

**IMPLEMENTASI CONVOLUTIONAL NEURAL NETWORK (CNN)  
BERBASIS MOBILENET UNTUK KLASIFIKASI GERAKAN ANGKA  
PADA BAHASA ISYARAT BISINDO**

**PRAKTIK KERJA LAPANGAN**



**UNIVERSITAS  
MA CHUNG**

MICHAEL GOSALI TEDJOKUSUMA  
312210017

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI DAN DESAIN  
UNIVERSITAS MA CHUNG  
MALANG  
2025**

**LEMBAR PENGESAHAN  
PRAKTIK KERJA LAPANGAN**

**IMPLEMENTASI CONVOLUTIONAL NEURAL NETWORK (CNN)  
BERBASIS MOBILENET UNTUK KLASIFIKASI GERAKAN ANGKA  
PADA BAHASA ISYARAT BISINDO**

Oleh:  
**MICHAEL GOSALI TEDJOKUSUMA**  
**NIM. 31220017**

dari:  
**PROGRAM STUDI TEKNIK INFORMATIKA**  
**FAKULTAS TEKNOLOGI DAN DESAIN**  
**UNIVERSITAS MA CHUNG**

Dosen Pembimbing,



**Prof. Dr. Eng. Romy Budhi, ST., MT., M.Pd.**  
**NIP.20070035**

Dekan Fakultas Teknologi dan Desain,



**Prof. Dr. Eng. Romy Budhi, ST., MT., M.Pd.**  
**NIP.20070035**

## Kata Pengantar

Segala puji dan syukur penulis panjatkan kepada Tuhan Yang Maha Esa atas limpahan rahmat dan karunia-Nya, sehingga pelaksanaan kegiatan Praktik Kerja Lapangan (PKL) ini dapat berjalan dengan lancar. Laporan ini disusun sebagai salah satu syarat untuk menyelesaikan mata kuliah PKL bagi mahasiswa Program Studi Teknik Informatika, Universitas Ma Chung Malang.

Pada kesempatan ini, penulis menyampaikan rasa terima kasih yang sebesar-besarnya kepada semua pihak yang telah memberikan bantuan dan dukungan selama proses pelaksanaan PKL maupun dalam penyusunan laporan ini. Ucapan terima kasih secara khusus ditujukan kepada:

1. Bapak Dr. Eng. Romy Budhi, ST, MT., selaku dosen pembimbing sekaligus kepala peneliti dalam kegiatan PKL ini,
2. Rekan-rekan mahasiswa serta para responden atau subjek penelitian yang turut berpartisipasi dan memberikan dukungan,
3. Bapak Prof. Dr. Eng. Romy Budhi, ST., MT., M.Pd. dalam kapasitasnya sebagai Dekan Fakultas Sains dan Teknologi Universitas Ma Chung,
4. Bapak Hendry Setiawan, ST, M.Kom., selaku Ketua Program Studi Teknik Informatika.

Penulis telah berusaha menyusun laporan ini dengan sebaik mungkin. Namun, penulis menyadari bahwa laporan ini masih memiliki kekurangan. Semoga laporan PKL ini dapat memberikan manfaat dan informasi yang berguna bagi para pembaca.

Malang, 10 Mei 2025



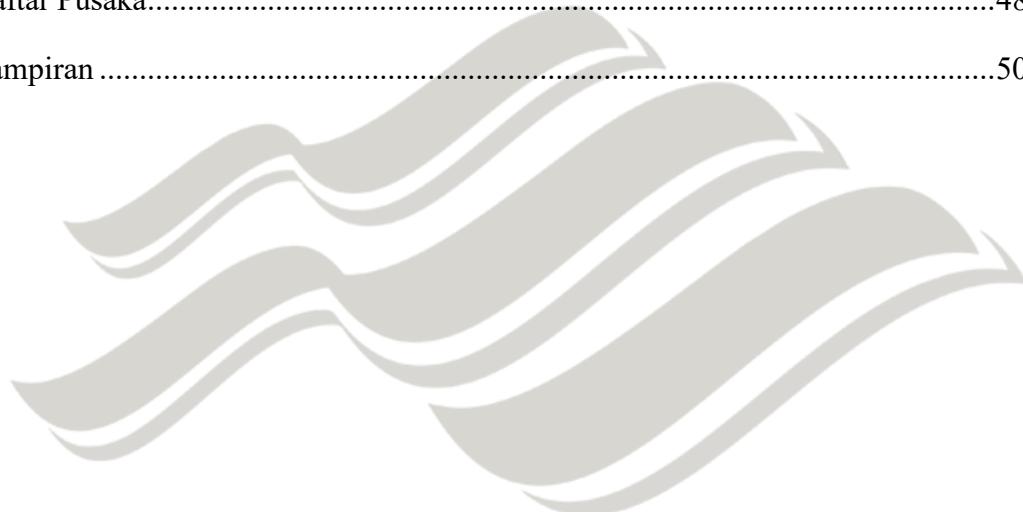
Michael Gosali Tedjokusuma

## Daftar Isi

Kata Pengantar .....	i
Daftar Isi.....	ii
Daftar Gambar.....	v
Daftar Tabel .....	vi
Bab I Pendahuluan.....	1
1.1. Latar Belakang .....	1
1.2. Identifikasi Masalah .....	2
1.3. Batasan Masalah.....	2
1.4. Rumusan Masalah .....	3
1.5. Tujuan Penelitian.....	3
1.6. Manfaat.....	3
Bab II Gambaran Umum Perusahaan.....	4
2.1. Universitas Ma Chung .....	4
2.1.1. Program Studi Teknik Informatika .....	6
2.2. Pusat Studi Human-Machine Interaction .....	7
Bab III Tinjauan Pustaka .....	9
3.1. Bahasa Isyarat.....	9
3.1.1. BISINDO (Bahasa Isyarat Indonesia).....	9
3.1.2. SIBI (Sistem Isyarat Bahasa Indonesia).....	10
3.2. Convolutional Neural Network (CNN) dan Arsitekturnya .....	10
3.3. Keras dan TensorFlow.....	13
3.3.1. NumPy .....	14
3.3.2. Pandas .....	15
3.3.3. OpenCV .....	15

3.3.4. Matplotlib.....	16
3.3.5. Seaborn.....	16
3.3.6. SciPy .....	16
3.3.7. Scikit-learn .....	17
3.3.8. Pickle.....	17
3.4. Pengujian Model Klasifikasi .....	18
3.4.1. Akurasi (Accuracy) .....	18
3.4.2. Presisi (Precision) .....	19
3.4.3. Recall.....	19
3.4.4. F1-Score .....	20
3.4.5. Confusion Matrix .....	20
Bab IV Deskripsi Data dan Hasil Praktik Kerja Lapangan.....	22
4.1. Alur Pengerjaan Penelitian .....	22
4.2. MobileNetV1 .....	23
4.2.2. Implementasi Arsitektur Model .....	25
4.3. Dataset .....	26
4.3.1. Pengambilan Data .....	27
4.3.2. Data Split.....	28
4.3.3. Preprocessing Data.....	28
4.3.4. Augmentation Data .....	28
4.4. Pembentukan Model Klasifikasi.....	29
4.5. Hasil Desain Model .....	31
4.5.1. Input Model.....	32
4.5.2. Preprocessing .....	32
4.5.3. Proses Pembuatan Model Klasifikasi.....	32
4.5.4. Evaluasi Model.....	34

4.5.5. Evaluasi Data Baru.....	38
4.6. Diskusi.....	41
4.6.1. Evaluasi Data Baru ke-2.....	42
Bab V Penutup.....	46
5.1. Simpulan.....	46
5.2. Saran .....	47
Daftar Pusaka.....	48
Lampiran .....	50



UNIVERSITAS  
**MA CHUNG**

## Daftar Gambar

Gambar 3. 1 Gestur angka 1-10 dalam BISINDO .....	10
Gambar 3. 2 Pemahaman Dasar CNN.....	11
Gambar 3. 3 Contoh <i>Confusion Matrix</i> .....	20
Gambar 4. 1 Skema Alur Pengerjaan Penelitian.....	22
Gambar 4. 3 1x1 Convolution.....	23
Gambar 4. 2 Depthwise Convolution.....	23
Gambar 4. 4 Proses Mengimpor model <i>MobileNet Pre-Trained</i> sebagai <i>Feature Extractor</i> .....	24
Gambar 4. 5 Proses pembekuan sebagian besar layer.....	24
Gambar 4.6 Proses <i>Transfer Learning</i> dan <i>Fine-Tuning</i> .....	25
Gambar 4. 7 Proses Kompilasi.....	26
Gambar 4. 8 Contoh gestur angka 7 (Sumber : Alexander, 2023).....	27
Gambar 4. 9 Sekema Pembagian Data .....	28
Gambar 4. 10 Proses Data Augmentation .....	31
Gambar 4. 11 <i>Confusion Matrix</i> MobileNet .....	34
Gambar 4. 12 Grafik Accuracy dan Loss.....	38
Gambar 4. 14 Kumpulan sampel subjek .....	41
Gambar 4. 15 <i>Data Replacement</i> .....	43

## **Daftar Tabel**

Tabel 4. 1 <i>Classification Report CNN</i> .....	35
Tabel 4. 2 Hasil <i>accuracy</i> menggunakan data baru berdasarkan subjek.....	39
Tabel 4. 3 Hasil <i>accuracy</i> menggunakan data baru berdasarkan gestur.....	40
Tabel 4. 4 Hasil <i>accuracy</i> menggunakan data baru ke-2 berdasarkan subjek .....	43
Tabel 4. 5 Hasil <i>accuracy</i> menggunakan data baru ke-2 berdasarkan gestur.....	44



UNIVERSITAS  
**MA CHUNG**

## Bab I

### Pendahuluan

#### 1.1. Latar Belakang

Di Indonesia, terdapat dua bentuk bahasa isyarat yang digunakan oleh penyandang tunarungu dan tunawicara, yaitu Bahasa Isyarat Indonesia (BISINDO) dan Sistem Isyarat Bahasa Indonesia (SIBI). Dalam penelitian ini, fokus utama akan diberikan pada penggunaan BISINDO. Sebagai bentuk bahasa visual, BISINDO mengandalkan gerakan tangan, mimik wajah, dan posisi tubuh dalam penyampaiannya, sehingga cukup sulit dipahami oleh orang awam yang tidak terbiasa menggunakannya. Kondisi ini menimbulkan hambatan dalam komunikasi antara penyandang disabilitas dengan masyarakat sekitarnya. Oleh karena itu, diperlukan pengembangan teknologi yang mampu menerjemahkan bahasa isyarat secara otomatis guna mendukung terciptanya komunikasi yang lebih inklusif.

Untuk mengatasi hambatan tersebut, teknologi kecerdasan buatan seperti deep learning telah banyak dimanfaatkan. Sebagai bagian dari pembelajaran mesin, deep learning memiliki kemampuan dalam mengekstraksi pola fitur yang kompleks dari data visual, seperti gambar maupun video, secara otomatis tanpa memerlukan teknik rekayasa fitur secara manual (LeCun et al., 2015). Dalam konteks pengenalan bahasa isyarat, model berbasis Convolutional Neural Network (CNN) terbukti efektif karena mampu mengenali pola spasial dari gambar melalui tahapan konvolusi dan pooling secara bertingkat (Krizhevsky et al., 2012).

