan model <i>deep learning</i> lainnya, langsung di sisi klien tanpa memerlukan server [35].
TensorFlow Lite	TensorFlow Lite adalah versi ringan dari TensorFlow yang dibuat untuk membantu pengembang <i>machine learning</i> dalam mengembangkan dan <i>deployment</i> model. <i>Library</i> ini dioptimalkan untuk menjalankan model pada perangkat dengan sumber daya terbatas, seperti <i>mobile phone</i> dan <i>embedded system</i> . Hal ini memungkinkan penggunaan model <i>machine learning</i> secara efisien pada perangkat dengan kapasitas komputasi yang lebih rendah [36].

### 3.2.2.1 Keras

Keras adalah API tingkat tinggi dari TensorFlow yang dirancang untuk mempermudah pemecahan masalah *machine learning*, terutama dalam *deep learning*. Dengan Keras, pengguna dapat dengan mudah melakukan seluruh proses *machine learning*, mulai dari pemrosesan data, *hyperparameter tuning*, hingga *deployment*. Keras memanfaatkan kekuatan TensorFlow dalam hal skalabilitas dan kemampuan lintas platform, sehingga model dapat dijalankan di TPU *Pod*, kluster GPU besar, serta memudahkan ekspor untuk digunakan di *browser* atau perangkat *mobile*. Selain itu, model Keras juga dapat disajikan melalui API web. Keras dirancang untuk mengurangi kompleksitas dengan menyediakan antarmuka yang sederhana dan konsisten, meminimalkan langkah-langkah yang diperlukan untuk tugas-tugas umum, memberikan pesan kesalahan yang jelas, dan memungkinkan pengguna untuk memulai dengan mudah serta mengembangkan kemampuan seiring waktu. Dengan demikian, Keras membantu pengembang menulis kode yang lebih ringkas dan mudah dibaca [37].

### 3.2.3 Google Colab

Google Colab atau ‘Colaboratory’ adalah platform berbasis web yang memungkinkan pengguna untuk menulis dan menjalankan kode Python secara langsung di *browser* tanpa perlu instalasi atau konfigurasi perangkat lunak tambahan. Platform ini memungkinkan pengguna untuk bekerja dengan *notebooks* yang memadukan kode, visualisasi, dan teks dalam satu tempat, sehingga menjadi alat yang sangat berguna untuk eksperimen, analisis data, dan *machine learning*. Salah satu fitur utama dari Google Colab adalah akses gratis ke GPU dan TPU (*Tensor Processing Unit*), yang memungkinkan pengguna untuk menjalankan model *deep learning* yang memerlukan komputasi intensif tanpa memerlukan perangkat keras mahal [38].

### 3.2.4 Kaggle

Kaggle adalah komunitas *online* untuk *data scientist* dan praktisi *machine learning*. Dengan bergabung dalam komunitas, pengguna dapat mengakses perkembangan terbaru dalam teknik-teknik pembelajaran mesin, berpartisipasi

dalam kompetisi, serta memanfaatkan model dan *dataset* publik yang dapat digunakan untuk latihan atau diimplementasikan dalam proyek pribadi. Kaggle merupakan sumber daya yang sangat berharga bagi *data scientist* dan *machine learning* yang ingin meningkatkan keterampilan mereka, berkolaborasi dengan orang lain, dan menyelesaikan masalah data dunia nyata [39].




### 3.3 Kakas Pemodelan

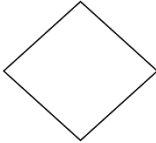

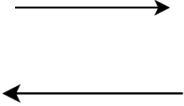
Dalam membangun aplikasi ini, dilakukan pendekatan pemrograman terstruktur. Pemrograman terstruktur merupakan pendekatan atau paradigma dalam pemrograman yang mengutamakan penggunaan struktur logis yang terorganisir dan mudah dipahami dalam penulisan kode [40]. Kakas pemodelan yang digunakan adalah *flowchart* dan *Data Flow Diagram*.

#### 3.3.1 Flowchart

*Flowchart* adalah alat visualisasi yang digunakan untuk menggambarkan proses atau cara kerja dalam bentuk diagram. Dalam dunia pemrograman, *flowchart* digunakan untuk merencanakan langkah-langkah dalam menyelesaikan suatu tugas atau masalah. *Flowchart* digambarkan menggunakan simbol-simbol yang memiliki fungsinya masing-masing. Simbol *flowchart* yang sering digunakan antara lain adalah *terminal*, *flowline*, *input/output*, *decision*, dan *process*. Berikut adalah simbol *flowchart* dan penjelasan masing-masing simbol [41].

Tabel 3.3 Simbol *Flowchart* [41]

Simbol	Nama	Penjelasan
	<i>Terminator</i>	Simbol yang menyatakan awal atau akhir program.
	<i>Input/Output</i>	Simbol proses program mengambil <i>input</i> dan menampilkan <i>output</i> .
	<i>Process</i>	Simbol yang menyatakan suatu proses yang dilakukan komputer.

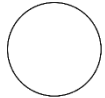
Simbol	Nama	Penjelasan
	<i>Decision</i>	Simbol yang menunjukkan kondisi yang akan menghasilkan dua kemungkinan jawaban, yaitu ya dan tidak.
	<i>On-Page Reference</i>	Simbol untuk keluar – masuk atau penyambungan proses dalam lembar kerja yang sama.
	<i>Flow</i>	Simbol yang digunakan untuk menggabungkan antara simbol yang satu dengan simbol yang lain.



### 3.3.2 Data Flow Diagram

DFD atau *Data Flow Diagram* adalah diagram yang menggambarkan aliran data dalam suatu sistem atau proses. Diagram ini memberikan wawasan tentang *input* dan keluaran setiap entitas serta proses yang terjadi. Dalam *Data Flow Diagram* (DFD), tidak terdapat aliran kontrol, *loop*, atau aturan keputusan. Operasi tertentu yang bergantung pada jenis data dapat dijelaskan dengan menggunakan diagram alir. Sebagai alat grafis, DFD sangat berguna untuk berkomunikasi dengan pengguna, manajer, dan personel lainnya. Selain itu, DFD juga bermanfaat dalam menganalisis sistem yang ada maupun yang diusulkan.

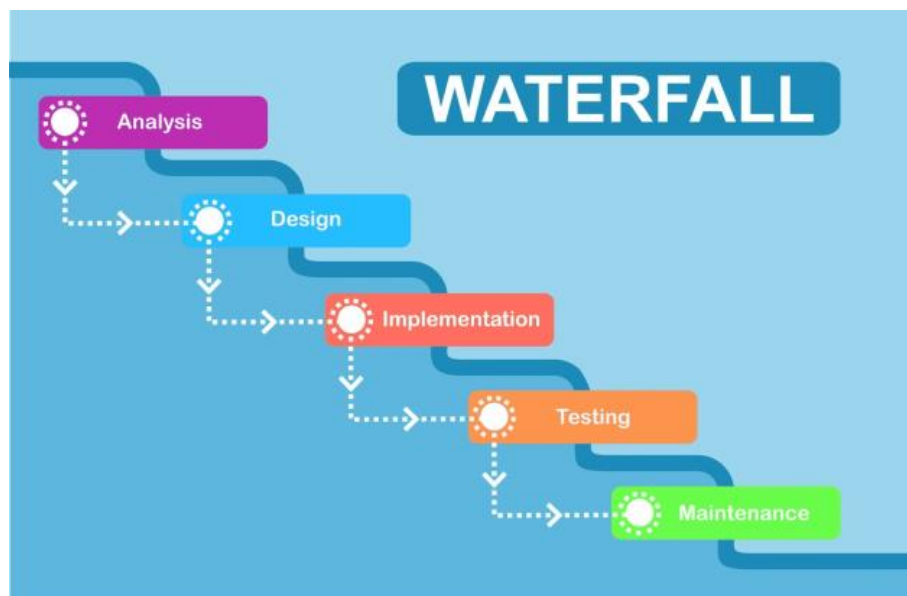
Pada DFD, terdapat beberapa notasi yang digunakan untuk menggambarkan komponen sistem. Entitas digambarkan dengan kotak persegi panjang yang mewakili aktor eksternal. Proses digambarkan dengan lingkaran yang menunjukkan kegiatan dalam sistem. Aliran data diwakili dengan panah yang menghubungkan entitas dan proses [42].

Tabel 3.4 Simbol *Data Flow Diagram* [42]

Simbol	Nama	Penjelasan
	<i>Process</i>	Simbol yang menyatakan proses atau kegiatan dalam sistem.

Simbol	Nama	Penjelasan
	<i>Terminator</i>	Simbol entitas yang berada di luar sistem dan berhubungan dengan sistem.
	<i>Data Flow</i>	Simbol yang digunakan untuk menggabungkan antara simbol yang satu dengan simbol yang lain.

### 3.4 Metodologi Pengembangan Sistem



Gambar 3.2 Metodologi Waterfall [43]

Metodologi *Waterfall* adalah pendekatan yang digunakan dalam pengembangan model machine learning yang memiliki alur kerja yang jelas dan terstruktur. Metode ini dipilih karena sifatnya yang linier, di mana setiap tahap dilakukan secara berurutan, memungkinkan pengelolaan proyek dengan langkah-langkah yang jelas dan mudah dipantau. Setiap fase dalam *Waterfall* memiliki tujuan dan hasil yang harus dicapai sebelum melanjutkan ke tahap berikutnya. Berikut adalah tahapan dari metodologi *Waterfall* [43]:

#### 1. Analisis

Tahap analisis merupakan proses pengumpulan dan analisis data yang berkaitan dengan sistem yang akan dibangun. Pada tahap ini, pengembang harus

mengidentifikasi dan memahami kebutuhan pengguna serta sistem perangkat lunak yang diperlukan. Pengumpulan informasi ini bertujuan untuk memberikan dasar yang jelas bagi tahap-tahap selanjutnya, memastikan bahwa perangkat lunak yang dikembangkan dapat memenuhi tujuan dan kebutuhan pengguna.

## 2. Desain

Setelah analisis selesai, tahap desain dilakukan untuk merancang arsitektur sistem secara menyeluruh. Pengembang akan menyusun spesifikasi kebutuhan pengguna dan mendefinisikan solusi teknis yang akan digunakan. Desain ini mencakup pembuatan diagram alur sistem, desain antarmuka, serta spesifikasi teknis lainnya yang memberikan gambaran tentang bagaimana sistem akan dibangun.

## 3. Implementasi

Pada tahap implementasi, pengembang mulai membangun sistem sesuai dengan desain yang telah dibuat sebelumnya. Program-program dan fitur-fitur yang diperlukan akan ditulis menggunakan bahasa pemrograman yang sesuai. Tahap ini adalah fase di mana kode sumber ditulis dan aplikasi mulai terbentuk berdasarkan spesifikasi yang telah ditentukan.

## 4. Pengujian

Setelah implementasi selesai, tahap pengujian dimulai. Pada tahap ini, pengembang akan mengintegrasikan berbagai komponen program yang telah dibuat dan menguji jalannya sistem secara keseluruhan. Pengujian bertujuan untuk mendeteksi bug atau kesalahan yang mungkin ada dalam aplikasi. Evaluasi dilakukan untuk memastikan bahwa perangkat lunak berjalan dengan baik, sesuai dengan kebutuhan yang sudah dianalisis pada tahap awal, dan memenuhi standar kualitas yang ditetapkan.

## 5. Pemeliharaan

Tahap pemeliharaan adalah tahap terakhir dalam metodologi *Waterfall*, di mana pengembang melakukan perawatan dan perbaikan pada aplikasi setelah implementasi dan pengujian. Pada tahap ini, *bug* atau eror yang terdeteksi setelah perangkat lunak digunakan di dunia nyata akan diperbaiki. Selain itu, pembaruan atau perubahan fitur mungkin dilakukan untuk menyesuaikan dengan perubahan kebutuhan pengguna atau teknologi yang berkembang.

### **3.5 Prosedur Pengumpulan Data**

Data yang dibutuhkan berupa data gambar untuk model *machine learning* CNN. Data akan diambil dari *website* Kaggle untuk mendapatkan data yang tepat dan memudahkan proses pengumpulan data. Data yang dicari adalah data gambar alfabet bahasa isyarat SIBI dan BISINDO. Proses pencarian data dilakukan dengan memperhatikan jumlah data gambar, kualitas gambar, akurasi gambar sesuai labelnya, dan kecocokan format *file* gambar.

## BAB IV

### PEMBAHASAN

#### 4.1 Analisis

Pada tahap analisis, proses pengumpulan dan pengolahan data dijelaskan secara mendalam dengan tujuan untuk memahami komposisi model yang akan dilatih. Tahap ini juga mencakup identifikasi masalah yang ada, peluang yang dapat dimanfaatkan, serta analisis terkait target pengguna dan spesifikasi teknis yang dibutuhkan. Selain itu, analisis ini berfokus pada identifikasi sumber daya yang diperlukan untuk mendukung pengembangan sistem yang lebih efisien.

##### 4.1.1 Pengumpulan dan Pengolahan Data

Proses pengumpulan data digunakan dari sumber data sekunder, di mana data yang digunakan diambil dari Kaggle. *Dataset* yang digunakan berasal dari dua sumber, yaitu *dataset* SIBI dan *dataset* BISINDO. Kedua *dataset* ini dapat diunggah menggunakan *Kagglehub* dalam Google Colab. Data dikumpulkan dan diproses oleh *Machine Learning Cohort*. Data yang valid merupakan data gambar, yang memiliki ekstensi *file* '.jpg', '.jpeg', dan '.png'. Semua data diterima dengan ukuran 224 x 224 pixel sehingga seragam secara ukuran.