Salah satu arsitektur CNN yang dikenal karena efisiensinya adalah MobileNetV1. Arsitektur ini dirancang untuk digunakan pada perangkat dengan sumber daya terbatas seperti smartphone atau sistem tertanam. Dengan menggunakan metode *depthwise separable convolution*, MobileNetV1 mampu mengurangi jumlah parameter serta beban komputasi, tanpa menurunkan performa akurasi secara signifikan (Howard et al., 2017). Oleh karena itu, MobileNetV1 sangat cocok diterapkan pada sistem pengenalan gerakan tangan berbasis bahasa isyarat secara portabel dan waktu nyata.

Melalui penerapan deep learning, khususnya CNN dengan arsitektur MobileNetV1, penelitian ini bertujuan untuk mengembangkan sistem klasifikasi gerakan angka dalam BISINDO. Sistem ini diharapkan dapat membantu menerjemahkan isyarat tangan ke dalam informasi yang dapat dimengerti oleh masyarakat umum, sehingga mendukung terciptanya komunikasi yang lebih inklusif bagi penyandang disabilitas.

## **1.2. Identifikasi Masalah**

Komunikasi merupakan aspek penting dalam kehidupan sosial manusia. Namun, penyandang tunarungu dan tunawicara sering menghadapi kendala dalam berkomunikasi dengan masyarakat umum karena perbedaan dalam sistem bahasa yang digunakan, yaitu bahasa isyarat. Bahasa Isyarat Indonesia (BISINDO), sebagai salah satu bentuk utama komunikasi visual bagi komunitas tunarungu di Indonesia, belum dipahami secara luas oleh masyarakat awam. Hal ini mengakibatkan adanya kesenjangan komunikasi antara penyandang disabilitas dan lingkungan sosialnya.

Di sisi lain, teknologi pengenalan citra berbasis kecerdasan buatan telah menunjukkan potensi besar dalam mendeteksi dan mengklasifikasikan pola visual seperti gerakan tangan. Namun, penerapan teknologi ini dalam konteks pengenalan bahasa isyarat, khususnya untuk mengenali gerakan angka dalam BISINDO, masih terbatas. Tantangan utama terletak pada kebutuhan akan model yang ringan, efisien, dan akurat agar dapat diimplementasikan pada perangkat dengan sumber daya terbatas seperti smartphone.

## **1.3. Batasan Masalah**

- a) Penelitian ini hanya berfokus pada pengenalan dan klasifikasi gerakan angka (0–10) dalam Bahasa Isyarat Indonesia (BISINDO).
- b) Citra gerakan tangan yang digunakan bersumber dari dataset citra statis dan tidak mencakup isyarat dinamis atau sekuensial.
- c) Implementasi model dilakukan menggunakan arsitektur Convolutional Neural Network (CNN) berbasis MobileNetV1.

- d) Pengujian dan evaluasi sistem dilakukan dalam lingkungan terkontrol, tanpa melibatkan data real-time dari kamera secara langsung.
- e) Bahasa isyarat yang digunakan adalah BISINDO dan tidak mencakup sistem bahasa isyarat lain seperti SIBI atau ASL.

#### **1.4. Rumusan Masalah**

- a) Bagaimana membangun model klasifikasi citra yang mampu mengenali gerakan angka dalam BISINDO secara akurat?
- b) Apakah arsitektur CNN berbasis MobileNetV1 efektif dan efisien dalam melakukan klasifikasi citra gerakan tangan bahasa isyarat?
- c) Bagaimana performa sistem klasifikasi yang dibangun dalam mengenali gerakan angka BISINDO berdasarkan metrik evaluasi seperti akurasi dan confusion matrix?

#### **1.5. Tujuan Penelitian**

- a) Mengembangkan sistem klasifikasi gerakan angka pada Bahasa Isyarat Indonesia (BISINDO) menggunakan pendekatan deep learning.
- b) Menerapkan arsitektur MobileNetV1 dalam rangka menghasilkan model yang ringan namun tetap akurat untuk pengenalan gerakan tangan.
- c) Mengevaluasi performa sistem klasifikasi menggunakan metrik evaluasi seperti akurasi, precision, recall, dan confusion matrix.

#### **1.6. Manfaat**

- a) Memberikan kontribusi dalam pengembangan teknologi pengenalan bahasa isyarat berbasis deep learning.
- b) Menjadi referensi untuk penelitian selanjutnya dalam pengembangan sistem klasifikasi citra dengan model efisien seperti MobileNet.
- c) Membantu masyarakat umum dalam mengenali gerakan angka dalam BISINDO secara otomatis.
- d) Mendorong terciptanya komunikasi yang lebih inklusif antara penyandang disabilitas dan masyarakat luas melalui teknologi pengenalan bahasa isyarat.

## **Bab II**

### **Gambaran Umum Perusahaan**

#### **2.1. Universitas Ma Chung**

Universitas Ma Chung merupakan perguruan tinggi swasta yang terletak di Kota Malang dan berada di bawah pengelolaan Yayasan Harapan Bangsa Sejahtera. Diresmikan pada tanggal 7 Juli 2007, kampus ini berlokasi di Villa Puncak Tidar N-01, Karang Widoro, Dau, Kabupaten Malang, Jawa Timur. Universitas ini memiliki visi untuk memuliakan Tuhan Yang Maha Esa melalui pembentukan karakter, pengembangan ilmu pengetahuan, serta memberikan kontribusi nyata sebagai insan akademis yang kreatif dan inovatif. Adapun misi yang dijalankan mencakup enam poin utama berikut:

1. Menyelenggarakan Tri Dharma Perguruan Tinggi—pendidikan, penelitian, dan pengabdian masyarakat—with standar tinggi, fokus, serta relevan dengan perkembangan zaman.
2. Membentuk generasi pemimpin dan penggerak masyarakat yang memiliki integritas, jiwa kepemimpinan, dan kewirausahaan, dengan menekankan nilai-nilai karakter mulia, kerendahan hati, semangat pelayanan, serta peran sebagai insan paripurna.
3. Mendorong pola pikir kritis, prinsipil, dan kreatif-realistik yang dilandasi oleh nurani luhur.
4. Menghasilkan lulusan berkualitas yang siap menghadapi dunia kerja dan mampu bersaing secara global.
5. Berperan aktif dalam kemajuan peradaban global dengan mencetak alumni yang berpikiran terbuka, toleran, cinta damai, dan produktif dalam menciptakan karya yang berdampak positif bagi kemanusiaan.
6. Menjalankan sistem tata kelola perguruan tinggi dengan prinsip efisiensi dan transparansi.

Universitas Ma Chung juga memiliki landasan filosofis berupa 12 nilai dasar:

1. Orisinal

Keyakinan bahwa setiap individu memiliki keunikan tersendiri sehingga

harus berani tampil autentik tanpa meniru orang lain. Seluruh civitas akademika didorong untuk mengedepankan keorisinilan dalam setiap karya dan inisiatif.

**2. Terpercaya**

Menjunjung tinggi kejujuran dalam pemikiran, tindakan, dan perkataan untuk membangun institusi dan komunitas akademik yang kredibel, terhormat, andal, dan dapat dipercaya.

**3. Gigih**

Keyakinan bahwa kesabaran dan ketekunan dapat mengatasi segala hambatan dan masalah. Komitmen untuk membangun budaya pantang menyerah, tekun, tidak mengenal lelah, dan tidak mudah putus asa.

**4. Kreatif**

Mendorong terciptanya budaya kerja yang inovatif, produktif, dan imajinatif untuk senantiasa mengembangkan pendekatan dan metode baru dalam bekerja dan meraih kesuksesan.

**5. Dinamis**

Komitmen menciptakan lingkungan kerja dan pembelajaran yang selalu hidup, bersemangat, dan aktif sehingga memungkinkan seluruh civitas akademika untuk mengantisipasi, beradaptasi, dan mengakomodasi perubahan.

**6. Ramah dan Menyenangkan**

Mendorong terciptanya lingkungan kerja dan pembelajaran yang tertib, penuh kegembiraan dan menyenangkan untuk menghasilkan SDM yang ramah, toleran, pembawa kedamaian, dan sukacita.

**7. Meritokratik**

Menghargai prestasi, kerja keras, dan kontribusi nyata dengan prinsip bahwa pencapaian menentukan posisi seseorang.

**8. Profesional**

Komitmen membangun etos kerja yang selalu mengutamakan kualitas dan motivasi untuk menjadi yang terbaik dalam setiap upaya.

#### 9. Bertanggung Jawab

Mendorong seluruh civitas akademika untuk mampu mempertanggungjawabkan semua pemikiran, tindakan, dan perkataan secara baik dan benar.

#### 10. Sinergi

Mendorong kemampuan untuk mempertanggungjawabkan semua pemikiran, tindakan, dan perkataan dengan baik dan benar.

#### 11. Rendah Hati

Mendorong sikap untuk selalu mampu mempertanggungjawabkan semua pemikiran, tindakan, dan perkataan dengan baik dan benar.

#### 12. Kewarganegaraan

Berperan proaktif dalam memberikan kontribusi untuk membangun masyarakat dan lingkungan hidup yang aman, sehat, damai, sejahtera, adil, dan makmur.

Saat ini, Universitas Ma Chung memiliki 11 program studi, antara lain: Manajemen Bisnis, Akuntansi, Magister Manajemen Inovasi, Sastra Inggris, Pendidikan Bahasa Mandarin, Teknik Informatika, Sistem Informasi, Desain Komunikasi Visual, Teknik Industri, Optometri, serta Farmasi dan Profesi Apoteker.

#### 2.1.1. Program Studi Teknik Informatika

Program Studi Teknik Informatika di Universitas Ma Chung dirancang untuk memenuhi kebutuhan industri teknologi informasi masa kini. Pembelajaran disusun secara berkualitas dan didukung oleh fasilitas yang memadai guna menunjang proses akademik. Mahasiswa memiliki kesempatan untuk mengikuti program magang industri dan Kampus Berdampak yang memberikan pengalaman praktis langsung di lapangan.

Terdapat dua jalur spesialisasi dalam program ini:

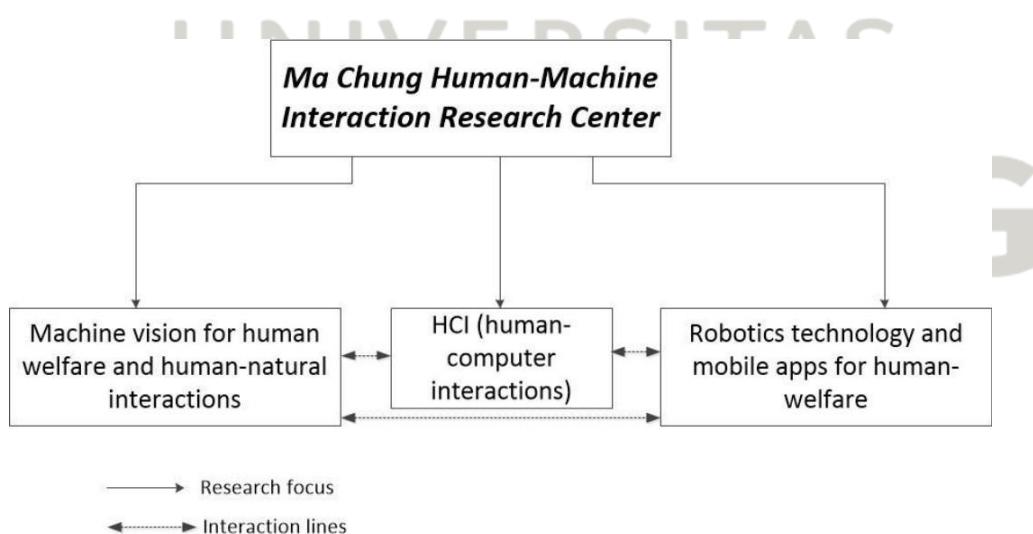
- Sistem Cerdas, yang fokus pada pengembangan kecerdasan buatan (AI).
- Sistem Komputer, yang menekankan pembelajaran mengenai jaringan perangkat cerdas dan sistem yang saling terhubung.

Kegiatan Praktik Kerja Lapangan menjadi bagian esensial dari kedua jalur tersebut dan merupakan bagian dari kegiatan riset berbasis Kampus Berdampak Universitas Ma Chung.

## 2.2. Pusat Studi Human-Machine Interaction

Pusat Studi Human-Machine Interaction (HMI) dibentuk pada 11 September 2019 di bawah naungan Program Studi Teknik Informatika Universitas Ma Chung. Pusat studi ini bertujuan untuk meneliti dan mengembangkan teknologi dalam konteks interaksi manusia dan mesin. Berlokasi di lantai 6 Gedung R&D, pusat studi ini memiliki tiga fokus utama:

- a) *Machine vision for human welfare and human-natural interactions* Area ini mengkaji penerapan teknologi visi mesin untuk mendukung kesejahteraan manusia serta interaksi natural manusia.
- b) *Human-computer interactions (HCI)* Area ini melakukan riset mengenai perancangan, implementasi, dan evaluasi perangkat atau *User Interface (UI)* dan *User Experience (UX)*.
- c) *Robotics technology and mobile apps for human welfare* Area ini meneliti implementasi teknologi robotika dalam meningkatkan kesejahteraan manusia.



Gambar 2. 1 Diagram bidang fokus riset pusat studi *Human-Machine Interaction*

Pada Gambar 2.1 ditunjukkan diagram bidang fokus riset dari Ma Chung Human-Machine Interaction Research Center. Pusat studi ini memiliki tiga bidang utama yang menjadi fokus pengembangannya, yaitu: *Machine Vision for Human Welfare and Human-Natural Interactions*, *Human-Computer Interaction (HCI)*, serta *Robotics Technology and Mobile Apps for Human Welfare*. Masing-masing bidang ini saling terhubung melalui garis interaksi (*interaction lines*), yang menunjukkan bahwa pendekatan riset dilakukan secara integratif dan lintas bidang. Fokus utama dari pusat studi ini adalah menciptakan teknologi yang mampu meningkatkan kualitas hidup manusia melalui interaksi yang lebih alami dan efisien antara manusia dan mesin. Arah panah solid dalam diagram menggambarkan fokus penelitian utama dari masing-masing subbidang, sedangkan panah putus-putus menandakan adanya kolaborasi dan keterkaitan antara satu subbidang dengan lainnya dalam mendukung tujuan besar pengembangan *Human-Machine Interaction* yang berorientasi pada kesejahteraan manusia.

Berikut beberapa riset dan produk yang telah dikembangkan oleh Pusat Studi Human-Machine Interaction (HMI) di Universitas Ma Chung:

1. Mouse Difabel untuk tunadaksa lengan atas  
Alat bantu komputer untuk tunadaksa tanpa lengan bawah, terdiri dari modul arm (penggerak kursor) dan modul foot (klik dan scroll). Kini dalam tahap hilirisasi industri (Universitas Ma Chung, 2024).
2. Sistem deteksi bahasa isyarat berbasis Machine Learning (BISINDO)  
Sistem berbasis *machine learning* untuk mengenali gerakan tangan pengguna tunarungu dan menerjemahkannya menjadi teks atau suara (Malang Pariwara, 2024).
3. Riset digitalisasi dan pemetaan penyandang disabilitas  
Proyek ini bertujuan membangun sistem pendataan digital yang komprehensif untuk membantu organisasi seperti LINKSOS bekerja lebih efektif di tingkat desa dan kelurahan di wilayah Jawa Timur (Lingkar Sosial Indonesia, 2022b).

## **Bab III**

### **Tinjauan Pustaka**

#### **3.1. Bahasa Isyarat**

Bahasa isyarat merupakan sistem komunikasi visual yang digunakan oleh penyandang tunarungu dan tunawicara untuk menyampaikan informasi melalui kombinasi gerakan tangan, ekspresi wajah, dan posisi tubuh. Berbeda dengan bahasa verbal, bahasa isyarat tidak mengandalkan suara melainkan menggunakan elemen visual dan kinestetik sebagai sarana utama dalam berkomunikasi.

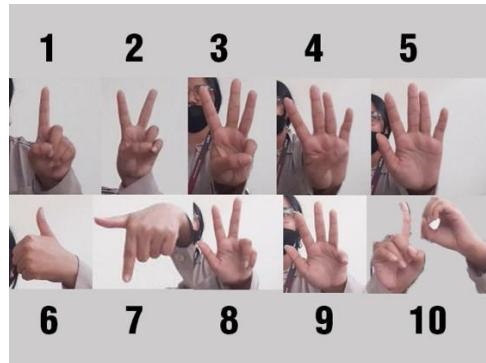
Di Indonesia, terdapat dua jenis bahasa isyarat yang umum digunakan, yaitu Bahasa Isyarat Indonesia (BISINDO) dan Sistem Isyarat Bahasa Indonesia (SIBI). Meskipun keduanya bertujuan memfasilitasi komunikasi bagi komunitas disabilitas pendengaran, kedua sistem ini memiliki perbedaan yang signifikan dalam hal struktur dan penggunaannya.

##### **3.1.1. BISINDO (Bahasa Isyarat Indonesia)**

BISINDO merupakan bahasa isyarat alami yang berkembang secara organik di dalam komunitas tunarungu di Indonesia. Bahasa ini tidak mengikuti struktur tata bahasa Indonesia secara ketat, melainkan lebih bebas, ekspresif, dan visual. BISINDO memiliki variasi regional yang dipengaruhi oleh budaya lokal masing-masing daerah. Elemen komunikasi dalam BISINDO tidak hanya mencakup gerakan tangan, tetapi juga ekspresi wajah dan arah pandangan, yang semuanya penting dalam membentuk makna isyarat (Syahputra et al., 2021).

Dalam penelitian ini, BISINDO menjadi fokus utama karena lebih umum digunakan dalam kehidupan sehari-hari oleh komunitas tunarungu. Penelitian ini secara khusus membahas gerakan angka 0 hingga 10 dalam BISINDO seperti yang

dapat dilihat pada Gambar 3.1, yang menjadi bagian penting dalam komunikasi numerik.



Gambar 3. 1 Gestur angka 1-10 dalam BISINDO

### 3.1.2. SIBI (Sistem Isyarat Bahasa Indonesia)

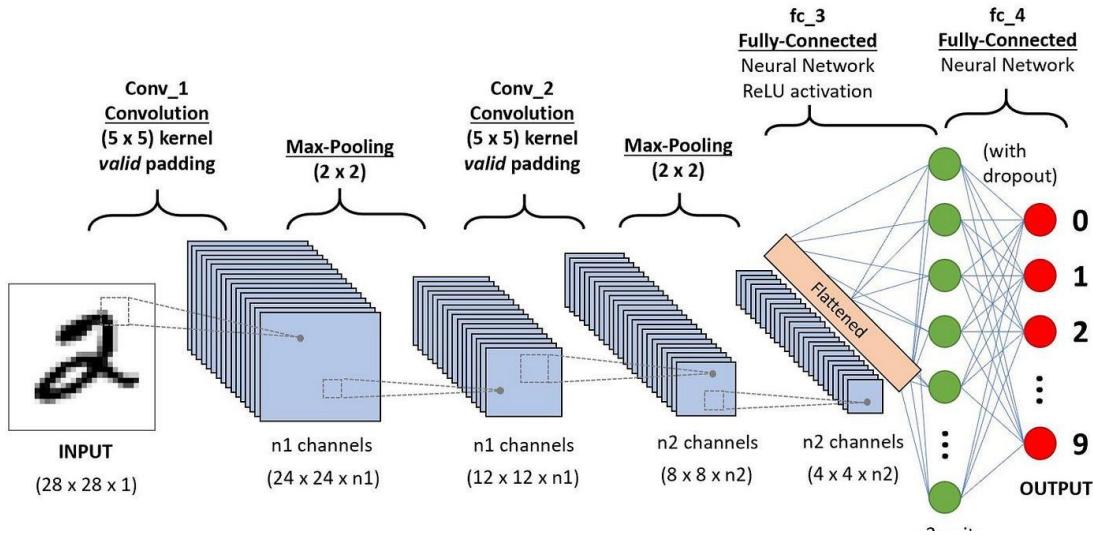
SIBI (Sistem Isyarat Bahasa Indonesia) merupakan alat bantu komunikasi bagi individu tunarungu yang menggabungkan unsur bahasa lisan, gerakan isyarat, ekspresi wajah, dan gerak tubuh lainnya. Pemerintah menetapkan SIBI sebagai bahasa isyarat resmi yang digunakan di Sekolah Luar Biasa (SLB). Namun, banyak penyandang tunarungu merasa bahwa SIBI tidak sepenuhnya mewakili cara berkomunikasi mereka, karena SIBI menggunakan aturan isyarat yang cenderung menyesuaikan dengan struktur bahasa lisan dalam menyampaikan kosakata (Nugraheni et al., 2021).

## 3.2. Convolutional Neural Network (CNN) dan Arsitekturnya

*Convolutional Neural Network* (CNN) merupakan salah satu arsitektur jaringan saraf tiruan dalam bidang *deep learning* yang telah terbukti sangat efektif dalam mengolah data visual seperti citra maupun video. CNN dirancang untuk meniru cara kerja sistem penglihatan manusia dalam mengenali pola, dengan kemampuan melakukan ekstraksi fitur secara otomatis dari citra masukan. Proses ini dilakukan melalui lapisan-lapisan berjenjang seperti *convolutional layer*, *activation layer*, dan *pooling layer*. Lapisan konvolusi bertugas mengekstraksi fitur lokal dari gambar menggunakan filter atau kernel yang melakukan pemindaian ke seluruh citra untuk menghasilkan *feature map* yang menyoroti pola-pola penting seperti tepi, sudut, dan

tekstur. Selanjutnya, fungsi aktivasi seperti *ReLU* memperkenalkan sifat non-linear ke dalam jaringan sehingga memungkinkan pembelajaran fitur yang lebih kompleks. Sementara itu, *pooling layer* berfungsi untuk mereduksi dimensi spasial dari data fitur agar lebih efisien dan mengurangi risiko *overfitting* (Krizhevsky et al., 2012).