Jumlah data valid dari *dataset* SIBI adalah 5280 data yang berasal dari 24 kelas alfabet kecuali alfabet J dan Z yang merupakan alfabet gerakan (*motion*). Hal ini berarti setiap kelas alfabet memiliki masing-masing 220 data. Jumlah data valid dari *dataset* BISINDO adalah 9169 data yang berasal dari 26 kelas alfabet dari A sampai Z. Setelah data di-*load* ke dalam variabel gambar dan label gambar, maka dilakukan *split* data untuk *training* dan *validation*. *Dataset* SIBI dilakukan pembagian 90:10, 90% untuk data *training* dan 10% untuk data *validation*. *Dataset* BISINDO dilakukan pembagian 80:20, 80% untuk data *training* dan 20% untuk data *validation*.

Setelah dilakukan *splitting* data, selanjutnya dilakukan augmentasi data dengan beberapa pengaturan augmentasi ke *dataset* yang digunakan. Yang pertama dilakukan *rescaling* dimana nilai pixel dari gambar digeneralisir sehingga *range* nilai pixel data gambar berada pada *range* yang sama dan mengurangi variasi nilai

yang terlalu jauh. Setelah itu dilakukan augmentasi seperti rotasi, *shift* (pergeseran), *shear* (pemotongan), *zoom*, dan *horizontal flip*. Tujuan teknik augmentasi ini adalah untuk meningkatkan variasi dan kualitas dari data sehingga mewakili situasi dan kondisi yang mungkin terjadi pada model yang digunakan. Data yang sudah diolah dan diaugmentasi akan digunakan dalam proses *training* model menggunakan metode CNN sehingga dapat belajar berdasarkan data dan melakukan klasifikasi.

#### 4.1.2 Analisis Hasil Pengumpulan dan Pengolahan Data

*Dataset* SIBI dan *dataset* BISINDO memiliki data yang sudah siap dipakai dan tidak perlu melakukan proses *cleaning* seperti menghilangkan nilai null dan menghilangkan *outlier* (nilai yang jauh dari rata-rata). *Dataset* SIBI memiliki jumlah data yang sedikit dan kurang bervariasi di mana terdapat 10 variasi latar belakang saja, dengan posisi tangan yang tegak. Tidak ada variasi data dengan objek atau tambahan komponen dalam data. Hal ini berarti mengharuskan gambar yang akan diklasifikasikan harus mengikuti gaya dan pengaturan yang mirip dengan data pada *dataset*.

Solusi untuk masalah ini adalah melakukan augmentasi sehingga mengantisipasi gaya gambar *input* yang akan dimasukan pengguna. Untuk posisi tangan akan sangat berpengaruh jika posisi tangan tidak terlalu tegak, sehingga dilakukan rotasi pada data yang teraugmentasi. Untukantisipasi pengguna yang menggunakan tangan kanan atau kiri, maka digunakan *horizontal flip* pada data augmentasi. Namun, tidak menggunakan *vertical flip* karena akan mengganggu proses klasifikasi, contohnya alfabet K jika terbalik secara vertikal maka akan mirip dengan alfabet P.

Dengan variasi data yang terbatas dan jumlah data yang sedikit, untuk meningkatkan kualitas model *machine learning* SIBI digunakan *transfer learning* dari model CNN VGG16 dengan bobot *imagenet*.

#### 4.1.3 Identifikasi Masalah dan Kesempatan

Dalam bagian ini, akan diuraikan permasalahan yang diangkat dan apa penyebab dan kesempatan yang timbul dari permasalahan tersebut, yang mengakibatkan penulis mengerjakan proyek ini. Masalah yang diuraikan

berdasarkan hasil penelitian dari tim pengembang melalui sesi *brainstorm* bersama. Penelitian dilakukan melalui *website* WHO (*World Health Organization*), Sensus Penduduk, dan beberapa *website* lainnya [1] [2]. Berikut adalah daftarnya.

**Tabel 4.1 Daftar Masalah dan Kesempatan**

<b>Masalah/Kesempatan</b>	<b>Penyebab</b>	<b>Solusi Potensial</b>
Sering ada kesulitan komunikasi antara individu Tuli dengan non-Tuli karena kurangnya inklusivitas untuk individu Tuli dalam masyarakat.	Minimnya fasilitas umum yang mendukung komunikasi dua arah, seperti penerjemah atau alat bantu. Dan kurangnya pendidikan inklusif mengenai bahasa isyarat di sekolah dan tempat kerja.	<ol style="list-style-type: none"> <li>1. Aplikasi penerjemah bahasa isyarat</li> <li>2. Mengadakan pelatihan bahasa isyarat untuk masyarakat umum.</li> <li>3. Peningkatan aksesibilitas teknologi seperti teks otomatis</li> </ol>
Kurangnya pengetahuan dan kesadaran akan komunitas Tuli dan cara mereka berkomunikasi.	Kurang kepedulian atau promosi dari pemerintah mengenai komunitas Tuli di Indonesia. Masih ada <i>gap</i> antara komunitas Tuli dengan non-Tuli.	<ol style="list-style-type: none"> <li>1. Integrasi aplikasi penerjemah bahasa isyarat dengan teknologi <i>machine learning</i>.</li> <li>2. Kampanye nasional untuk meningkatkan kesadaran tentang budaya dan kebutuhan komunitas Tuli.</li> </ol>

#### **4.1.4 Identifikasi Target Pengguna**

Dalam bagian target pengguna, akan diuraikan daftar pengguna yang menjadi target pembangunan proyek sampai dengan relevansi mereka dengan proyek. Penjelasan ini mencakup pengidentifikasian karakteristik utama dari masing-masing kelompok pengguna, alasan mengapa mereka dipilih sebagai target, serta bagaimana kebutuhan dan preferensi mereka relevan terhadap tujuan proyek.

Tabel 4.2 Daftar Pengguna

Pengguna	Peran/Tanggungjawab
Masyarakat	Masyarakat Tuli menggunakan aplikasi untuk mempermudah komunikasi dengan non-Tuli di berbagai situasi, seperti bekerja, belajar, atau aktivitas sehari-hari.
	Memanfaatkan aplikasi untuk memahami dan berkomunikasi dengan individu Tuli, meningkatkan hubungan sosial dan profesional.

#### 4.1.5 Identifikasi Sumber Daya

Dalam bagian identifikasi sumber daya, akan diuraikan daftar sumber daya yang digunakan dalam pembangunan proyek aplikasi penerjemah bahasa isyarat. Sumber daya terbagi menjadi tiga, yaitu pengembang, perangkat keras, dan perangkat lunak.

Tabel 4.3 Daftar Pengembang Aplikasi

Nama	<i>Learning Path</i>	Tanggungjawab
John Jong	<i>Machine Learning</i>	<i>Team lead</i> dan membangun model ML
Yulianto Aryaseta	<i>Machine Learning</i>	Membangun model ML
Rama Diaz	<i>Cloud Computing</i>	Menangani pemrograman <i>backend</i> bagian <i>cloud</i>
Kevin Sipahutar	<i>Cloud Computing</i>	Menangani Google Cloud Platform

Tabel 4.4 Daftar Perangkat Keras

Perangkat Keras ( <i>Hardware</i> )	Spesifikasi
Laptop	HP Pavillion x360 14-dh1052TX Intel(R) Core(TM) i5-10210U CPU @ 1.60GHz 16GB 512GB

Tabel 4.5 Daftar Perangkat Lunak

Perangkat Lunak (Software)	Spesifikasi
<i>Operating System</i>	Windows 11
Google Colab (200 Compute Unit)	Virtual Hardware A100 GPU ( <i>Memory</i> 80GB HBM2e, <i>Memory Bandwidth</i> 2TB/s)
Microsoft Word	Versi 2411
Draw.io	Versi 26.0.3
Visual Studio Code	Versi 1.96.2

#### 4.1.6 Daftar Spesifikasi Persyaratan

Daftar spesifikasi persyaratan akan menguraikan daftar hal yang butuh untuk dipenuhi dalam aplikasi yang akan dibangun. Spesifikasi persyaratan dibagi menjadi spesifikasi persyaratan pengguna dan spesifikasi persyaratan sistem. Berikut adalah daftar spesifikasi persyaratan.

##### A. Spesifikasi Persyaratan Pengguna

1. Pengguna dapat menerjemahkan alfabet bahasa isyarat melalui aplikasi.
2. Pengguna dapat memilih menerjemahkan alfabet bahasa isyarat SIBI dan BISINDO.
3. Pengguna dapat melakukan *input* gambar atau video untuk diterjemahkan.
4. Pengguna dapat melakukan proses autentikasi sebagai persyaratan untuk melakukan penerjemahan.

##### B. Spesifikasi Persyaratan Sistem

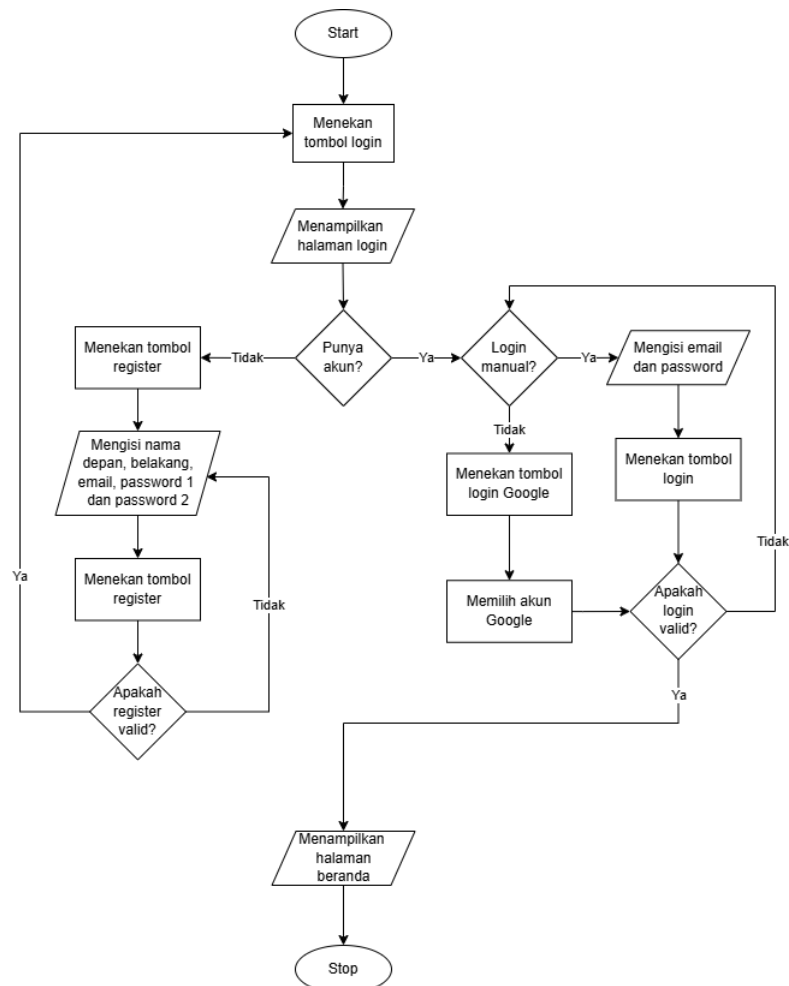
1. Aplikasi dapat menerjemahkan alfabet bahasa isyarat menggunakan model *machine learning*.
2. Aplikasi dapat menerjemahkan alfabet bahasa isyarat SIBI dan BISINDO, tergantung pilihan pengguna.
3. Aplikasi dapat menerima *input* gambar atau video dari pengguna untuk diterjemahkan.
4. Aplikasi dapat menerima informasi pengguna dan melakukan validasi proses autentikasi yang dilakukan pengguna.

## 4.2 Desain

Pada tahap desain akan dilakukan pemodelan sistem yang akan dibangun menggunakan *flowchart* dan *Data Flow Diagram* (DFD). Selain pemodelan sistem, akan dilakukan desain *storyboard* untuk penggambaran antarmuka yang akan diimplementasikan.

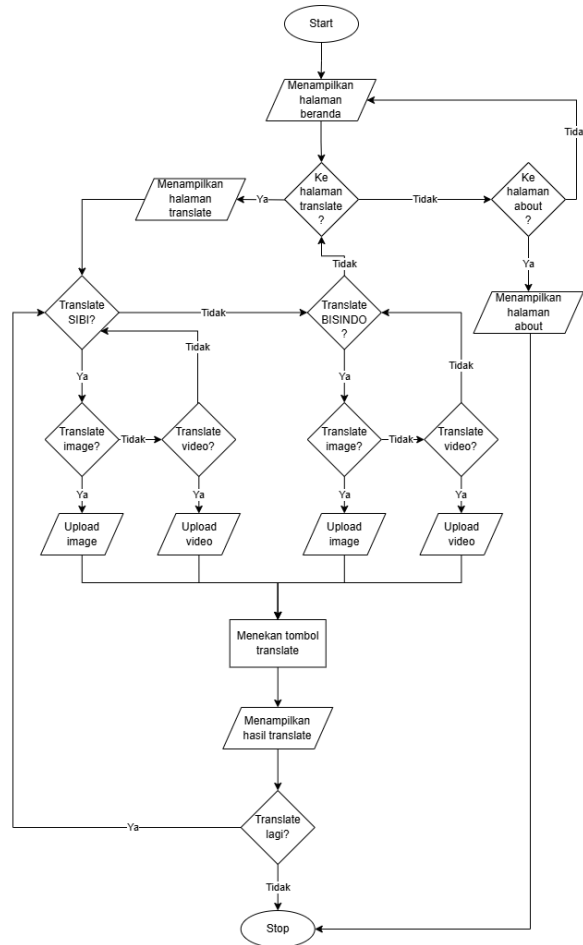
### 4.2.1 Flowchart

Pada bagian ini akan diuraikan pemodelan sistem menggunakan *flowchart* untuk melakukan penggambaran alur kerja sistem yang dibangun. Sistem yang digambarkan adalah sistem autentikasi dan sistem penerjemah.