Pada Gambar 3.2 tergambar arsitektur dasar dari sebuah *Convolutional*



Gambar 3. 2 Struktur Dasar CNN

(Sumber : <https://medium.com/data-folks-indonesia/pemahaman-dasar-convolutional-neural-networks-bfa1bf0b06e1>)

*Neural Network* (CNN), yang merupakan model *deep learning* yang secara khusus dirancang untuk mengenali pola dari data citra. Proses dimulai dari gambar input berukuran  $28 \times 28$  piksel, yang kemudian diproses melalui serangkaian lapisan konvolusi (*convolution layer*). Lapisan ini menggunakan kernel berukuran  $5 \times 5$  untuk mengekstraksi fitur penting seperti tepi dan sudut dari gambar. Setelah itu, hasilnya diperkecil melalui *max-pooling layer* berukuran  $2 \times 2$ , yang bertujuan mengurangi dimensi data sambil mempertahankan fitur yang paling menonjol. Proses konvolusi dan pooling ini diulang beberapa kali untuk menghasilkan representasi fitur yang lebih kompleks dan abstrak. Setelah melewati semua lapisan konvolusi dan pooling, data yang dihasilkan kemudian di-flatten menjadi vektor satu dimensi agar bisa diproses oleh *fully connected layer* (*dense layer*). Lapisan *fully connected* ini menggunakan

fungsi aktivasi seperti *ReLU* dan *dropout* untuk mencegah *overfitting*, dan akhirnya mengarah ke output layer yang berfungsi mengklasifikasikan input ke dalam 11 kelas.

Namun, dalam implementasi model yang digunakan pada penelitian ini, tidak digunakan lagi *flatten layer* maupun *fully connected layer* awal seperti pada arsitektur CNN konvensional. Sebagai gantinya, digunakan lapisan *GlobalAveragePooling2D()* setelah *MobileNetV1* sebagai *feature extractor*. *Global Average Pooling* bertujuan mereduksi *output* dari fitur spasial menjadi satu vektor per channel tanpa perlu meratakan seluruh volume data seperti *flatten*, sehingga lebih efisien secara komputasi dan lebih tahan terhadap *overfitting*. Selain itu, pendekatan ini juga sejalan dengan arsitektur modern seperti *MobileNetV1* yang memang dirancang ringan dan efisien. Oleh karena itu, penggunaan *GlobalAveragePooling2D()* merupakan strategi yang tepat dalam menyederhanakan representasi fitur dari citra masukan sebelum diteruskan ke dense layer klasifikasi, tanpa kehilangan informasi penting yang relevan.

Seiring meningkatnya kompleksitas model dan ukuran dataset dalam berbagai aplikasi modern, kebutuhan terhadap arsitektur CNN yang ringan dan efisien menjadi semakin mendesak. Hal ini terutama penting bagi aplikasi yang dijalankan pada perangkat terbatas sumber daya seperti *smartphone*, tablet, atau *embedded system*. Dalam konteks penelitian ini, arsitektur yang dipilih adalah *MobileNetV1*, yaitu model CNN yang dirancang secara khusus untuk perangkat dengan keterbatasan komputasi dan konsumsi daya rendah. *MobileNetV1* memperkenalkan pendekatan bernama *depthwise separable convolution* yang secara signifikan mengurangi jumlah parameter dan kebutuhan komputasi tanpa mengorbankan akurasi secara berarti. Pendekatan ini membagi proses konvolusi menjadi dua tahap: *depthwise convolution*, yang mengaplikasikan satu filter per channel input secara terpisah, dan *pointwise convolution* (konvolusi 1x1), yang bertugas menggabungkan hasil dari *depthwise convolution* tersebut. Hasil dari kombinasi dua proses ini tetap menghasilkan fitur yang informatif, tetapi dengan kompleksitas komputasi yang jauh lebih rendah dibanding konvolusi konvensional (Howard et al., 2017).

Efisiensi yang dihasilkan oleh *MobileNetV1* tidak hanya terbatas pada kecepatan pelatihan atau inferensi, tetapi juga berdampak langsung pada ukuran *file*

model dan daya yang dibutuhkan. Hal ini menjadikannya sangat cocok untuk aplikasi *real-time* dan berbasis *mobile*, seperti sistem pengenalan gestur angka dalam bahasa isyarat yang dikembangkan dalam penelitian ini. Pada praktiknya, *MobileNetV1* digunakan sebagai *feature extractor* utama, dan hanya lapisan-lapisan akhir yang dilatih kembali (*fine-tuned*) agar dapat beradaptasi dengan *dataset* spesifik yang digunakan dalam klasifikasi gerakan angka. Strategi ini memungkinkan pemanfaatan kemampuan *representational learning* yang sudah dipelajari dari *dataset* besar seperti *ImageNet*, sekaligus menghemat waktu dan sumber daya saat melatih model pada *dataset* baru dengan ukuran terbatas.

*MobileNetV1* juga telah terbukti memiliki performa klasifikasi yang kompetitif jika dibandingkan dengan arsitektur CNN lain seperti *VGGNet*, *GoogLeNet*, atau *ResNet*. Dalam studi awalnya, Howard et al. menunjukkan bahwa *MobileNetV1* dapat mencapai akurasi klasifikasi yang mendekati *VGG16* dengan hanya menggunakan sekitar 1/30 jumlah parameter dan 1/27 beban komputasi. Performa ini menjadikan *MobileNetV1* sebagai solusi yang ideal dalam pengembangan sistem klasifikasi citra berbasis bahasa isyarat yang menekankan efisiensi dan kemampuan berjalan di perangkat dengan sumber daya terbatas. Dengan mengintegrasikan *MobileNetV1* dalam arsitektur klasifikasi citra, penelitian ini mampu membangun sistem yang efisien, akurat, dan praktis untuk digunakan dalam skenario dunia nyata, khususnya dalam upaya penerjemahan gerakan angka dari bahasa isyarat BISINDO secara otomatis.

### 3.3. Keras dan TensorFlow

*TensorFlow* adalah sebuah *framework* sumber terbuka yang dikembangkan oleh *Google Brain* untuk mendukung pembangunan dan pelatihan model *machine learning* dan *deep learning* secara efisien. *TensorFlow* menyediakan struktur berupa *graph computation* yang memungkinkan pengguna untuk menjalankan operasi numerik dalam bentuk *node* dan *edge*, serta dapat dioptimalkan melalui eksekusi paralel baik di *CPU* maupun *GPU* (Abadi et al., 2016). Kemampuan ini menjadikan *TensorFlow* sebagai salah satu alat utama dalam pengembangan sistem berbasis kecerdasan buatan, termasuk di bidang *computer vision* dan pengolahan bahasa alami.

Di atas *TensorFlow* terdapat *Keras*, yaitu *API* tingkat tinggi yang menyederhanakan proses pengembangan model *deep learning* dengan sintaks yang lebih intuitif dan *user-friendly*. *Keras* memungkinkan pengguna untuk dengan cepat membangun arsitektur *neural network* melalui pendekatan berbasis objek, baik dengan menggunakan model *sekuensial* maupun model fungsional. Dalam penelitian ini, *Keras* digunakan sebagai antarmuka utama dalam membangun model *Convolutional Neural Network* berbasis arsitektur *MobileNet*, menyusun alur *layer*, serta melakukan proses pelatihan, validasi, dan evaluasi model secara langsung. Selain itu, *Keras* juga berperan penting dalam proses augmentasi data melalui *ImageDataGenerator*, serta dalam pemanggilan model yang telah disimpan untuk keperluan evaluasi *real-time*.

Integrasi antara *TensorFlow* dan *Keras* memberikan fleksibilitas yang sangat besar dalam proses pengembangan, mulai dari pengolahan data, pembentukan model, hingga evaluasi performa dan penyimpanan model. Keduanya saling melengkapi dalam menyediakan solusi *end-to-end* untuk kebutuhan sistem pembelajaran mesin yang kompleks namun tetap efisien. Dengan memanfaatkan kekuatan *TensorFlow* sebagai mesin komputasi dan kemudahan pemrograman dari *Keras*, penelitian ini berhasil membangun sebuah sistem klasifikasi gerakan angka dalam bahasa isyarat yang dapat dijalankan secara *real-time*.

### 3.3.1. NumPy

*NumPy* (*Numerical Python*) merupakan pustaka fundamental dalam bahasa pemrograman *Python* yang digunakan untuk komputasi ilmiah, khususnya operasi terhadap *array* multidimensi dan matriks. Pustaka ini menyediakan fungsi-fungsi matematika tingkat tinggi seperti operasi linear *algebra*, transformasi *Fourier*, dan operasi statistik yang sangat efisien. *NumPy* juga mampu mempercepat perhitungan numerik karena didukung oleh implementasi dalam bahasa C di balik layarnya, sehingga sangat cocok untuk diterapkan dalam pemrosesan data skala besar (Oliphant, 2006).

### **3.3.2. Pandas**

*Pandas* adalah pustaka *Python* yang dirancang khusus untuk analisis dan manipulasi data dalam format tabel atau *dataframe*. Pustaka ini sangat bermanfaat dalam konteks penelitian dan ilmu data karena memungkinkan penanganan data dalam bentuk baris dan kolom secara efisien. Fungsi-fungsi seperti *filtering*, *grouping*, *merging*, dan agregasi data dapat dilakukan dengan mudah menggunakan *Pandas* (McKinney, 2010).

Dalam penelitian ini, *Pandas* digunakan untuk mengelola informasi terkait distribusi kelas, hasil klasifikasi, dan penyusunan laporan akhir dalam bentuk tabel. Salah satu penggunaannya adalah untuk membuat *dataframe* yang merepresentasikan jumlah gambar dari setiap kelas, memudahkan pemahaman terhadap distribusi data. Selain itu, *Pandas* juga digunakan untuk menyimpan hasil prediksi dalam format .csv yang dapat dianalisis lebih lanjut atau diolah dengan perangkat lunak lainnya. Kemampuannya dalam membaca dan menulis berbagai format file, seperti CSV dan Excel, membuat *Pandas* menjadi komponen penting dalam *pipeline* analisis dan pelaporan hasil model klasifikasi.

### **3.3.3. OpenCV**

*OpenCV* (*Open Source Computer Vision Library*) adalah pustaka yang digunakan secara luas dalam pengolahan citra dan visi komputer. *OpenCV* menyediakan beragam fungsi yang dapat digunakan untuk membaca gambar, mengubah ukuran gambar (*resizing*), mengatur skala warna, menerapkan filter, deteksi objek, dan berbagai operasi lain yang berkaitan dengan analisis visual (Bradski, 2000).