**Gambar 4.1 Flowchart Autentikasi**

Gambar di atas merupakan penggambaran alur sistem untuk proses autentikasi. Proses autentikasi melibatkan proses *login* (masuk) dan *register* (daftar).



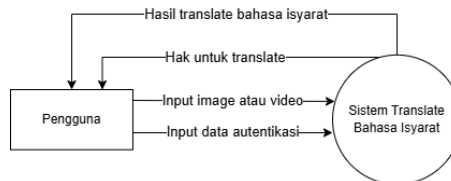
**Gambar 4.2 Flowchart Terjemah Bahasa Isyarat**

Gambar di atas merupakan penggambaran alur sistem pada aplikasi. Aplikasi akan mulai dengan halaman beranda dan menyediakan pilihan untuk terjemah dan halaman *about*. Untuk terjemah memiliki dua fitur, terjemahan SIBI dan BISINDO. Untuk masing-masing bahasa isyarat memiliki dua jenis *input* data yaitu gambar dan video.

**4.2.2 Data Flow Diagram**

Pada bagian ini akan dilampirkan penggambaran alur data yang digunakan dalam aplikasi penerjemah bahasa isyarat menggunakan *Data Flow Diagram*.

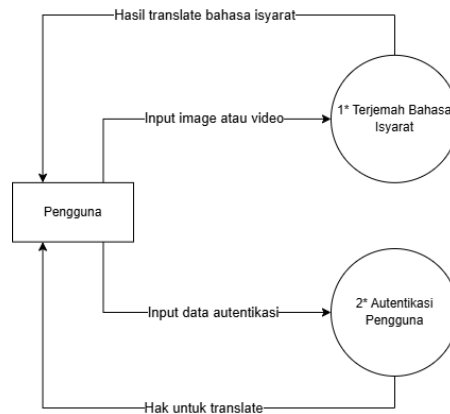
**4.2.2.1 DFD Level 0**



**Gambar 4.3 DFD Level 0**

DFD level 0 memberikan penggambaran alur data tingkat tinggi berdasarkan hubungan interaksi entitas eksternal dengan sistem. Pengguna merupakan entitas eksternal yang memberikan *input* gambar atau video dan mendapatkan *output* berupa hasil terjemahan.

#### 4.2.2.2 DFD Level 1



**Gambar 4.4 DFD Level 1**

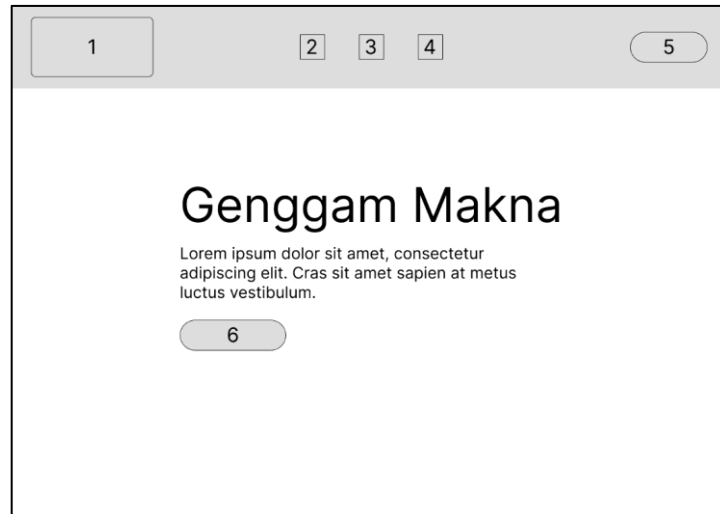
DFD level 1 memberikan penggambaran yang lebih terperinci mengenai aplikasi penerjemah bahasa isyarat. Terdapat dua proses utama, yaitu penerjemah SIBI dan penerjemah BISINDO. Pengguna yang merupakan entitas eksternal, melakukan *input* data gambar atau video dan mendapat *output* hasil terjemahan dari masing-masing penerjemah.

#### 4.2.3 Storyboard

Perancangan *storyboard* untuk memberikan gambaran alur aplikasi yang akan dibangun dan tata letak fitur yang ada. Berikut adalah rancangan *storyboard* untuk aplikasi yang dibuat.

##### 4.2.3.1 Storyboard Beranda

Dalam aplikasi ini, akan dirancang dan dibangun sebuah halaman beranda yang berfungsi sebagai halaman awal yang ditampilkan saat pengguna membuka aplikasi. Halaman ini akan dirancang secara intuitif dan menarik, dengan tata letak yang memudahkan pengguna untuk mengenali fungsi utama aplikasi dan langsung memahami navigasi yang tersedia.



**Gambar 4.5 Storyboard Beranda**

Berikut adalah penjelasan dari penomoran fitur pada halaman beranda.

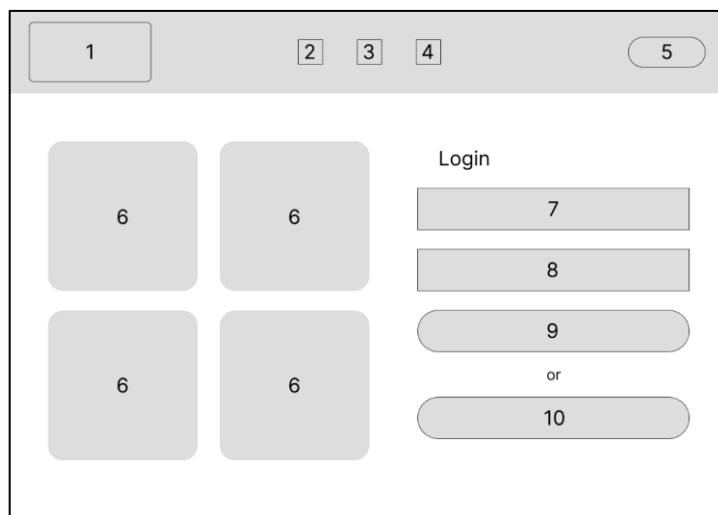
1. Logo: Menampilkan logo aplikasi yang berisi simbol dan nama aplikasi
2. Simbol beranda (*homepage*): Tombol untuk ke halaman beranda
3. Simbol bahasa isyarat: Tombol untuk ke halaman penerjemah
4. Simbol *information* (*about us*): Tombol untuk ke halaman *about us*
5. Tombol *login*: Tombol untuk ke halaman *login*
6. Tombol *Try Now*: Tombol lain untuk ke halaman penerjemah

#### 4.2.3.2 Storyboard Register dan Login

Dalam aplikasi ini akan dibangun halaman autentikasi yang memiliki halaman register (daftar) dan halaman *login*.



**Gambar 4.6 Storyboard Register**



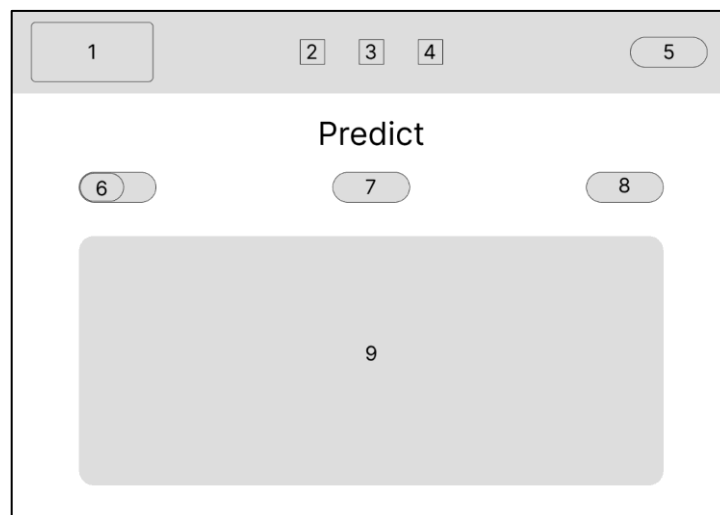
**Gambar 4.7 Storyboard Login**

Berikut adalah penjelasan dari penomoran fitur halaman register dan halaman *login*.

1. Logo: Menampilkan logo aplikasi yang berisi simbol dan nama aplikasi
2. Simbol beranda (*homepage*): Tombol untuk ke halaman beranda
3. Simbol bahasa isyarat: Tombol untuk ke halaman penerjemah
4. Simbol *information* (*about us*): Tombol untuk ke halaman *about us*
5. Tombol *login*: Tombol untuk ke halaman *login*
6. Gambar sebagai hiasan halaman (bahasa isyarat)
7. *Field Email Address*: *Field* untuk meng-*input* email pengguna
8. *Field Password*: *Field* untuk meng-*input* *password* pengguna
9. Tombol *Submit*: Tombol untuk mengirim *input* email dan *password* pengguna
10. Tombol Google: Tombol untuk melakukan *login* menggunakan Google
11. *Field First name*: *Field* untuk meng-*input* nama pertama pengguna
12. *Field Last name*: *Field* untuk meng-*input* nama terakhir pengguna

#### **4.2.3.3 Storyboard Penerjemahan**

Dalam aplikasi ini, terdapat sebuah halaman khusus penerjemah bahasa isyarat yang menjadi salah satu fitur utama yang dirancang dan dibangun dengan cermat. Halaman ini berfungsi untuk menerjemahkan gambar bahasa isyarat ke dalam teks, sehingga dapat memfasilitasi komunikasi antara pengguna bahasa isyarat dan orang-orang yang tidak menguasai bahasa isyarat.



**Gambar 4.8 Storyboard Penerjemahan**

Berikut adalah penjelasan penomoran fitur yang ada pada halaman penerjemah.

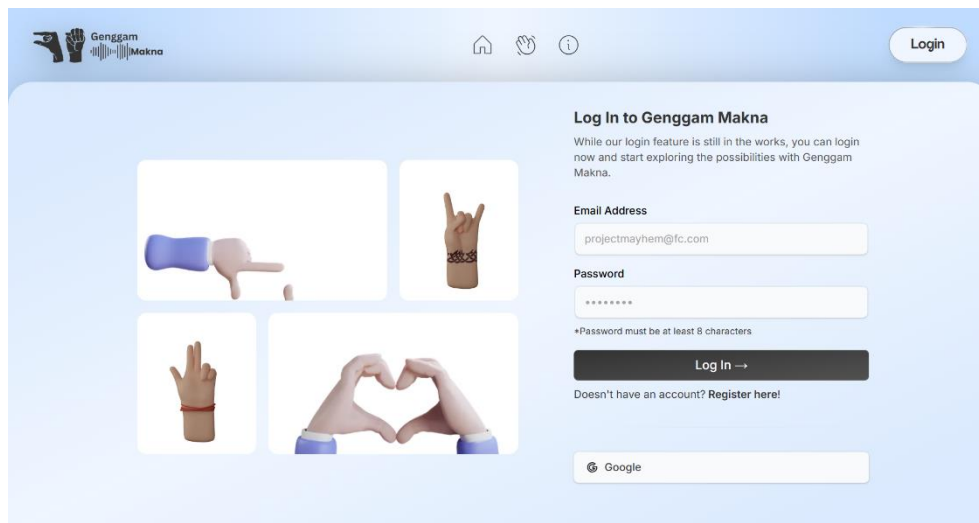
1. Logo: Menampilkan logo aplikasi yang berisi simbol dan nama aplikasi
2. Simbol beranda (*homepage*): Tombol untuk ke halaman beranda
3. Simbol bahasa isyarat: Tombol untuk ke halaman penerjemah
4. Simbol *information* (*about us*): Tombol untuk ke halaman *about us*
5. Tombol *login*: Tombol untuk ke halaman *login*
6. Tombol *switch* jenis *input*: Tombol untuk mengganti jenis *input* antara gambar dan video
7. Tombol *switch* bahasa: Tombol untuk mengganti jenis bahasa isyarat yang akan diterjemahkan
8. Tombol terjemahkan: Tombol yang ditekan untuk melakukan proses terjemahan *input*
9. Tombol *upload input*: Tombol untuk mengakses penyimpanan lokal *device* untuk *upload input* gambar atau video

### 4.3 Implementasi

Pada tahap ini, akan dilakukan implementasi berdasarkan tahap yang dilakukan sebelumnya, yaitu tahap desain. Implementasi yang dicantumkan adalah implementasi antarmuka yang berisi antarmuka aplikasi dan implementasi pemrograman yang berisi kode program.

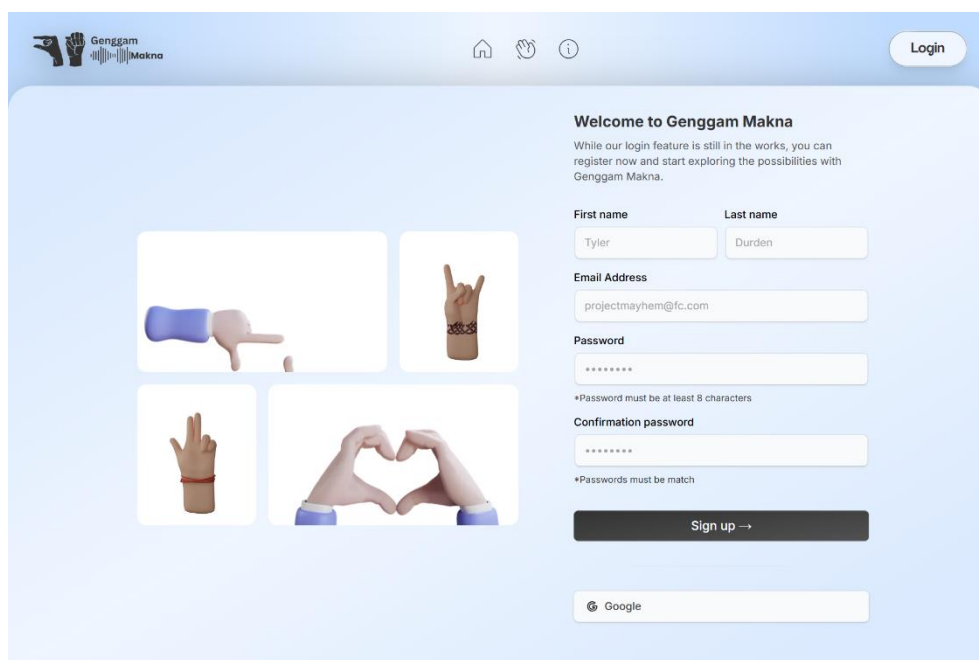
### 4.3.1 Implementasi Tampilan Antarmuka

Tampilan antarmuka diimplementasikan menggunakan bahasa pemrograman dan *storyboard* yang dirancang sesuai pada tahap desain.



**Gambar 4.9 Halaman Login**

Halaman *login* dapat diakses melalui berbagai tombol *login* pada navigasi dan halaman terjemah. Halaman *login* berisi *field* email dan *field* password, serta tombol *log in*. Fitur tambahan adalah tombol *login* menggunakan Google pada bagian bawah *login*.



**Gambar 4.10 Halaman Register**

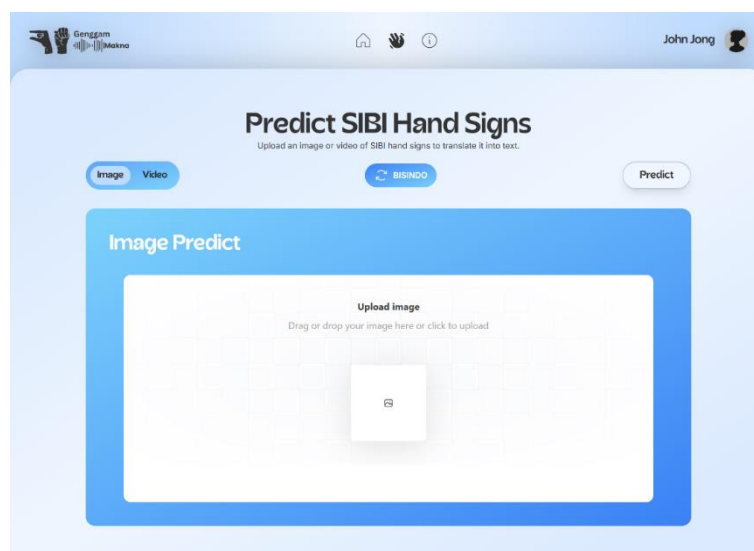
Halaman register merupakan halaman yang muncul ketika pengguna menekan tombol register saat belum mempunyai akun. *Field* di register lebih

banyak dari halaman *login*, yaitu *field First name*, *field Last name*, *field email*, *field password*, dan *confirm password*. Setelah itu ada tombol *sign up* dan tombol *sign up* menggunakan Google.



**Gambar 4.11 Halaman Beranda**

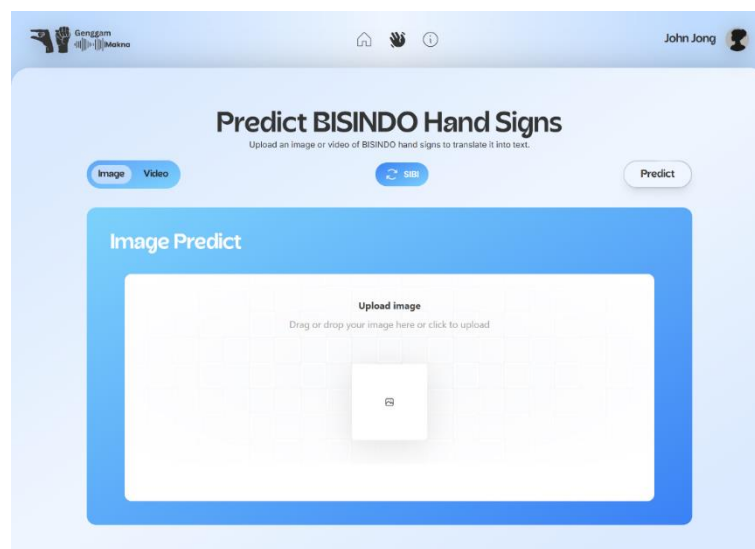
Halaman beranda adalah halaman awal yang ditampilkan muncul pada saat mengakses aplikasi GenggamMakna. Halaman ini berisi pengenalan singkat dan tombol untuk ke halaman terjemah. Navigasi aplikasi berisi logo, tombol beranda, tombol terjemah, tombol *about us*, dan tombol *login*.



**Gambar 4.12 Halaman Terjemah SIBI**

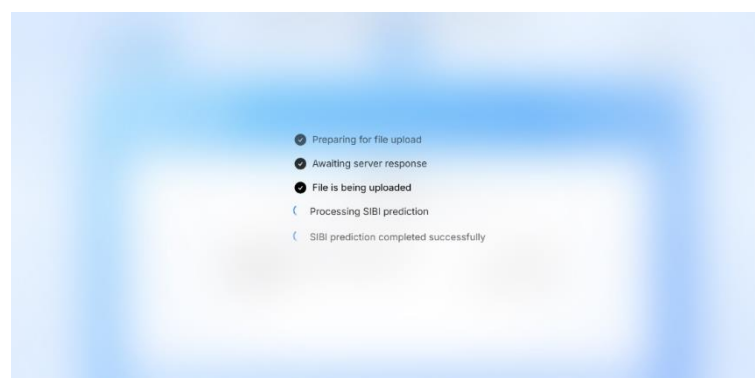
Halaman terjemah memiliki beberapa komponen, yang pertama adalah penerjemah bahasa isyarat SIBI. Halaman ini dapat berubah menggunakan tombol BISINDO di bagian tengah halaman. Pada bagian kiri atas juga ada tombol *Image*

dan *Video* untuk memilih format *file* yang menjadi *input*. Kemudian ada tombol untuk melakukan terjemahan.



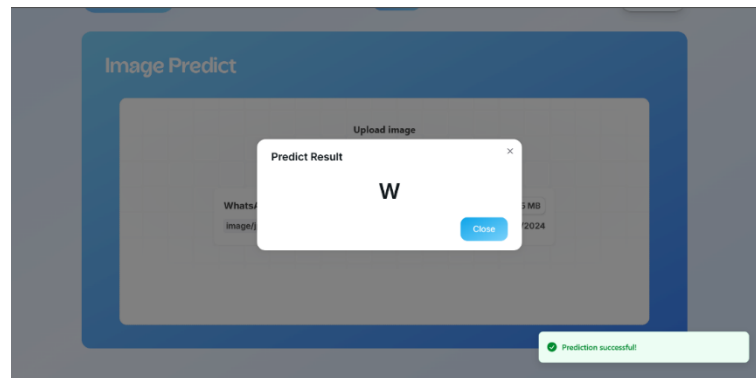
**Gambar 4.13 Halaman Terjemah BISINDO**

Halaman terjemah yang kedua adalah penerjemah bahasa isyarat BISINDO. Halaman ini dapat berubah menggunakan tombol BISINDO di bagian tengah halaman. Pada bagian kiri atas juga ada tombol *Image* dan *Video* untuk memilih format *file* yang menjadi *input*. Kemudian ada tombol untuk melakukan terjemahan.



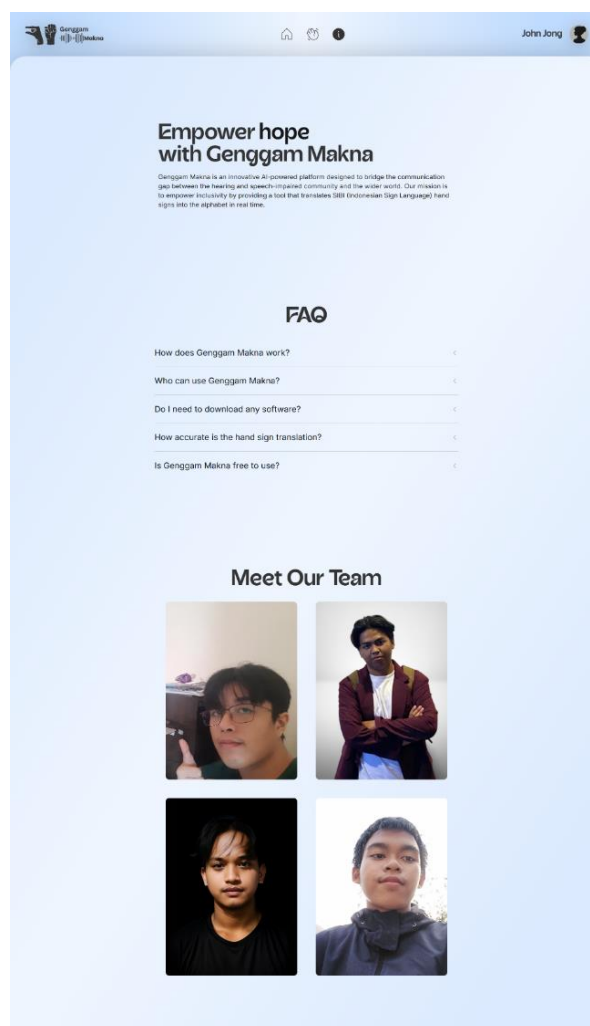
**Gambar 4.14 Halaman *Loading* Terjemahan**

Halaman di atas merupakan halaman *loading* saat melakukan terjemahan. Akan ditunjukkan proses yang dilakukan dari *file upload* sampai dengan proses penerjemahan.



**Gambar 4.15 Tampilan Hasil Terjemahan**

Tampilan di atas merupakan tampilan hasil terjemahan yang muncul saat melakukan penerjemahan. Yang muncul adalah hasil terjemahan dan status terjemahan, apakah berhasil atau gagal.



**Gambar 4.16 Halaman About Us**

Halaman terakhir adalah halaman *about us* di mana halaman ini merupakan halaman berisi FAQ (*Frequently Asked Question*) dan halaman pengenalan tim pengembang.

### 4.3.2 Implementasi Pemrograman

**Tabel 4.6 Kode Model SIBI**

Fitur/Fungsi	Kode Model SIBI
<p><i>Load</i> dan proses data</p>	<pre>import kagglehub path = kagglehub.dataset_download("alvinbintang/ sibi-dataset") print("Path to dataset files:", path)  train_dir = os.path.join(path, "SIBI") eval_dir = os.path.join(path, "SIBI")  valid_extensions = {".jpg", ".jpeg", ".png"} total_images = 0 for root, _, files in os.walk(train_dir):     for file in files:         if os.path.splitext(file)[1].lower() in valid_extensions:             total_images += 1 print("Total valid images:", total_images)  def load_images(directory):     images = []     labels = []     for idx, label in enumerate(class_label):         for file in os.listdir(directory + "/" + label):             filepath = directory + "/" + label + "/" + file             image = cv2.resize(cv2.imread(filepath), (224, 224))              images.append(image)             labels.append(idx)     images = np.array(images)     labels = np.array(labels)     return(images, labels)  class_label = sorted(os.listdir(train_dir))</pre>

Fitur/Fungsi	Kode Model SIBI
	<pre> images, labels = load_images(directory = train_dir) if class_label == sorted(os.listdir(eval_dir)):     X_eval, y_eval = load_images(directory = eval_dir)  X_train, X_test, y_train, y_test = train_test_split(images, labels, test_size = 0.1, stratify = labels)  # one-hot encode categorical labels y_train = keras.utils.to_categorical(y_train) y_test = keras.utils.to_categorical(y_test) y_eval = keras.utils.to_categorical(y_eval)  # convert datatype and normalization X_train = X_train.astype('float32')/255.0 X_test = X_test.astype('float32')/255.0 X_eval = X_eval.astype('float32')/255.0 </pre>
<p><b>Augmentasi data</b></p>	<pre> # augmentation train_datagen = ImageDataGenerator(     rotation_range=30, # Random rotations between 0 and 30 degrees     width_shift_range=0.2, # Random horizontal shifts     height_shift_range=0.2, # Random vertical shifts     shear_range=0.2, # Random shearing zoom_range=0.2, # Random zooms     horizontal_flip=True, # Random horizontal flips     fill_mode='nearest', # Fill empty pixels after transformations )  train_data_aug = train_datagen.flow(X_train, y_train) </pre>
<p><b>Model building, transfer learning, dan model training</b></p>	<pre> # transfer learning input_layer = Input(shape=(224, 224, 3)) pt_model = VGG16(input_tensor=input_layer, include_top=False, weights='imagenet')  # freeze all the layers of the pre- trained model for layer in pt_model.layers:     layer.trainable = False </pre>