Dalam penelitian ini, *OpenCV* berperan sebagai alat bantu utama dalam tahap *preprocessing* citra. Fungsi-fungsi seperti pembacaan gambar dari direktori, konversi warna dari BGR ke RGB, dan perubahan ukuran gambar agar sesuai dengan input model dilakukan menggunakan pustaka ini. Kemampuan *OpenCV* dalam mengolah gambar dengan cepat dan fleksibel membuatnya sangat cocok untuk diterapkan dalam sistem yang memerlukan pemrosesan banyak gambar, seperti pada proses augmentasi dan eksplorasi *dataset*. Penggunaan *OpenCV* juga mempermudah integrasi *pipeline* dari proses awal hingga model siap dilatih.

### **3.3.4. Matplotlib**

*Matplotlib* adalah pustaka visualisasi data dua dimensi yang paling populer di lingkungan *Python*. Dengan *Matplotlib*, pengguna dapat membuat beragam jenis grafik, mulai dari grafik garis, diagram batang, hingga *scatter plot* dan histogram. Pustaka ini memberikan kontrol penuh atas elemen grafik seperti warna, label, ukuran sumbu, dan penempatan legenda, yang membuat visualisasi data menjadi informatif dan profesional (Hunter, 2007).

Dalam konteks penelitian ini, *Matplotlib* digunakan untuk menampilkan hasil pelatihan model dalam bentuk grafik *loss* dan akurasi terhadap jumlah *epoch*. Grafik-grafik tersebut memudahkan pemantauan performa model, baik dari segi kestabilan pelatihan maupun indikasi terjadinya *overfitting*. Selain itu, *Matplotlib* juga digunakan untuk menampilkan gambar prediksi saat evaluasi dilakukan, memperlihatkan label prediksi serta tingkat kepercayaan (*confidence score*) dari model terhadap gambar yang diuji. Kemampuannya untuk menyesuaikan elemen visual sangat membantu dalam menyajikan hasil yang mudah dibaca dan diinterpretasikan.

### **3.3.5. Seaborn**

*Seaborn* adalah *library* visualisasi statistik berbasis *Matplotlib* yang dirancang untuk menyederhanakan pembuatan grafik yang menarik dan informatif. *Seaborn* menyediakan antarmuka tingkat tinggi untuk membuat grafik seperti *heatmap*, *boxplot*, dan *violin plot* yang sangat berguna dalam eksplorasi data dan analisis distribusi (Waskom, 2021). Dalam konteks penelitian ini, *Seaborn* digunakan untuk menyajikan *confusion matrix* dan visualisasi distribusi data hasil klasifikasi.

### **3.3.6. SciPy**

*SciPy* adalah pustaka *Python* yang dirancang untuk komputasi ilmiah dan teknis. Di antara banyak *submodulnya*, *SciPy* menyediakan fungsi-fungsi untuk optimisasi, integrasi, interpolasi, aljabar linier, dan lainnya. Salah satu fungsi yang digunakan dalam penelitian ini adalah *make\_interp\_spline* dari modul interpolasi, yang berfungsi untuk melakukan *smoothing* atau penghalusan data agar kurva yang ditampilkan menjadi lebih halus dan mudah dipahami (Virtanen et al., 2020).

Dalam konteks penelitian ini, *SciPy* digunakan untuk memperhalus grafik akurasi dan *loss* yang dihasilkan selama proses pelatihan model. Alih-alih menampilkan grafik dengan fluktuasi tajam yang mungkin menyulitkan interpretasi, interpolasi *spline* digunakan untuk menghasilkan kurva dengan bentuk yang lebih natural. Ini sangat membantu dalam menilai tren performa model dari waktu ke waktu dan mendukung pengambilan keputusan terhadap parameter pelatihan, seperti jumlah *epoch* dan penerapan *early stopping*.

### 3.3.7. Scikit-learn

*Scikit-learn* adalah pustaka pembelajaran mesin yang menyediakan berbagai algoritma dan fungsi evaluasi untuk *supervised* dan *unsupervised learning*. Dalam penelitian ini, *Scikit-learn* dimanfaatkan untuk mengevaluasi performa model klasifikasi melalui metrik-metrik seperti *precision*, *recall*, *F1-score*, serta *confusion matrix*. Fungsi *classification\_report* dari pustaka ini menghasilkan ringkasan statistik dari performa model pada masing-masing kelas, sementara *confusion\_matrix* digunakan untuk menyusun matriks klasifikasi berdasarkan hasil prediksi (Pedregosa et al., 2011).

Dengan *Scikit-learn* memungkinkan proses evaluasi model dilakukan dengan mudah dan komprehensif. Setiap metrik yang dihitung memberikan informasi tambahan tentang kualitas prediksi model, baik dari segi akurasi keseluruhan, ketepatan (*precision*), sensitivitas (*recall*), maupun keseimbangan antar keduanya (*F1-score*). Dengan bantuan pustaka ini, performa model dapat dianalisis secara lebih menyeluruh dan akurat.

### 3.3.8. Pickle

*Pickle* adalah pustaka standar dalam *Python* yang digunakan untuk melakukan *serialisasi* dan *deserialisasi* objek. Dalam praktiknya, *Pickle* digunakan untuk menyimpan objek *Python* ke dalam *file* sehingga dapat dimuat kembali di kemudian hari tanpa harus merekonstruksi ulang dari awal. Dalam penelitian ini, *Pickle* digunakan untuk menyimpan objek model dan struktur *class index* yang telah dilatih,

sehingga proses prediksi *real-time* dapat dilakukan tanpa harus melakukan pelatihan ulang.

Dengan menyimpan model dalam format .pkl dan .keras, penelitian ini menunjukkan pendekatan praktis dalam penggunaan kembali model pada sesi berbeda, termasuk ketika mengimplementasikan aplikasi klasifikasi dalam bentuk antarmuka pengguna atau sistem *real-time*. *Pickle* mempermudah manajemen model, memungkinkan efisiensi waktu dan sumber daya selama pengujian dan *deployment*.

### 3.4. Pengujian Model Klasifikasi

Pengujian terhadap model dilakukan dengan memanfaatkan 20% data yang telah disisihkan sebagai data uji. Data ini belum pernah digunakan dalam proses pelatihan, sehingga dapat memberikan gambaran objektif mengenai kemampuan generalisasi model.

Pengujian dilakukan dengan dua pendekatan, yakni menggunakan fungsi *model.evaluate()* pada dataset validasi untuk mendapatkan metrik performa, serta menggunakan fungsi *model.predict()* pada dataset uji untuk memperoleh hasil prediksi yang kemudian dibandingkan dengan label sebenarnya. Hasil prediksi tersebut kemudian dianalisis melalui *classification report* dan *confusion matrix*.

#### 3.4.1. Akurasi (Accuracy)

Akurasi adalah metrik dasar yang menunjukkan seberapa sering prediksi model sesuai dengan label yang sebenarnya. Secara matematis, akurasi dihitung sebagai rasio jumlah prediksi yang benar terhadap total jumlah prediksi. Dalam *classification report*, akurasi mencerminkan performa keseluruhan dari model pada seluruh kelas.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

Keterangan:

TP = *True Positive*

TN = *True Negative*

FP = *False Positive*

$FN = False\ Negative$

### 3.4.2. Presisi (Precision)

Presisi menggambarkan seberapa akurat model dalam mengklasifikasikan suatu kelas. Nilai presisi dihitung sebagai rasio antara jumlah prediksi benar untuk suatu kelas dengan total prediksi yang mengarah ke kelas tersebut. Presisi yang tinggi menandakan bahwa model jarang salah mengklasifikasikan data dari kelas lain sebagai kelas yang dimaksud.

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

Keterangan:

$TP = True\ Positive$

$TN = True\ Negative$

$FP = False\ Positive$

$FN = False\ Negative$

### 3.4.3. Recall

*Recall* mengukur seberapa banyak data dari suatu kelas yang berhasil dikenali dengan benar oleh model. Nilai *recall* dihitung sebagai rasio antara jumlah prediksi benar terhadap total jumlah data aktual dari kelas tersebut. *Recall* yang tinggi menandakan bahwa model berhasil menangkap sebagian besar contoh dari kelas tersebut.

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

Keterangan:

$TP = True\ Positive$

$TN = True\ Negative$

$FP = False\ Positive$

$FN = False\ Negative$

#### 3.4.4. F1-Score

*F1-score* merupakan metrik gabungan yang merepresentasikan rata-rata harmonis dari presisi dan *recall*. Metrik ini berguna terutama ketika terdapat ketidakseimbangan data antar kelas. *F1-score* memberikan keseimbangan antara kemampuan model dalam mengenali kelas secara benar dan ketepatannya dalam melakukan klasifikasi.

$$F1 - Score = \frac{2 \times P \times R}{P + R} \quad (4)$$

Keterangan:

P = *Precision*

R = *Recall*

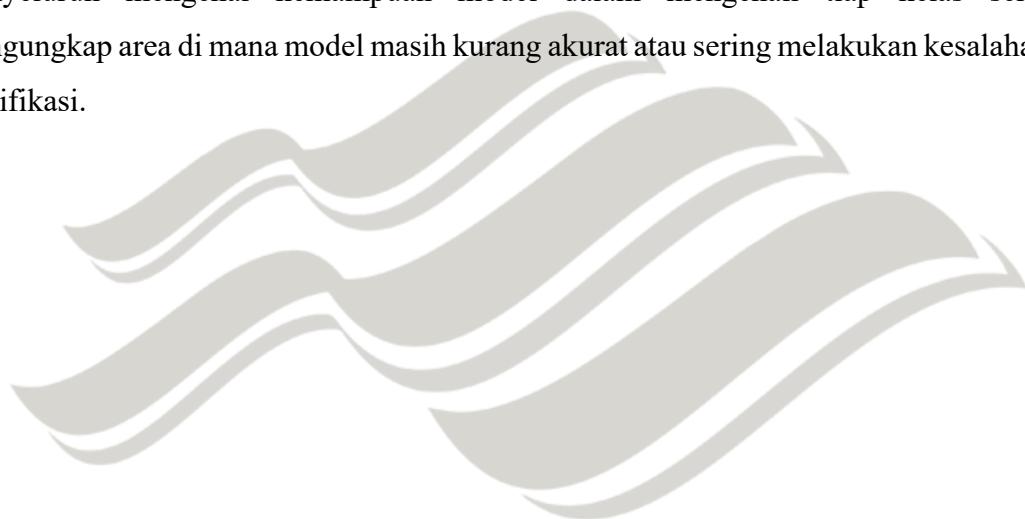
#### 3.4.5. Confusion Matrix

*Confusion matrix* adalah representasi visual dari performa klasifikasi model dalam bentuk tabel. Setiap baris dalam tabel menunjukkan jumlah prediksi aktual untuk suatu kelas, sementara setiap kolom menunjukkan jumlah prediksi yang dilakukan model. Dengan *confusion matrix*, dapat diketahui seberapa sering model keliru mengklasifikasikan suatu kelas menjadi kelas lain, serta kelas mana yang paling sering salah atau benar diklasifikasikan.