Fitur/Fungsi	Kode Model SIBI
	<pre> x = pt_model.output  x = Flatten() (x) x = Dense(units=128, activation='relu') (x) x = BatchNormalization() (x) x = Dense(units=64, activation='relu') (x) x = BatchNormalization() (x) x = Dense(units=32, activation='relu') (x) x = BatchNormalization() (x) x = Dropout(0.1) (x) x = Dense(units=32, activation='relu') (x) x = BatchNormalization() (x) x = Dense(units=64, activation='relu') (x) x = BatchNormalization() (x) x = Dense(units=128, activation='relu') (x) x = BatchNormalization() (x) x = Dense(units=24, activation='softmax') (x)  model = Model(inputs = input_layer, outputs = x)  model.summary()  # compile the model model.compile(     optimizer=RMSprop(learning_rate=0.001),     loss='categorical_crossentropy',     metrics=['accuracy'] ) # callbacks class CustomCallback(tf.keras.callbacks.Callback):     def on_epoch_end(self, epoch, logs=None):         if logs.get('accuracy') &gt;= 0.90:             print("\nReached 90% accuracy, stopping training...")             self.model.stop_training = True callback = CustomCallback() # fit (train) history = model.fit(     train_data_aug,     epochs=200,     validation_data=(X_test,y_test),     callbacks=[callback], batch_size=64 ) </pre>
<i>Model testing</i>	<pre> model = load_model('models\lfinal.h5') </pre>

Fitur/Fungsi	Kode Model SIBI
	<pre> def classify_image(img_path, top_n=3):     img = image.load_img(img_path, target_size=(224, 224))      img_array = image.img_to_array(img)     img_array = np.expand_dims(img_array, axis=0)     img_array = img_array.astype('float32') / 255.0      prediction = model.predict(img_array)      class_labels = sorted(os.listdir(train_dir))      top_indices = np.argsort(prediction[0])[:, :-1][:top_n]      top_classes = [(class_labels[idx], prediction[0][idx]) for idx in top_indices]      print("Top predictions:")     for label, prob in top_classes:         print(f"{label}: {prob * 100:.2f}%")      img = cv2.imread(img_path)     img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)      plt.imshow(img)     plt.title("\n".join([f"{label}: {prob * 100:.2f}%" for label, prob in top_classes]))     plt.axis('off')     plt.show()  img_path = '/content/M.jpeg' classify_image(img_path, top_n=5) </pre>

Tabel 4.7 Kode Model BISINDO

Fitur/Fungsi	Kode Model BISINDO
Load, proses, dan augmentasi data	<pre> import kagglehub # Download latest version path = kagglehub.dataset_download("agungmrf/indo nesian-sign-language-bisindo") print("Path to dataset files:", path) </pre>

Fitur/Fungsi	Kode Model BISINDO
	<pre> train_datagen = ImageDataGenerator(     rescale=1./255,     horizontal_flip=True,     shear_range = 0.2,     zoom_range=0.15,     width_shift_range=0.2,     height_shift_range=0.2,     fill_mode = 'nearest',     validation_split=0.2)  validation_datagen = ImageDataGenerator(rescale = 1.0/255,  validation_split=0.2) train_generator = train_datagen.flow_from_directory(     path_train,     target_size=(100,100),     batch_size=32,     class_mode='categorical',     shuffle=True,     subset='training')  validation_generator = validation_datagen.flow_from_directory(     path_train,     target_size=(100,100),     batch_size=32,     class_mode='categorical',     shuffle=False,     subset='validation') </pre>
<p><i>Model building dan model training</i></p>	<pre> model = tf.keras.models.Sequential([     tf.keras.layers.Conv2D(32, (3, 3), activation='relu', padding='same', input_shape=(100, 100, 3)),     tf.keras.layers.MaxPooling2D(2, 2, padding='same'),     tf.keras.layers.Dropout(0.2),      tf.keras.layers.Conv2D(64, (3, 3), activation='relu', padding='same'),     tf.keras.layers.MaxPooling2D(2, 2, padding='same'),     tf.keras.layers.Dropout(0.3),      tf.keras.layers.Conv2D(128, (3, 3), activation='relu', padding='same'),     tf.keras.layers.MaxPooling2D(2, 2, padding='same'),     tf.keras.layers.Dropout(0.4),      tf.keras.layers.Flatten(), </pre>

Fitur/Fungsi	Kode Model BISINDO
	<pre> tf.keras.layers.Dense(512, activation='relu'), tf.keras.layers.Dropout(0.2), tf.keras.layers.Dense(128, activation='relu'), tf.keras.layers.Dropout(0.3),  tf.keras.layers.Dense(26, activation='softmax') ]) optimizer = tf.keras.optimizers.Adam(learning_rate=1e -4) model.compile(loss='categorical_crossentropy', optimizer=optimizer, metrics=['accuracy']) es=EarlyStopping(monitor='accuracy', verbose=1, patience=8, mode='auto') mc=ModelCheckpoint(filepath='/content/keras', monitor='accuracy', verbose=1, save_best_only=True) lr=ReduceLRonPlateau(monitor='accuracy', verbose=1, patience=5, min_lr=0.001)  history = model.fit( train_generator, epochs=200, validation_data=validation_generator, callbacks=[mc, es, lr] ) </pre>
<i>Model testing</i>	<pre> model = load_model('/content/my_model.h5')  # Function to preprocess image and predict class def classify_image(img_path): img = image.load_img(img_path, target_size=(100, 100))  img_array = image.img_to_array(img) img_array = np.expand_dims(img_array, axis=0) img_array = img_array.astype('float32') / 255.0  # Predict the class prediction = model.predict(img_array)  predicted_class_idx = np.argmax(prediction[0]) </pre>

Fitur/Fungsi	Kode Model BISINDO
	<pre> class_labels = sorted(os.listdir(train_dir)) predicted_class_label = class_labels[predicted_class_idx]  # Print and display the result print(f"Predicted class: {predicted_class_label} (Index: {predicted_class_idx}) ")  # Show the image img = cv2.imread(img_path) img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB) plt.imshow(img) plt.title(f"Predicted: {predicted_class_label}") plt.axis('off') plt.show()  img_path = '/content/WhatsApp Image 2024- 12-10 at 21.11.52_62876d9c.jpg' classify_image(img_path) </pre>

#### 4.4 Pengujian

Pada tahap terakhir bab pembahasan akan dibahas pengujian yang dilakukan pada aplikasi dan model *machine learning*. Pengujian bertujuan untuk menilai dan menganalisis kinerja aplikasi yang memenuhi spesifikasi persyaratan.

##### 4.4.1 Tujuan Pengujian

Berikut merupakan tujuan dilakukan pengujian aplikasi penerjemah bahasa isyarat.

1. Memastikan aplikasi dapat dijalankan pada berbagai *browser*.
2. Memastikan bahwa aplikasi dapat menampilkan tampilan yang diimplementasikan pada tahap sebelumnya.
4. Memastikan bahwa fungsionalitas pada aplikasi berjalan dengan baik dan dapat digunakan pengguna.
5. Memastikan bahwa model penerjemah SIBI dan BISINDO dapat mengenali gambar dan video yang di-*input*.

6. Memastikan bahwa akurasi penerjemah model tinggi dan dapat menerjemahkan bahasa isyarat dengan tepat.

#### **4.4.2 Kriteria Pengujian**

Kriteria pengujian menentukan keberhasilan atau kegagalan kondisi pengujian yang dilakukan terhadap aplikasi. Berikut adalah kriteria pengujian untuk aplikasi yang dibuat.

1. Pengguna berhasil mengakses aplikasi melalui *browser* Microsoft Edge, Mozilla Firefox, dan Google Chrome.
2. Pengguna berhasil melakukan navigasi ke setiap halaman aplikasi dengan efisien dan tanpa eror.
3. Pengguna berhasil melakukan proses autentikasi register dan *login* ke aplikasi.
4. Pengguna berhasil mengunggah gambar atau video dari *device* mereka ke aplikasi, dan menekan tombol terjemah untuk mendapatkan hasil terjemahan.
5. Pengguna berhasil mendapatkan hasil terjemahan yang tepat dan persentase ketepatan yang tinggi dari total hasil terjemahan yang dilakukan.

#### **4.4.3 Kasus Pengujian**

Kasus pengujian merupakan skenario cara melakukan pengujian terhadap aplikasi. Berikut adalah kasus pengujian aplikasi yang dibuat.

1. Pengujian pada berbagai *browser*.
2. Pengujian navigasi dan akses halaman aplikasi.
3. Pengujian autentikasi register dan *login*.
4. Pengujian menerjemahkan bahasa isyarat SIBI dan BISINDO.

#### **4.4.4 Pelaksanaan Pengujian**

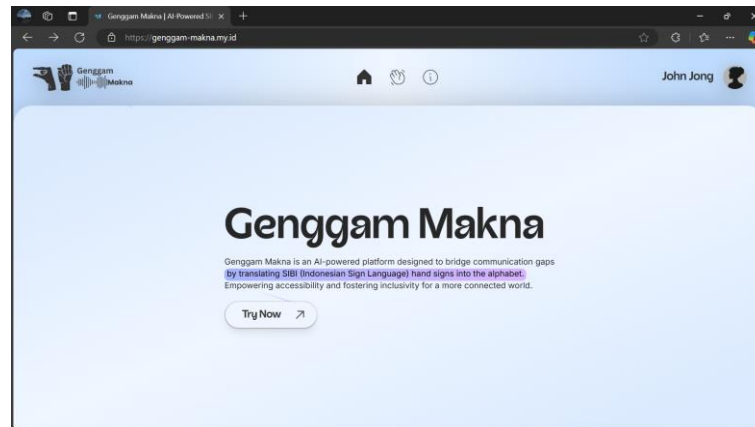
Pada bagian ini akan dilampirkan hasil dari pengujian yang dilakukan pada aplikasi. Pelaksanaan pengujian dilakukan berdasarkan kasus pengujian yang diuraikan pada bagian sebelumnya. Berikut adalah pelaksanaan pengujian.

##### **4.4.4.1 Pelaksanaan Pengujian pada *Browser***

Pengujian pada *browser* dilakukan menggunakan *browser* Microsoft Edge, Mozilla Firefox, dan Google Chrome dengan versi tertentu saat pengujian.

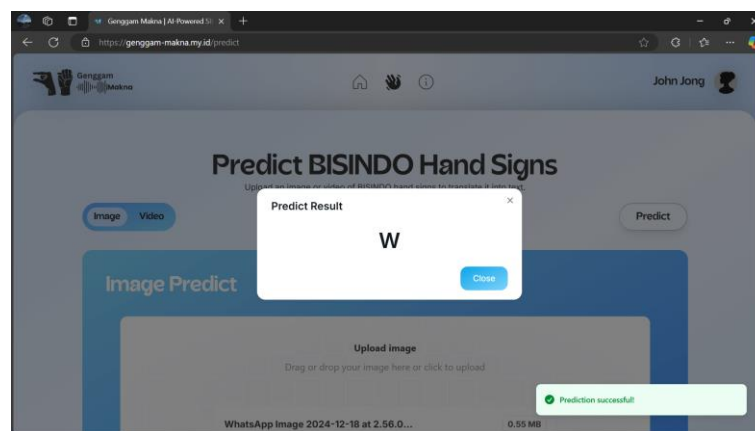
## 1. Microsoft Edge

Pelaksanaan pengujian di *browser* yang pertama adalah di Microsoft Edge dengan versi 131.0.2903.112 (*Official build*) (64-bit). Pengujian di Microsoft Edge dilakukan sebagai salah satu *browser* populer, dilakukan untuk memastikan kompatibilitas agar aplikasi dapat dijalankan lintas platform.



**Gambar 4.17 Halaman Beranda pada Microsoft Edge**

Gambar di atas merupakan halaman beranda dari aplikasi GenggamMakna yang sudah dilakukan proses *login*.

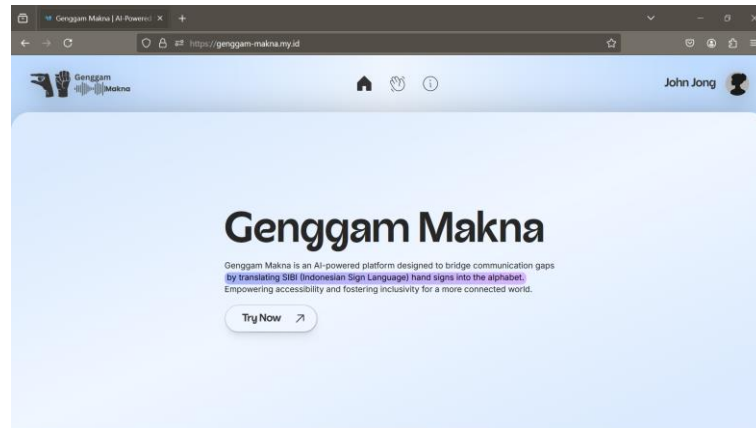


**Gambar 4.18 Hasil Terjemahan pada Microsoft Edge**

Gambar kedua merupakan gambar hasil terjemahan yang muncul setelah pengguna meng-*input* gambar dan menekan tombol terjemah. Aplikasi dapat dijalankan pada *browser* Microsoft Edge dengan semua fungsionalitasnya berjalan.