		Actual Values	
		1 (Positive)	0 (Negative)
Predicted Values	1 (Positive)	TP (True Positive)	FP (False Positive) <small>Type I Error</small>
	0 (Negative)	FN (False Negative) <small>Type II Error</small>	TN (True Negative)

Gambar 3. 3 Contoh *Confusion Matrix*

*Confusion matrix* seperti yang ditampilkan pada Gambar 3. berfungsi sebagai alat evaluasi yang memberikan gambaran lebih mendalam terhadap kinerja model, terutama ketika data yang digunakan bersifat tidak seimbang, yakni jumlah sampel tiap kelas tidak merata. Melalui *confusion matrix*, berbagai metrik evaluasi seperti akurasi, presisi, *recall* (sensitivitas), dan *F1-score* dapat dihitung untuk menilai seberapa baik model dalam melakukan klasifikasi. Matriks ini memberikan informasi yang lebih menyeluruh mengenai kemampuan model dalam mengenali tiap kelas serta mengungkap area di mana model masih kurang akurat atau sering melakukan kesalahan klasifikasi.



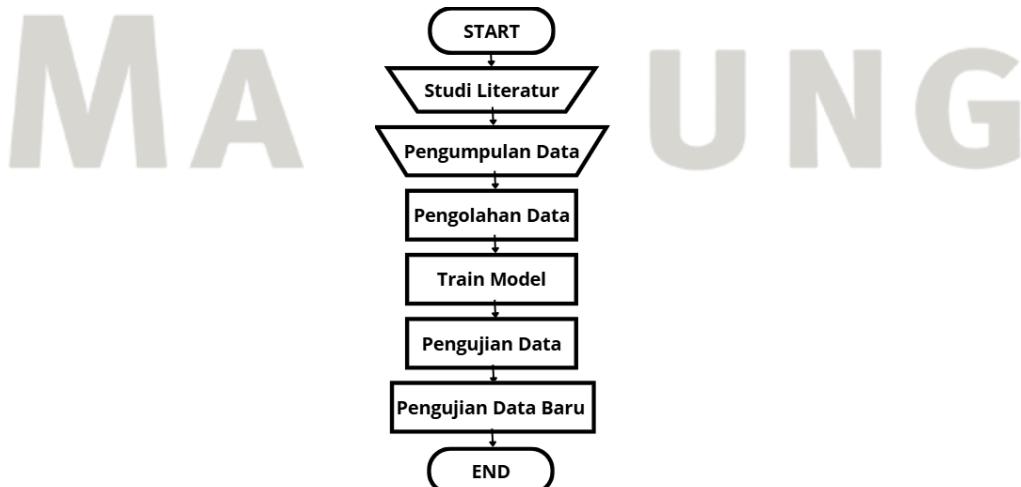
UNIVERSITAS  
**MA CHUNG**

## Bab IV

### Deskripsi Data dan Hasil Praktik Kerja Lapangan

#### 4.1. Alur Pengerjaan Penelitian

Alur pengerjaan penelitian ini dimulai dengan tahap Studi Literatur, di mana peneliti mengumpulkan informasi dan referensi yang relevan untuk memahami konsep-konsep dasar yang mendasari penelitian, seperti Bahasa Isyarat Indonesia (BISINDO), *Convolutional Neural Network* (CNN), dan arsitektur MobileNetV1. Setelah itu, dilakukan Pengumpulan Data, yakni proses mendapatkan data citra gestur tangan angka 0 hingga 10 dari partisipan. Tahap selanjutnya adalah Pengolahan Data, yang mencakup proses preprocessing seperti pengubahan ukuran gambar, normalisasi piksel, serta augmentasi citra untuk memperkaya variasi data. Setelah data siap, dilakukan Pelatihan Model (*Train Model*) menggunakan CNN berbasis MobileNetV1, di mana model dilatih untuk mengenali pola gestur dari data yang telah diolah. Setelah model selesai dilatih, dilakukan Pengujian Data untuk mengevaluasi kinerja model, baik menggunakan data uji yang diambil dari dataset maupun data baru dari subjek yang belum pernah dilibatkan sebelumnya dengan dataset yang memiliki karakteristik background yang beragam dan juga data baru dari ke-2 dimana terdapat 3 subjek yang menggunakan data baru dengan background berwarna hitam guna meningkatkan akurasi. Terakhir, proses ditutup pada tahap END, setelah seluruh eksperimen dan evaluasi selesai dilakukan seperti yang dapat dilihat pada Gambar 4.1.



Gambar 4. 1 Skema Alur Pengerjaan Penelitian

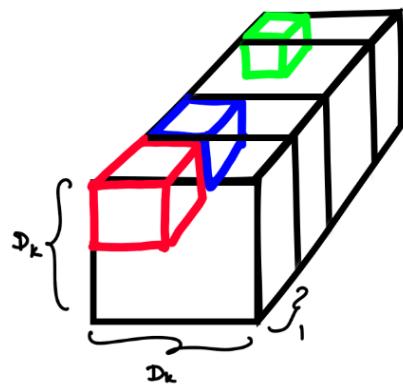
## 4.2. MobileNetV1

*MobileNetV1* adalah arsitektur *Convolutional Neural Network* (CNN) yang dikembangkan oleh *Google* dengan tujuan utama menyediakan model yang efisien, ringan, dan cepat, terutama untuk digunakan pada perangkat dengan keterbatasan sumber daya komputasi, seperti *smartphone* dan *embedded devices* (Howard et al., 2017).

Keunggulan utama dari *MobileNetV1* adalah penggunaan *Depthwise Separable Convolution*, yaitu teknik yang memisahkan operasi konvolusi standar menjadi dua tahap:

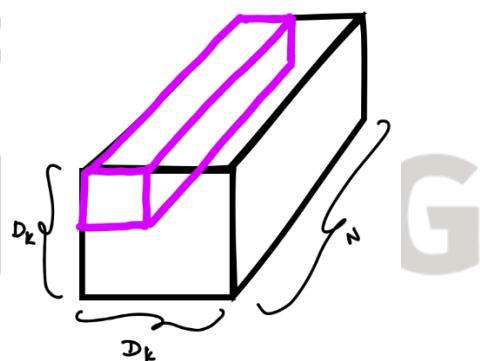
1. *Depthwise Convolution*, proses ini menerapkan satu filter pada masing-masing *channel* dari input, bukan seluruh *channel* sekaligus. Dengan ini, operasi konvolusi dilakukan secara terpisah pada setiap *channel* citra.
2. *Pointwise Convolution (1x1 convolution)*, setelah *depthwise* selesai, hasilnya digabung dengan menggunakan konvolusi 1x1 di semua *channel*. Ini berfungsi untuk menggabungkan informasi antar *channel* yang sebelumnya diproses terpisah.

Depthwise Convolutions



Gambar 4. 3 Depthwise Convolution

1x1 Convolutions



Gambar 4. 2 1x1 Convolution

Kombinasi seperti pada Gambar 4.2 dan juga 4.3 terlihat dapat mengurangi jumlah parameter dan operasi komputasi secara signifikan dibandingkan konvolusi biasa. Secara umum, *depthwise separable convolution* mengurangi kompleksitas

model hingga 8–9 kali lipat dikarenakan penggabungan dari *Depthwise Convolution* dan juga *Pointwise Convolution (1x1 Convolution)*.

*MobileNetV1* memiliki model yang telah dilatih sebelumnya (*pre-trained* model) pada *dataset ImageNet*, sebuah *dataset* besar berisi jutaan gambar dari ribuan kategori objek. Ini memungkinkan untuk melakukan *transfer learning*, yaitu mengambil model yang telah belajar fitur umum dari gambar (misal bentuk, warna, tekstur) dan menyesuaikannya untuk klasifikasi tugas khusus dalam kasus ini adalah klasifikasi gestur angka Bahasa Isyarat BISINDO.

```
1 mobilenet = MobileNet(  
2     input_shape=(224, 224, 3),  
3     include_top=False,  
4     weights='imagenet')
```

Gambar 4. 4 Proses Mengimpor model *MobileNet Pre-Trained*  
sebagai *Feature Extractor*

Pada Gambar 4.4, untuk dapat menambahkan layer baru sesuai kebutuhan klasifikasi sebanyak 11 kelas angka, membutuhkan *include\_top=False* yang berarti bagian *fully connected* layer bawaan *ImageNet* dihilangkan. *weights='imagenet'* menunjukkan bahwa bobot awal model diambil dari pelatihan pada *dataset ImageNet*. Untuk efisiensi dan stabilitas pelatihan, layer dari model tidak dilatih dari awal, melainkan hanya 30-layer terakhir yang diaktifkan untuk di-*train*:

```
1 for layer in mobilenet.layers[:-30]:  
2     layer.trainable = False  
3 for layer in mobilenet.layers[-30:]:  
4     layer.trainable = True
```

Gambar 4. 5 Proses pembekuan sebagian besar layer

Pada Gambar 4.5, *Layer* awal (awal jaringan) biasanya telah belajar fitur umum seperti deteksi tepi, tekstur, dan pola dasar. *Layer* akhir bersifat spesifik terhadap tugas tertentu dan perlu disesuaikan agar relevan dengan tugas baru, yaitu klasifikasi gestur BISINDO. Oleh karena itu, lapisan-lapisan tersebut dibuka dan dilatih ulang agar mampu mengenali pola dari data yang digunakan dalam penelitian ini. Hal ini merupakan strategi *fine-tuning* yang menjaga bobot stabil dari *pretrained* model, sambil tetap menyesuaikan fitur ke domain baru.

#### 4.2.2. Implementasi Arsitektur Model

Setelah bagian *feature extraction* selesai, Arsitektur klasifikasi akan disusun kembali menggunakan *Sequential* seperti pada Gambar 4.6:

```
1 model = Sequential([
2     mobilenet,
3     GlobalAveragePooling2D(),
4     Dropout(0.5),
5     Dense(256, activation='relu'),
6     Dropout(0.3),
7     Dense(num_classes, activation='softmax',
           name='classification')])
```

Gambar 4.6 Proses *Transfer Learning* dan *Fine-Tuning*

*Mobilenet* digunakan sebagai *feature extractor* yang telah dipersiapkan dengan bobot *pretrained* dan *fine-tuned*. *GlobalAveragePooling2D()* digunakan untuk Mengubah *output* tensor 2D dari *MobileNet* menjadi vektor 1D dengan cara merata-ratakan nilai dari setiap *feature map* (lebih ringan daripada *Flatten* dan mengurangi *overfitting*). *Dropout(0.5)* berfungsi untuk Menghilangkan 50% neuron secara acak saat pelatihan untuk mencegah *overfitting*. *Dense(256, activation='relu')* menyatakan *Layer fully connected* untuk pembelajaran non-linear terhadap fitur yang telah diekstraksi. *Dropout(0.3)* merupakan tambahan sebagai regularisasi. *Dense(num\_classes, activation='softmax')* merupakan *Layer* akhir untuk klasifikasi

*multi-kelas* (angka 0 sampai 10). Fungsi *softmax* digunakan karena ini adalah klasifikasi kategori eksklusif (satu kelas per input).