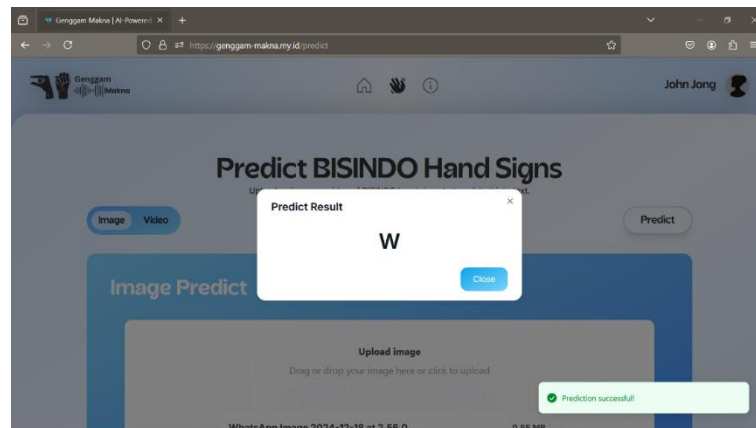
## 2. Mozilla Firefox

Pelaksanaan pengujian *browser* yang kedua adalah pada Mozilla Firefox dengan versi 128.0 (64-bit). Mozilla Firefox merupakan salah satu *browser* yang populer dan pengujian pada *browser* ini untuk memastikan aplikasi dapat dijalankan secara lintas platform.



**Gambar 4.19 Halaman Beranda pada Mozilla Firefox**

Gambar di atas merupakan hasil pencarian aplikasi dan halaman beranda aplikasi, di mana sudah dilakukan proses *login*.

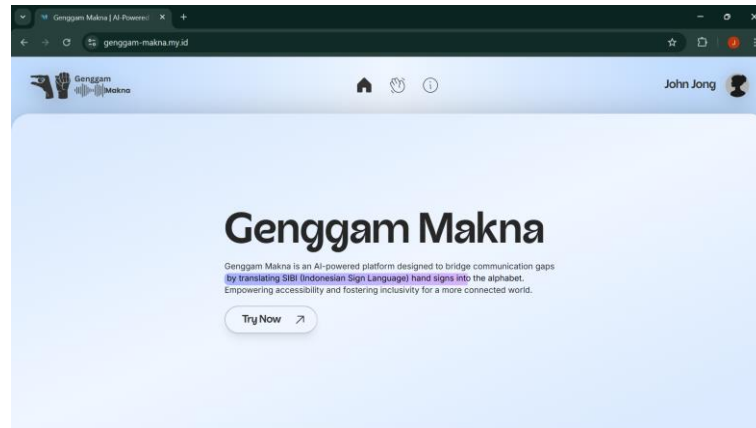


**Gambar 4.20 Tampilan Hasil Terjemahan pada Mozilla Firefox**

Gambar di atas merupakan hasil terjemahan setelah pengguna melakukan *input* gambar dengan mengunggah gambar dari penyimpanan lokal. Aplikasi dapat dijalankan di *browser* Mozilla Firefox dan semua fungsionalitasnya berjalan dengan baik.

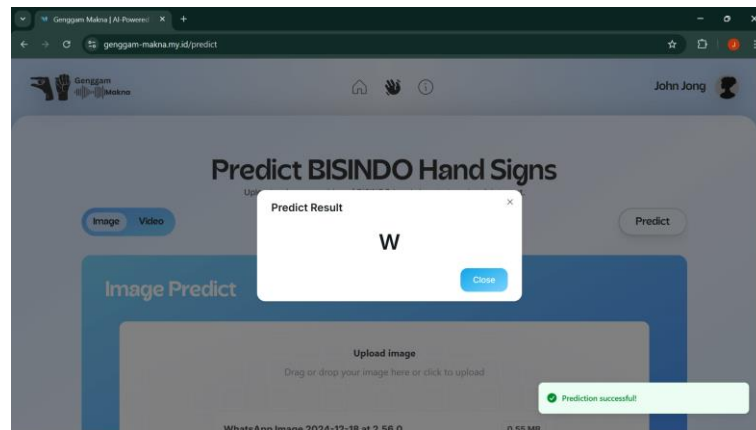
### 3. Google Chrome

Pelaksanaan pengujian pada browser yang ketiga adalah pada Google Chrome versi 131.0.6778.205 (*Official Build*) (64-bit). Google Chrome merupakan salah satu *browser* pengujian untuk aplikasi yang dibangun. Penggunaan Google Chrome sebagai salah satu alat pengujian juga memberikan jaminan bahwa aplikasi tersebut mendukung teknologi web terkini, seperti HTML5, CSS3, dan JavaScript.



**Gambar 4.21 Halaman Beranda pada Google Chrome**

Gambar di atas merupakan halaman beranda yang sudah ada nama pengguna setelah melakukan proses *login*.

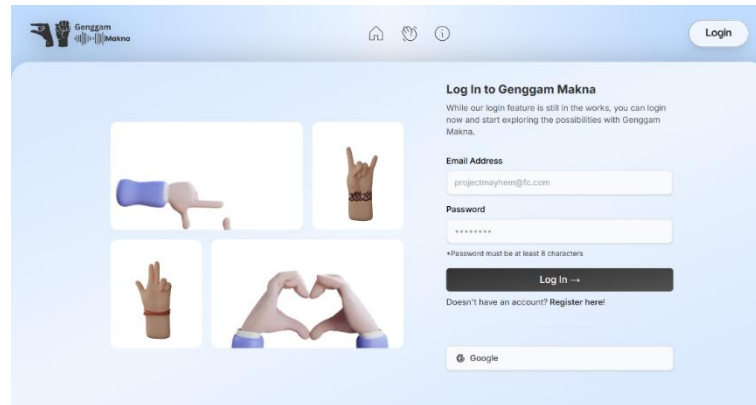


**Gambar 4.22 Tampilan Hasil Terjemahan pada Google Chrome**

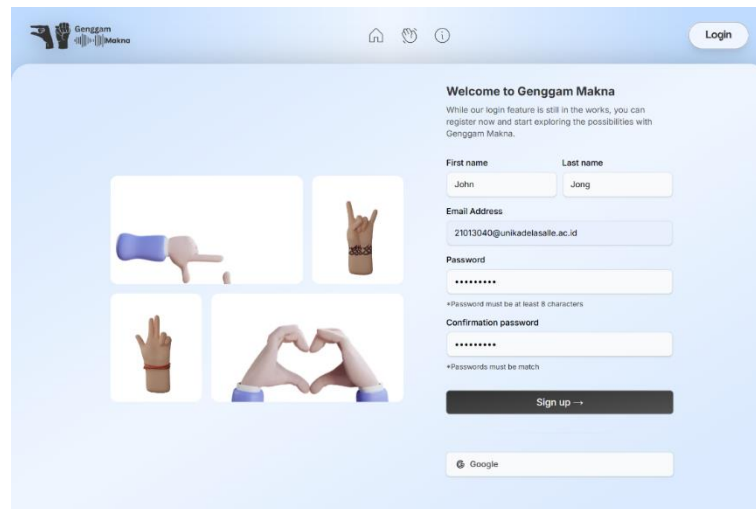
Gambar di atas merupakan tampilan hasil terjemahan yang dilakukan pada Google Chrome. Hasilnya tepat dan dapat disimpulkan bahwa aplikasi dapat dijalankan pada *browser* Google Chrome, serta sesuai dalam semua fungsionalitasnya.

#### **4.4.4.2 Pelaksanaan Pengujian Register dan *Login***

Pelaksanaan pengujian register dan *login* dilakukan pada *browser* Google Chrome. Pengujian akan dimulai dengan register akun baru dan dilanjutkan dengan proses *login* menggunakan akun yang sudah terdaftar. Proses dimulai dari mengakses halaman *login* dengan menekan tombol *Login* pada kanan atas halaman aplikasi.

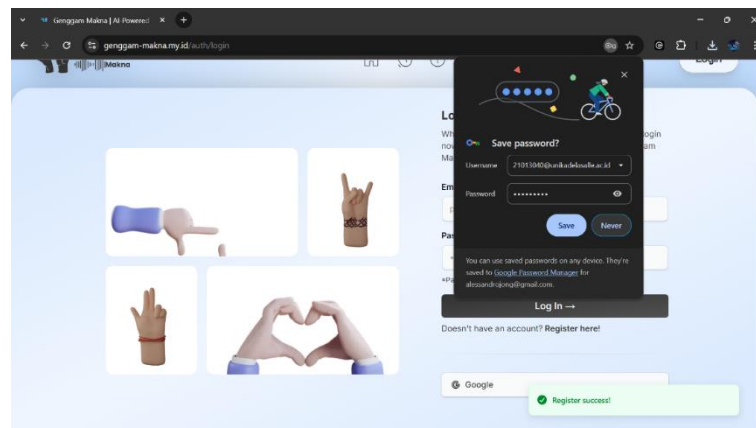


**Gambar 4.23 Mengakses Halaman *Login***



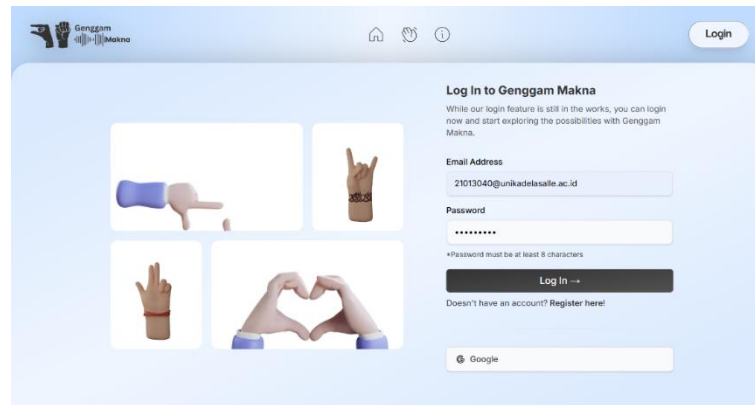
**Gambar 4.24 Mengisi Field Untuk Register**

Gambar di atas merupakan halaman register pengguna setelah menekan tombol *Register here*. *Field First name* dan *Last name* diisi dengan nama penulis. Kemudian *field* email diisi dengan email penulis. *Field* 'Password' dan 'Confirm password' diisi dengan *password*. Kemudian tombol *Sign up* ditekan.



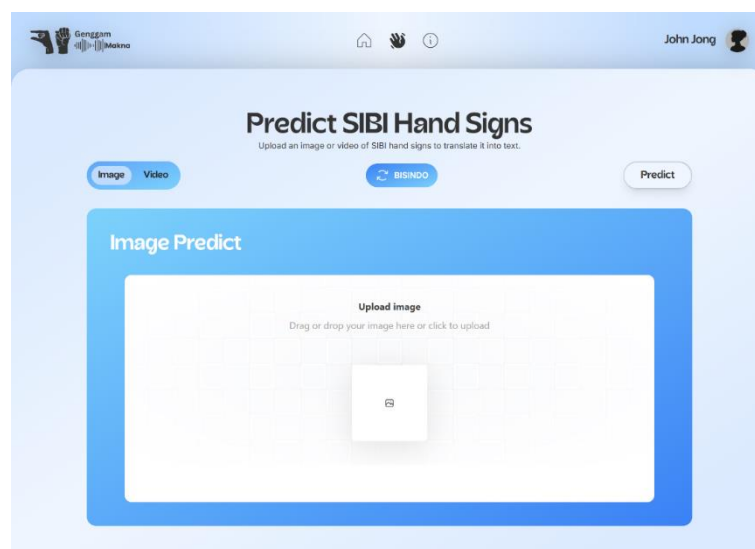
**Gambar 4.25 Berhasil Register Akun**

Gambar di atas adalah gambar hasil register setelah menekan tombol *Sign up* setelah mengisi semua *field* dengan jawaban yang valid.



**Gambar 4.26 Mengisi Field Login**

Gambar di atas merupakan halaman *login* yang muncul setelah melakukan proses register atau pendaftaran akun. *Field* email dan *field* password diisi sesuai akun yang terdaftar dan tekan tombol *Log in*.



**Gambar 4.27 Tampilan Setelah Berhasil Login**

Gambar di atas merupakan halaman yang muncul setelah melakukan *login* pada aplikasi. Nama dan profil akan muncul menggantikan tombol *Login* dan aplikasi sekarang dapat melakukan terjemahan bahasa isyarat.

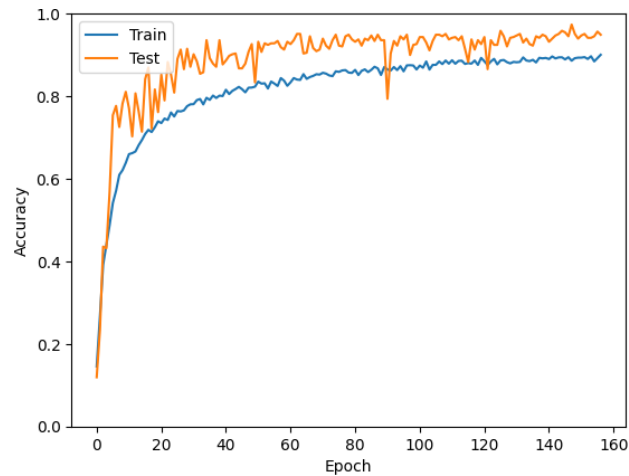
#### 4.4.4.3 Pelaksanaan Pengujian Model

Pelaksanaan pengujian model adalah pelaksanaan pengujian model *machine learning* yang dibangun untuk menilai kelayakan dan nilai dari suatu model

*machine learning*. Pengujian dilakukan terhadap masing-masing model bahasa isyarat, yaitu SIBI dan BISINDO.