Lalu model dikompilasi dengan:

```
1 model.compile(  
2     optimizer='adam',  
3     loss='categorical_crossentropy',  
4     metrics=['accuracy'])  
5 }
```

Gambar 4. 7 Proses Kompilasi

Pada Gambar 4.7 ditunjukkan proses kompilasi model yang dilakukan sebelum tahap pelatihan, di mana model dikonfigurasi menggunakan fungsi *compile()* dengan *optimizer adam*, fungsi *loss categorical\_crossentropy*, dan metrik evaluasi *accuracy*. *Optimizer adam* dipilih karena mampu mempercepat proses pembelajaran melalui penyesuaian laju secara adaptif, sedangkan *categorical\_crossentropy* digunakan untuk menghitung selisih antara hasil prediksi dan label sebenarnya pada kasus klasifikasi multikelas dengan format *one-hot encoding*. Metrik *accuracy* digunakan untuk memantau persentase prediksi model yang benar terhadap data uji, sehingga model dapat belajar secara optimal selama proses pelatihan.

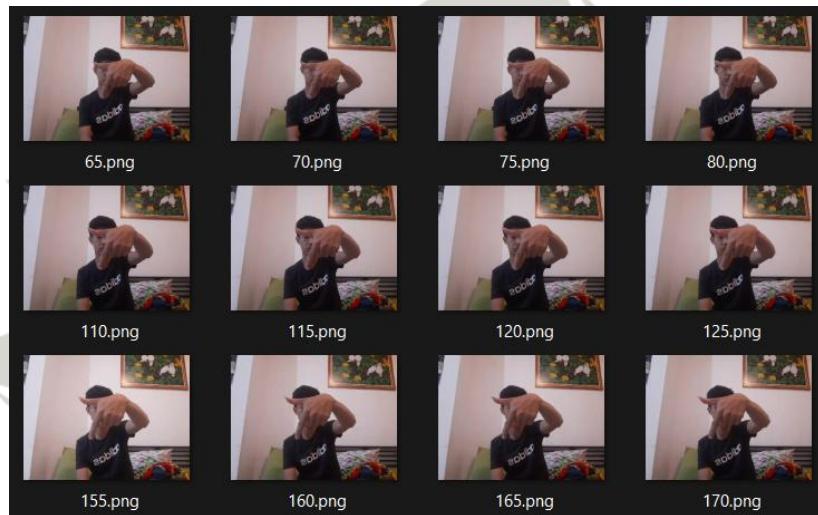
### 4.3. Dataset

*Dataset* merupakan kumpulan data yang digunakan dalam proses pelatihan dan pengujian model. Dalam konteks *machine learning*, *dataset* terdiri dari input (fitur) dan output (label) yang digunakan untuk melatih model agar mampu melakukan prediksi.

Pada penelitian ini, *dataset* yang digunakan merupakan *data gesture* angka dari Bahasa Isyarat Indonesia (BISINDO). *Dataset* ini berasal dari penelitian terdahulu oleh Nico Alexander Handoko, mahasiswa Universitas Ma Chung. Dataset mencakup gesture angka dari 0 hingga 10, masing-masing mewakili satu kelas.

#### 4.3.1. Pengambilan Data

Proses pengambilan data melibatkan 5 orang responden. Masing-masing responden menyumbangkan 200 gambar per kelas, sehingga untuk setiap kelas (misal kelas 0), terdapat total 1000 gambar (200 gambar  $\times$  5 responden). Dengan 11 kelas, total keseluruhan *dataset* berjumlah sekitar 11.000 gambar. Gambar dikumpulkan dengan latar belakang beragam dan kondisi pencahayaan yang bervariasi agar model lebih bagus untuk digunakan pada saat *training* data, *validation*, dan juga *test* data.



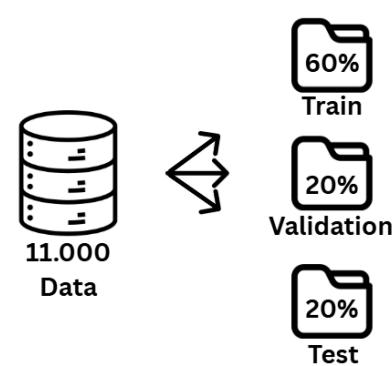
Gambar 4. 8 Contoh gestur angka 7

(Sumber : Alexander, 2023)

Setelah itu, terdapat pengambilan data baru dimana data ini dipakai untuk menguji kemampuan generalisasi model. Terdapat 6 Responden yang dimana masing-masing responden menyumbangkan 10 gambar per kelas, sehingga untuk setiap kelas (misal kelas 0), terdapat total 60 gambar (110 gambar  $\times$  6 responden). Dengan 11 kelas, total keseluruhan *dataset* berjumlah sekitar 660 gambar. Gambar dikumpulkan dengan latar belakang beragam dan kondisi pencahayaan yang bervariasi demikian, dan setelah itu terdapat pengambilan data dari 3 subjek sebelumnya yang memiliki rata-rata terendah dengan latar belakang hitam dan kondisi pencahayaan yang stabil untuk menggantikan data dari 3 subjek sebelumnya yang memiliki rata-rata terendah.

#### 4.3.2. Data Split

Setelah data berhasil dikumpulkan, tahap selanjutnya adalah membagi *dataset* ke dalam tiga kelompok utama, yaitu data pelatihan, data validasi, dan data pengujian. Proporsi pembagian data dilakukan secara acak dengan komposisi 60% untuk pelatihan, 20% untuk validasi, dan 20% untuk pengujian. Pembagian ini dilakukan menggunakan fungsi *train\_test\_split* dari *library Scikit-learn*. Seluruh gambar kemudian disusun ulang ke dalam folder terpisah yaitu *dataset\_split/train*, *dataset\_split/val*, dan *dataset\_split/test*. Strategi ini memastikan bahwa model memiliki cukup data untuk belajar, sekaligus menyediakan data independen untuk mengukur performa dan kemampuan model. Lalu terdapat 660 data baru dari 6 subjek baru yang tidak berkaitan dengan data train, validation, dan test diatas untuk menguji kemampuan generalisasi model.



Gambar 4. 9 Skema Pembagian Data

#### 4.3.3. Preprocessing Data

*Preprocessing* atau prapemrosesan data merupakan tahap penting untuk menyiapkan citra sebelum diberikan ke dalam model. Berdasarkan implementasi kode, *preprocessing* dilakukan dengan beberapa langkah utama. Gambar-gambar awal diubah ukurannya menjadi 224x224 piksel untuk disesuaikan dengan input standar dari arsitektur *MobileNet*. Nilai piksel pada gambar yang semula berada dalam skala 0–255 kemudian dinormalisasi menjadi skala 0–1 dengan membagi seluruh piksel dengan 255.

#### 4.3.4. Augmentation Data

Augmentasi data pada tahap pelatihan dilakukan untuk memperbanyak variasi data secara buatan guna meningkatkan kemampuan generalisasi model dan mencegah

*overfitting*. Teknik augmentasi ini diterapkan melalui *ImageDataGenerator* dengan sejumlah transformasi yang disesuaikan. Pertama, gambar diputar secara acak dalam rentang hingga 30 derajat (*rotation\_range*=30) untuk membuat model lebih toleran terhadap variasi sudut tangan. Selain itu, gambar juga digeser secara horizontal dan vertikal hingga 20% dari ukuran gambar (*width\_shift\_range*=0.2 dan *height\_shift\_range*=0.2) agar model mampu mengenali gestur yang tidak selalu berada di tengah. Variasi ukuran juga dilakukan melalui *zoom in* atau *zoom out* sebesar ±20% (*zoom\_range*=0.2), yang membantu model mengenali objek tangan pada skala yang berbeda-beda. Selanjutnya, tingkat pencahayaan pada gambar diubah dalam rentang 70% hingga 130% (*brightness\_range*=[0.7, 1.3]) untuk mensimulasikan kondisi pencahayaan yang berbeda, seperti ruangan terang atau redup. Terakhir, dilakukan pemanjangan gambar secara horizontal (*horizontal\_flip*=True) agar model bisa mengenali gestur tangan dari sisi kiri maupun kanan. Dengan kombinasi augmentasi ini, model diharapkan dapat belajar dari beragam variasi citra gestur tanpa harus secara eksplisit melihat semua kondisi tersebut dalam data aslinya.

#### 4.4. Pembentukan Model Klasifikasi

Pada tahap ini, dilakukan pembentukan arsitektur model klasifikasi citra berbasis *Convolutional Neural Network* (CNN) dengan pendekatan *Transfer Learning* menggunakan arsitektur *MobileNet* sebagai *feature extractor* dengan 2 macam *dataset* yaitu dari *Pre-trained Dataset (ImageNet)* dan juga *Custom Dataset* yaitu 11.000 data yang digunakan untuk data *Train*, data *Validation*, dan juga data *Test*.

*MobileNet* dipilih karena merupakan model ringan dan efisien yang telah dilatih sebelumnya pada dataset berskala besar (*ImageNet*), sehingga mampu mengenali fitur visual dasar secara efektif. Model ini dimuat dengan parameter *include\_top*=False untuk menghilangkan bagian klasifikasi aslinya dan menggantinya dengan *layer* baru yang disesuaikan dengan jumlah kelas pada *dataset* gestur angka seperti yang dapat dilihat pada Gambar 4.4.

Untuk memaksimalkan kemampuan generalisasi model sekaligus mengurangi beban komputasi, dilakukan proses *fine-tuning* secara selektif dengan membekukan seluruh *layer* awal dan hanya melatih ulang 30 *layer* terakhir dari *MobileNet*

(*trainable=True*). Setelah *feature extractor* disiapkan, model kemudian dibentuk menggunakan arsitektur *Sequential* yang terdiri atas beberapa lapisan tambahan, yaitu *GlobalAveragePooling2D* untuk meratakan output fitur dari *MobileNet* menjadi vektor satu dimensi, diikuti oleh lapisan Dense dengan 256 unit dan fungsi aktivasi *ReLU* sebagai hidden layer, serta *Dropout* sebesar 0.5 dan 0.3 untuk mengurangi risiko *overfitting*. Lapisan output menggunakan fungsi aktivasi softmax untuk mengklasifikasikan ke dalam sejumlah kelas (*num\_classes*) sesuai dengan banyaknya folder pada dataset pelatihan seperti yang dapat dilihat pada Gambar 4.6.

Model kemudian dikompilasi dengan menggunakan *optimizer Adam* yang adaptif dan efisien, fungsi *loss categorical\_crossentropy* yang sesuai untuk klasifikasi multikelas dengan label dalam format *one-hot encoding*, serta metrik evaluasi *accuracy* untuk memantau kinerja selama proses pelatihan seperti yang dapat dilihat pada Gambar 4.7.