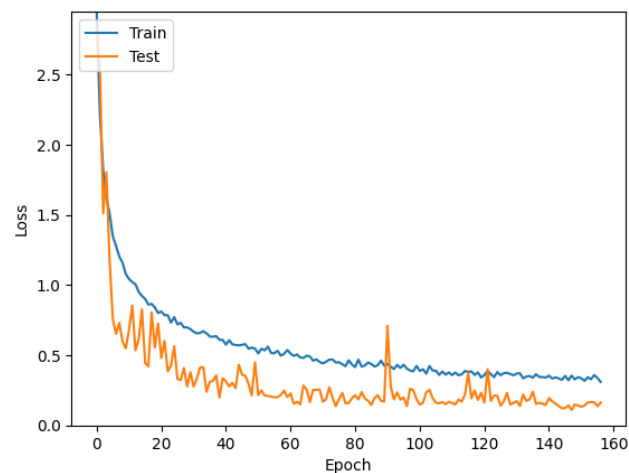
### 1. Pengujian Performa Model SIBI

Performa model SIBI berdasarkan dari akurasi *training* model yang dilakukan. Akurasi *training* terdiri dari dua yaitu akurasi *training* dan akurasi *testing* atau validasi. Akurasi digambarkan menggunakan Matplotlib dan dilakukan pada Google Colab.



**Gambar 4.28 Akurasi Model SIBI**

Akurasi *training* dan validasi relatif dekat dan hanya sedikit *overfitting* yang terjadi. Akurasi *training* adalah 90% dan akurasi validasi adalah 94%, kedua penggambaran akurasi ini dapat dilihat pada gambar diagram di atas berdasarkan tinggi akurasi dan jumlah *epoch* yang dilakukan. Selisih akurasi *training* dan validasi adalah 4%. *Training* model SIBI mencapai 157 *epoch*.

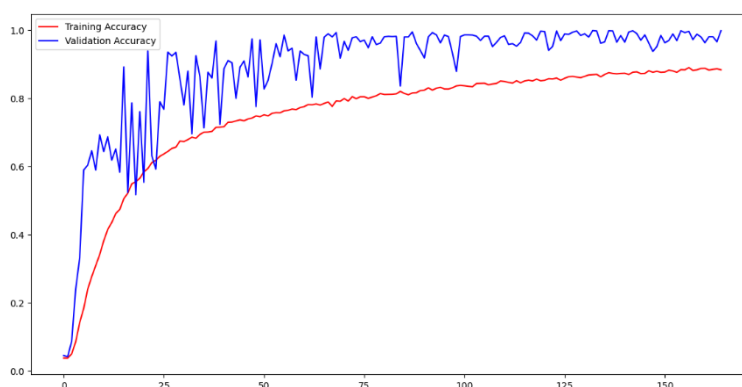


**Gambar 4.29 Loss Model SIBI**

Gambar grafik yang ditampilkan di atas menunjukkan *loss* terhadap *epoch* pada proses pelatihan model. *Loss* adalah metrik yang menunjukkan seberapa baik model memprediksi data dibandingkan dengan label sebenarnya. *Loss training* adalah 28% dan *loss testing* atau validasi adalah 16%, jumlah *loss* validasi lebih rendah dari *loss training*.

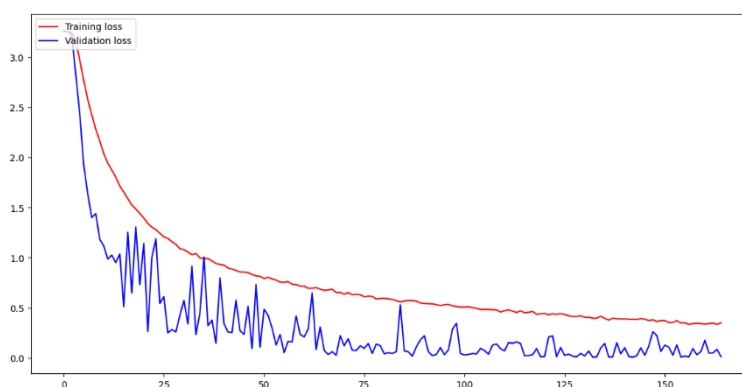
## 2. Pengujian Performa Model BISINDO

Performa model BISINDO dinilai menggunakan metrik yang sama dengan SIBI yaitu akurasi *training* dan akurasi *testing* atau validasi.



**Gambar 4.30 Akurasi Model BISINDO**

Akurasi *training* adalah 88% dan akurasi validasi adalah 99%, perbandingan akurasi ini dapat dilihat pada gambar diagram di atas berdasarkan tinggi akurasi dan jumlah *epoch* yang dilakukan. Selisih akurasi *training* dan validasi adalah 11%. *Training* model BISINDO mencapai 165 *epoch*.



**Gambar 4.31 Loss Model BISINDO**

Gambar grafik yang ditampilkan di atas menunjukkan *loss* model BISINDO. *Loss training* adalah 32% dan *loss testing* atau validasi adalah 1%, jumlah *loss* validasi jauh lebih rendah dari *loss training*.

### 3. Pengujian Penerjemahan Model


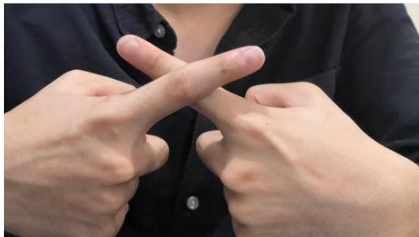


Pengujian model SIBI menggunakan berbagai gambar alfabet bahasa isyarat SIBI dan dilakukan untuk semua huruf kecuali huruf J dan Z.


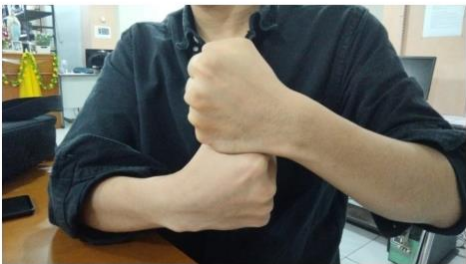



Gambar 4.32 Hasil Pengujian Model SIBI

Dari 24 alfabet yang dites, semuanya benar secara probabilitas yang terbesar, menghasilkan akurasi terjemahan 100%. Ada beberapa gambar yang memiliki probabilitas lumayan terbagi, seperti huruf M yang memiliki probabilitas kebenaran 39%. Hal ini terjadi karena ada kemiripan dari antar gambar dan alfabet bahasa isyarat, sehingga probabilitas antar alfabet lebih terbagi rata.

**Tabel 4.8 Hasil Pengujian Model BISINDO**

<b>Bahasa Isyarat</b>	<b>Kondisi</b>	<b>Hasil</b>
W (BISINDO)	Isyarat tepat dan minimal gangguan di <i>background</i>	<p>W: 100.00% H: 0.00% Y: 0.00%</p> 
X (BISINDO)	Isyarat kurang tepat dan terlalu <i>zoom</i>	<p>D: 29.07% X: 15.74% Q: 11.12%</p> 
Q (BISINDO)	Isyarat kurang tepat dan terlalu <i>zoom</i>	<p>P: 44.03% Q: 23.91% D: 17.36%</p> 
F (BISINDO)	Isyarat tepat namun ada gangguan tambahan tangan	<p>F: 74.29% T: 17.73% P: 4.86%</p> 

E (BISINDO)	Isyarat tepat dan <i>background</i> lumayan bersih	<p>E: 38.08% J: 20.55% Z: 12.80%</p> 
G (BISINDO)	Isyarat tepat dan <i>background</i> lumayan bersih	<p>G: 91.38% J: 2.43% F: 1.63%</p> 
A (BISINDO)	Isyarat tepat dan <i>background</i> lumayan bersih	<p>A: 76.78% O: 14.50% P: 3.40%</p> 

#### 4. Analisis Hasil Pengujian

Pengujian model SIBI menghasilkan hasil yang sangat memuaskan, di mana semua hasil terjemahan benar, walau ada beberapa persentase yang rendah. Hal ini berarti model berjalan dan bisa membedakan dengan baik, namun ada beberapa hal yang perlu ditegaskan. Yang pertama adalah posisi dan *background* dari gambar harus bersih sehingga model dapat menghasilkan terjemahan yang baik. Yang kedua adalah bentuk tangan saat melakukan bahasa isyarat, harus tepat dan tidak boleh salah, mengingat banyaknya kelas dan banyak kemiripan dari kulit, tekstur, dan bentuk tangan. Limitasi yang terjadi merupakan hasil di mana data untuk *training* model.

Pengujian model BISINDO memiliki beberapa hasil yang salah, tetapi wajar karena peragaan bahasa isyaratnya salah dan ada banyak *noise* di gambar yang dites. Model BISINDO memiliki keberagaman lebih besar daripada SIBI karena variasi datanya lebih banyak dan datanya lebih banyak daripada data SIBI. Model BISINDO memiliki limitasi yang sama di mana harus melakukan peragaan yang tepat dan memastikan *background* tidak terlalu kotor.

## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **5.1 Kesimpulan**

Berdasarkan proses pembangunan dan implementasi model pada aplikasi penerjemah bahasa isyarat menggunakan *machine learning*, maka dapat ditarik kesimpulan bahwa:

1. Semua *deliverables* dan spesifikasi persyaratan yang diidentifikasi pada tahap analisis sudah terpenuhi.
2. Aplikasi dapat diakses dan dijalankan dengan lancar pada *browser* Microsoft Edge dengan versi 131.0.2903.112, Mozilla Firefox dengan versi 128.0, dan Google Chrome versi 131.0.6778.205.
3. Model *machine learning* dapat melakukan terjemahan alfabet bahasa isyarat berdasarkan *input* gambar atau video alfabet bahasa isyarat. Model SIBI mendapat nilai akurasi *training* 90% dan validasi 94%. Model BISINDO mendapat nilai akurasi *training* 88% dan validasi 99%.
4. Kesalahan klasifikasi atau pengenalan terjadi jika bentuk alfabet bahasa isyarat kurang tepat diperagakan dan memiliki kemiripan besar antara satu gestur dengan yang lain.

#### **5.2 Saran**

Berdasarkan kesimpulan yang ditarik, ada beberapa saran untuk perbaikan dan pengembangan lebih lanjut di masa depan.

1. Menambahkan gestur angka dan gestur kata dari bahasa isyarat, agar nilai fungsi komunikasi menjadi semakin besar.
2. Menambahkan keberagaman kondisi dan lingkungan (*background*) bahasa isyarat sehingga dapat menerjemahkan isyarat dalam berbagai kondisi.
3. Mengimplementasikan aplikasi ke *android* sehingga dapat menggunakan fitur *live detection*.

## DAFTAR PUSTAKA

- [1] WHO, "Deafness and hearing loss," World Health Organization, [Online]. Available: [https://www.who.int/health-topics/hearing-loss#tab=tab\\_2](https://www.who.int/health-topics/hearing-loss#tab=tab_2). [Accessed 4 November 2024].
- [2] BPS, "Jumlah Penduduk Berumur 5 Tahun ke Atas menurut Kelompok Umur, Daerah Perkotaan/Perdesaan, Jenis Kelamin, dan Tingkat Kesulitan Mendengar, INDONESIA, Tahun 2022," Badan Pusat Statistik, [Online]. Available: <https://sensus.bps.go.id/topik/tabular/sp2022/145/0/0>. [Accessed 22 Desember 2024].
- [3] d. R. Fadli, "Wajib Tahu, Ini Perbedaan Tunarungu dan Tuli," Halodoc, 2 Desember 2022. [Online]. Available: [https://www.halodoc.com/artikel/wajib-tahu-ini-perbedaan-tunarungu-dan-tuli?srsId=AfmBOorhQJaAMtgznrCodtjItw7dAde7VD7OUL1uSMUh3h2\\_Q4GuzuA](https://www.halodoc.com/artikel/wajib-tahu-ini-perbedaan-tunarungu-dan-tuli?srsId=AfmBOorhQJaAMtgznrCodtjItw7dAde7VD7OUL1uSMUh3h2_Q4GuzuA). [Accessed 22 Desember 2024].
- [4] T. Hasna, "Pentingnya Bahasa Isyarat," TIga Serangkai, 13 Agustus 2024. [Online]. Available: <https://www.tigaserangkai.com/pentingnya-bahasa-isyarat/#:~:text=Bahasa%20Isyarat%20memiliki%20peran%20penting,tuna%20rungu%20dan%20tuna%20wicara..> [Accessed 17 Desember 2024].
- [5] d. M. D. C. Pane, "Penyebab Tuli konduktif dan Cara Mengatasinya," Alodokter, 24 Januari 2022. [Online]. Available: <https://www.alodokter.com/penyebab-tuli-konduktif-dan-cara-mengatasinya>. [Accessed 4 November 2024].
- [6] Z.-H. Zhou, Machine Learning, Nanjing: Springer, 2021.
- [7] K. Yasar, "image recognition," techtarget, Maret 2023. [Online]. Available: <https://www.techtarget.com/searchenterpriseai/definition/image-recognition>. [Accessed 4 November 2024].
- [8] IBM, "What are convolutional neural networks?," IBM, [Online]. Available: <https://www.ibm.com/think/topics/convolutional-neural-networks>. [Accessed 14 Desember 2024].
- [9] Kemendikbud, "Home | Merdeka Belajar - Kampus Merdeka," [Online]. Available: <https://kampusmerdeka.kemdikbud.go.id/>. [Accessed 14 November 2024].
- [10] Kemendikbud, "Kementerian Pendidikan dan Kebudayaan » Republik Indonesia," [Online]. Available: <https://www.kemdikbud.go.id/main/tentang-kemdikbud/sejarah-kemdikbudristek>. [Accessed 14 November 2024].
- [11] Kemendikbud, "Ketentuan Mitra & Program MSIB," Kampus Merdeka, [Online]. Available: <https://pusatinformasi.mitrakm.kemdikbud.go.id/hc/en-us/articles/5559233007129--Ketentuan-Mitra-Program-MSIB>. [Accessed 14 Desember 2024].
- [12] "Kementerian Pendidikan dan Kebudayaan » Republik Indonesia," Kementrian Pendidikan dan Kebudayaan, [Online]. Available:

- <https://www.kemdikbud.go.id/main/tentang-kemdikbud/visi-dan-misi>.  
[Accessed 14 Desember 2024].
- [13] "Kementerian Pendidikan dan Kebudayaan » Republik Indonesia," Kementerian Pendidikan dan Kebudayaan, [Online]. Available: <https://www.kemdikbud.go.id/main/informasi-publik/logo-kemdikbudristek>. [Accessed 14 Desember 2023].
- [14] "Kementerian Pendidikan dan Kebudayaan » Republik Indonesia," Kementerian Pendidikan dan Kebudayaan, [Online]. Available: <https://www.kemdikbud.go.id/main/informasi-publik/tugas-dan-fungsi>. [Accessed 14 Desember 2024].
- [15] I. Cianstury, F. H. Utami and Asnawati, "Sistem Informasi Pengajuan Judul Skripsi Berbasis Online Di Fakultas Ilmu Sosial," *Jurnal Ilmiah Binary STMIK Bina Nusantara Jaya Lubuklinggau*, vol. 5, no. 1, pp. 42-46, 2023.
- [16] A. A. Sendari, "Web adalah Bagian Penting dalam Internet, Kenali Pengertian dan Sejarahnya," *Liputan 6*, 21 Desember 2020. [Online]. Available: <https://www.liputan6.com/hot/read/4438604/web-adalah-bagian-penting-dalam-internet-kenali-pengertian-dan-sejarahnya?page=4>. [Accessed 14 Desember 2024].
- [17] A. Adnan, "Perbedaan BISINDO dan SIBI Saat Mempelajari Bahasa Isyarat," Telkom University, 17 Oktober 2024. [Online]. Available: <https://telkomuniversity.ac.id/perbedaan-bisindo-dan-sibi-saat-mempelajari-bahasa-isyarat/>. [Accessed 14 Desember 2024].
- [18] D. Intern, "Apa itu Machine Learning? Beserta Pengertian dan Cara Kerjanya," *Dicoding*, 19 Agustus 2020. [Online]. Available: <https://www.dicoding.com/blog/machine-learning-adalah/>. [Accessed 14 Desember 2024].
- [19] I. Goodfellow, Y. Bengio and A. Courville, *Deep Learning*, Cambridge: The MIT Press, 2016.
- [20] I. A. N, "Pengertian Data, Fungsi, Jenis-jenis, Manfaat dan Contohnya," Telkom University, 14 Desember 2023. [Online]. Available: <https://telkomuniversity.ac.id/pengertian-data-fungsi-jenis-jenis-manfaat-dan-contohnya/>. [Accessed 14 Desember 2024].
- [21] A. Jenks, E. Cowley and Q. Radich, "What is a machine learning model?," Microsoft, 17 April 2024. [Online]. Available: <https://learn.microsoft.com/en-us/windows/ai/windows-ml/what-is-a-machine-learning-model>. [Accessed 14 Desember 2024].
- [22] J. Murel Ph.D and E. Kavlakoglu, "What is transfer learning?," IBM, 12 Februari 2024. [Online]. Available: <https://www.ibm.com/topics/transfer-learning#:~:text=Transfer%20learning%20is%20a%20machine,improve%20generalization%20in%20another%20setting..> [Accessed 14 Desember 2024].
- [23] J. Wang and Y. Chen, *Introduction to Transfer Learning: Algorithms and Practice*, Springer, 2023.
- [24] R. G, "Everything you need to know about VGG16," *Medium*, 23 September 2021. [Online]. Available:

- <https://medium.com/@mygreatlearning/everything-you-need-to-know-about-vgg16-7315defb5918>. [Accessed 5 Januari 2024].
- [25] K. Le, "An overview of VGG16 and NiN models," Medium, 26 Maret 2021. [Online]. Available: <https://lekhuyen.medium.com/an-overview-of-vgg16-and-nin-models-96e4bf398484>. [Accessed 9 Januari 2025].
- [26] W. Setiawan, Deep Learning Menggunakan Convolutional Neural Network : Teori dan Aplikasi, Malang: MNC Publishing, 2022.
- [27] U. Michelucci, Advanced Applied Deep Learning: Convolutional Neural Networks and Object Detection 1st ed. Edition, Florence: Apress, 2019.
- [28] Google, "Klasifikasi: Akurasi, perolehan, presisi, dan metrik terkait," Developers Google, 11 Desember 2024. [Online]. Available: <https://developers.google.com/machine-learning/crash-course/classification/accuracy-precision-recall?hl=id>. [Accessed 15 Januari 2025].
- [29] Python, "What is Python? Executive Summary," Python, [Online]. Available: <https://www.python.org/doc/essays/blurb/>. [Accessed 14 Desember 2024].
- [30] T. Codecademy, "Apa itu Scikit-Learn?," codecademy, [Online]. Available: <https://www.codecademy.com/article/scikit-learn>. [Accessed 14 Desember 2024].
- [31] Numpy, "What is NumPy?," Numpy, 2024. [Online]. Available: <https://numpy.org/doc/stable/user/whatisnumpy.html>.
- [32] Matplotlib, "Matplotlib: Visualization with Python," Matplotlib, [Online]. Available: <https://matplotlib.org/>. [Accessed 14 Desember 2024].
- [33] G. Boesch, "What is OpenCV? The Complete Guide (2025)," Viso.AI, 1 Oktober 2024. [Online]. Available: <https://viso.ai/computer-vision/opencv/>. [Accessed 15 Desember 2024].
- [34] Wede, "Belajar Data Science : Apa yang dimaksud dengan Tensorflow dan Bagaimana Penggunaannya?," dqlab, 17 November 2020. [Online]. Available: <https://dqlab.id/belajar-data-science-pahami-tensflow>. [Accessed 14 Desember 2024].
- [35] R. Fahmi, "Tutorial Singkat Tentang Tensorflow.js," Medium, 25 Juli 2018. [Online]. Available: <https://medium.com/@rizafahmi22/screencast-singkat-tentang-tensorflow-js-7e7c3aa6506e>. [Accessed 14 Desember 2024].
- [36] K. LeViet, "Membuat dan men-deploy model deteksi objek kustom dengan TensorFlow Lite (Android)," Google, 3 Desember 2021. [Online]. Available: <https://developers.google.com/codelabs/tflite-object-detection-android?hl=id#0>. [Accessed 15 Desember 2024].
- [37] TensorFlow, "Keras: The high-level API for TensorFlow," TensorFlow, 6 Agustus 2023. [Online]. Available: <https://www.tensorflow.org/guide/keras>. [Accessed 14 Desember 024].
- [38] R. Fajri, "Google Colab: the power of the cloud for machine learning," DataScientest, 17 April 2024. [Online]. Available: <https://datascientest.com/en/google-colab-the-power-of-the-cloud-for-machine-learning>. [Accessed 14 Desember 2024].

- [39] C. Staff, "What Is Kaggle and What Is It Used For?," Coursera, 10 September 2024. [Online]. Available: <https://www.coursera.org/articles/kaggle>. [Accessed 14 Desember 2024].
- [40] T. Nolle, "pemrograman terstruktur (pemrograman modular)," TechTarget, Maret 2023. [Online]. Available: <https://www.techtarget.com/searchsoftwarequality/definition/structured-programming-modular-programming>. [Accessed 5 Januari 2025].
- [41] M. R. Ridho, "Pengertian Flowchart: Fungsi, Jenis, Simbol, dan Contohnya," Telkom University, 18 April 2024. [Online]. Available: <https://bee.telkomuniversity.ac.id/pengertian-flowchart-fungsi-jenis-simbol-dan-contohnya/>. [Accessed 15 Desember 2024].
- [42] "What is DFD(Data Flow Diagram)?," geeksforgeeks, 3 Januari 2025. [Online]. Available: <https://www.geeksforgeeks.org/what-is-dfddata-flow-diagram/>. [Accessed 5 Januari 2025].
- [43] A. D. Riski, "Analisis Model Waterfall : Pengertian, Tahapan, Kelebihan dan kekurangan," OSC, 27 Oktober 2022. [Online]. Available: <https://osc.medcom.id/community/analisis-model-waterfall-pengertian-tahapan-kelebihan-dan-kekurangan-4352>. [Accessed 15 Desember 2024].
- [44] Trivusi, "Mengenal 6 Jenis Loss Function pada Machine Learning," Trivusi, 17 September 2022. [Online]. Available: <https://www.trivusi.web.id/2022/08/loss-function.html>. [Accessed 15 Januari 2025].

**LAMPIRAN A**  
*USER ACCEPTANCE TEST*

Nama Penguji: Anisya Rengkuan			
Tipe Pengguna: Masyarakat			
No	Pertanyaan	Keterangan	
		Ya	Tidak
1.	Apakah aplikasi sudah sesuai dengan spesifikasi persyaratan?	/	
2.	Apakah alur aplikasi dan tampilan antarmuka mudah digunakan?	✓	
3.	Apakah aplikasi dapat melakukan proses autentikasi?	✓	
4.	Apakah aplikasi dapat <i>upload input</i> gambar dan video?	✓	
5.	Apakah aplikasi dapat memberikan hasil terjemahan yang akurat?	✓	

Saran:

Sudah baik namun bisa ditingkatkan kecepatan terjemah.

Manado, 13 Januari 2025



*USER ACCEPTANCE TEST*

Nama Penguji: Marchelino Raco			
Tipe Pengguna: Masyarakat			
No	Pertanyaan	Keterangan	
		Ya	Tidak
1.	Apakah aplikasi sudah sesuai dengan spesifikasi persyaratan?	✓	
2.	Apakah alur aplikasi dan tampilan antarmuka mudah digunakan?	/	
3.	Apakah aplikasi dapat melakukan proses autentikasi?	✓	
4.	Apakah aplikasi dapat <i>upload input</i> gambar dan video?	/	
5.	Apakah aplikasi dapat memberikan hasil terjemahan yang akurat?	✓	

Saran:

Proses autentikasi sudah berjalan baik, namun dapat ditingkatkan keamanannya dengan opsi autentikasi dua faktor.

Manado, 15 Januari 2025



*USER ACCEPTANCE TEST*

Nama Penguji: Uria Dutu			
Tipe Pengguna: Masyarakat			
No	Pertanyaan	Keterangan	
		Ya	Tidak
1.	Apakah aplikasi sudah sesuai dengan spesifikasi persyaratan?	✓	
2.	Apakah alur aplikasi dan tampilan antarmuka mudah digunakan?	✓	
3.	Apakah aplikasi dapat melakukan proses autentikasi?	✓	
4.	Apakah aplikasi dapat <i>upload input</i> gambar dan video?	✓	
5.	Apakah aplikasi dapat memberikan hasil terjemahan yang akurat?	✓	

Saran:

Antarmuka sudah mudah digunakan, tetapi dapat ditambahkan panduan singkat atau tutorial bagi pengguna baru.

Manado, 15 Januari 2025



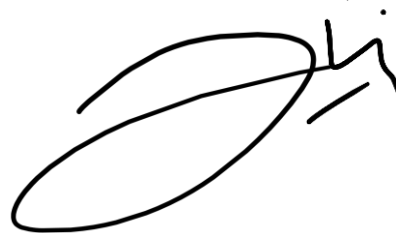
*USER ACCEPTANCE TEST*

Nama Penguji: Juan Karundeng			
Tipe Pengguna: Masyarakat			
No	Pertanyaan	Keterangan	
		Ya	Tidak
1.	Apakah aplikasi sudah sesuai dengan spesifikasi persyaratan?	✓	
2.	Apakah alur aplikasi dan tampilan antarmuka mudah digunakan?	✓	
3.	Apakah aplikasi dapat melakukan proses autentikasi?	✓	
4.	Apakah aplikasi dapat <i>upload input</i> gambar dan video?	✓	
5.	Apakah aplikasi dapat memberikan hasil terjemahan yang akurat?	✓	

Saran:

Fitur upload gambar dan video bekerja dengan baik, tetapi perlu ditinjau apakah ada cara untuk mempercepat proses unggahan atau mendukung file berukuran besar.

Manado, 15 Januari 2025



*USER ACCEPTANCE TEST*

Nama Penguji: Christian Kandori			
Tipe Pengguna: Masyarakat			
No	Pertanyaan	Keterangan	
		Ya	Tidak
1.	Apakah aplikasi sudah sesuai dengan spesifikasi persyaratan?	✓	
2.	Apakah alur aplikasi dan tampilan antarmuka mudah digunakan?	✓	
3.	Apakah aplikasi dapat melakukan proses autentikasi?	✓	
4.	Apakah aplikasi dapat <i>upload input</i> gambar dan video?	✓	
5.	Apakah aplikasi dapat memberikan hasil terjemahan yang akurat?	✓	

Saran:

Hasil terjemahan sudah akurat, namun kecepatan proses terjemahan dapat ditingkatkan untuk efisiensi pengguna.

Manado, 15 Januari 2025