Pada Gambar 4.10, *Dataset* pelatihan (*train\_ds*), validasi (*val\_ds*), dan pengujian (*test\_ds*) dihasilkan dari folder dataset yang telah dibagi sebelumnya menggunakan *ImageDataGenerator* dengan *preprocessing rescale=1./255* untuk normalisasi piksel. Pada tahap pelatihan, juga digunakan augmentasi citra seperti rotasi, pergeseran, *zoom*, perubahan kecerahan, dan *flipping* horizontal guna meningkatkan keragaman data dan mencegah *overfitting*. Selain itu, ditambahkan mekanisme *EarlyStopping* untuk menghentikan pelatihan lebih awal jika tidak ada peningkatan pada nilai loss validasi selama beberapa *epoch*, sekaligus memastikan bobot terbaik disimpan.

```

1  image_gen = ImageDataGenerator(
2      rescale=1./255,
3      rotation_range=30,
4      width_shift_range=0.2,
5      height_shift_range=0.2,
6      zoom_range=0.2,
7      brightness_range=[0.7, 1.3],
8      horizontal_flip=True
9  )
10 base_dir = 'dataset_split'
11 train_ds = image_gen.flow_from_directory(
12     os.path.join(base_dir, 'train'),
13     target_size=(224, 224),
14     batch_size=16,
15     class_mode='categorical',
16     shuffle=True,
17     seed=42
18 )
19 val_ds = ImageDataGenerator(rescale=1./255).flow_from_directory(
20     os.path.join(base_dir, 'val'),
21     target_size=(224, 224),
22     batch_size=16,
23     class_mode='categorical',
24     shuffle=True,
25     seed=42
26 )
27 early_stop = EarlyStopping(
28     monitor='val_loss',
29     patience=5,
30     restore_best_weights=True
31 )

```

Gambar 4. 10 Proses Data Augmentation

#### 4.5. Hasil Desain Model

Model klasifikasi yang telah dirancang dan dibangun dalam penelitian ini menggunakan arsitektur *MobileNetV1* yang dikombinasikan dengan beberapa layer tambahan. Model tersebut mampu mengenali gestur angka dari Bahasa Isyarat Indonesia (BISINDO) dengan performa yang cukup baik. Proses perancangan model meliputi serangkaian tahapan mulai dari pemrosesan citra sebagai input, penerapan

teknik augmentasi data dan normalisasi, hingga pembentukan arsitektur klasifikasi dan evaluasi model. Evaluasi tidak hanya dilakukan terhadap data uji yang telah dipersiapkan, namun juga dilakukan secara real time terhadap gambar-gambar baru yang belum pernah dikenali sebelumnya oleh model.

#### 4.5.1. Input Model

Input yang digunakan oleh model berupa citra digital berwarna (RGB) dalam format .jpg, .jpeg, atau .png, yang telah diubah ukurannya menjadi 224x224 piksel seperti yang dapat dilihat pada Gambar 4.7. Seluruh citra tersebut dinormalisasi dengan membagi nilai pikselnya menjadi rentang 0 hingga 1. Ukuran dan format input ini disesuaikan dengan spesifikasi arsitektur *MobileNetV1* agar dapat diproses secara optimal oleh jaringan.

#### 4.5.2. Preprocessing

Tahap *preprocessing* mencakup *resizing* gambar ke dimensi 224x224 piksel dan normalisasi nilai piksel dengan teknik *rescaling*. Untuk data pelatihan, dilakukan augmentasi data untuk memperkaya variasi input yang diberikan ke model, seperti rotasi, pergeseran, *zoom*, perubahan kecerahan, serta pembalikan horizontal. Seluruh proses ini dilakukan menggunakan *ImageDataGenerator*, yang secara otomatis menghasilkan *batch* data dalam bentuk *tensor* untuk dilatih oleh model seperti yang dapat dilihat pada Gambar 4.9.

#### 4.5.3. Proses Pembuatan Model Klasifikasi

Setelah melalui tahap *preprocessing* berupa normalisasi citra dan pembagian *dataset* menjadi data latih, validasi, dan uji, langkah selanjutnya adalah membangun model klasifikasi citra menggunakan pendekatan *Transfer Learning* dengan arsitektur dasar *MobileNetV1*. *Transfer learning* digunakan karena memungkinkan pemanfaatan bobot hasil pelatihan sebelumnya pada *dataset* besar (seperti *ImageNet*) untuk mempercepat proses pelatihan dan meningkatkan akurasi, khususnya ketika jumlah data relatif terbatas. Model dibangun menggunakan *API Sequential* dari *Keras*, yang menyusun *layer* secara linier.

Langkah pertama dalam membangun arsitektur model adalah menentukan jumlah kelas (*num\_classes*) berdasarkan jumlah *folder* yang terdapat dalam direktori data pelatihan. Kemudian, model dasar *MobileNetV1* diimpor dengan parameter *include\_top=False*, artinya bagian klasifikasi asli dari *MobileNet* dihilangkan agar dapat diganti dengan arsitektur klasifikasi baru yang disesuaikan dengan kebutuhan. Selain itu, parameter *weights='imagenet'* menunjukkan bahwa bobot awal model diambil dari pelatihan pada *dataset ImageNet*. Hal ini membuat *MobileNet* berfungsi sebagai *feature extractor* yang mampu mengenali pola visual umum seperti bentuk, warna, dan tekstur.

Untuk menyesuaikan model terhadap dataset gestur tangan yang digunakan, dilakukan proses *fine-tuning* secara parsial, di mana seluruh layer awal *MobileNet* dibekukan (*trainable=False*) kecuali 30 layer terakhir yang tetap dapat dilatih (*trainable=True*). Strategi ini digunakan agar model dapat beradaptasi pada data baru tanpa kehilangan kemampuan generalisasi dari pretraining awal.

Setelah bagian *feature extractor* siap, arsitektur dilanjutkan dengan menambahkan beberapa layer tambahan:

1. *GlobalAveragePooling2D*: berfungsi untuk meratakan hasil *output* dari *MobileNet* menjadi vektor satu dimensi tanpa kehilangan fitur spasial penting.
2. *Dropout (rate = 0.5)*: digunakan untuk regularisasi dengan cara mematikan secara acak 50% neuron saat pelatihan guna mengurangi *overfitting*.
3. *Dense (256 unit, ReLU)*: sebagai lapisan *fully connected* untuk memproses fitur dan menghubungkannya dengan kelas target.
4. *Dropout (rate = 0.3)*: ditambahkan lagi untuk pengurangan *overfitting* sebelum mencapai lapisan akhir.
5. *Dense (num\_classes, Softmax)*: lapisan output untuk klasifikasi multikelas, di mana fungsi aktivasi *softmax* digunakan untuk menghasilkan probabilitas dari setiap kelas.

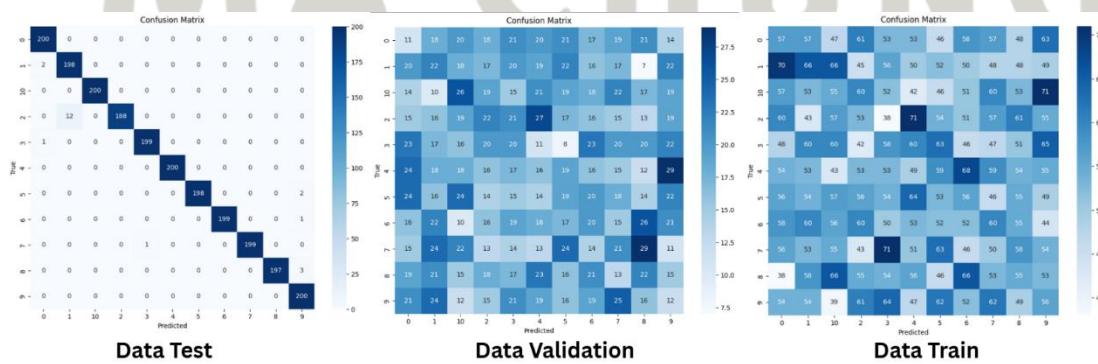
Model yang telah dirancang ini dikompilasi menggunakan fungsi *compile()* dengan pengaturan sebagai berikut:

- *Optimizer*: Adam digunakan karena adaptif dalam menyesuaikan *learning rate* selama proses pelatihan, sehingga efisien dan cocok untuk *deep learning*.
- *Loss Function*: *categorical\_crossentropy*, sesuai untuk klasifikasi multikelas dengan label dalam bentuk *one-hot encoding*.
- *Metrics*: *accuracy*, digunakan sebagai metrik utama untuk mengukur seberapa banyak prediksi model yang benar selama pelatihan dan validasi.

Dengan struktur dan konfigurasi tersebut, model siap untuk menjalani proses pelatihan menggunakan dataset yang telah diaugmentasi dan diproses sebelumnya. Pelatihan dilakukan dengan parameter *epochs*=100 serta ditambahkan callback *EarlyStopping* dengan *monitoring* pada *val\_loss* dan *patience*=5, yang berguna untuk menghentikan pelatihan lebih awal jika tidak ada peningkatan performa validasi, sekaligus menjaga bobot model terbaik yang ditemukan selama proses *training*.

#### 4.5.4. Evaluasi Model

Evaluasi terhadap model dilakukan melalui dua pendekatan. Pertama, evaluasi terhadap data uji yang diperoleh dari hasil *split dataset* dilakukan dengan menghitung *classification report* dan *confusion matrix*. Hasil ini menunjukkan seberapa baik model mengenali masing-masing kelas. Kedua, dilakukan evaluasi menggunakan data baru terhadap gambar-gambar dari folder *TestModel*. Dalam tahap ini, model memprediksi kelas untuk setiap gambar, menghitung tingkat kepercayaan (*confidence*), dan menampilkan hasil prediksi beserta visualisasi gambar yang diuji. Evaluasi ini menunjukkan bahwa model tidak hanya bekerja baik dalam kondisi terkontrol, tetapi juga memiliki potensi untuk diimplementasikan secara praktis dalam aplikasi



Gambar 4. 11 *Confusion Matrix* MobileNet

penerjemahan bahasa isyarat. Gestur-gestur yang digunakan dalam penelitian ini adalah gestur statis yaitu angka 0 sampai dengan 10 (11 kelas).

Pada Gambar 4.11, Berdasarkan tiga confusion matrix yang ditampilkan, performa model pada data test, validasi, dan train memperlihatkan variasi akurasi yang signifikan. Pada confusion matrix Data Test, model menunjukkan performa yang sangat baik dan hampir sempurna. Seluruh label (kelas 0–10) berhasil diklasifikasikan dengan benar, dengan nilai diagonal yang sangat tinggi (sebagian besar bernilai 198–200), dan kesalahan prediksi sangat minimal, hanya beberapa titik seperti kelas 2 ke kelas 1 (12 kesalahan) dan kelas 5 ke kelas 9 (2 kesalahan). Ini mengindikasikan bahwa model sangat akurat terhadap data test yang tidak digunakan dalam pelatihan, menunjukkan bahwa model mampu melakukan generalisasi dengan sangat baik terhadap data terpisah yang memiliki kualitas dan format yang serupa dengan data training.

Sebaliknya, confusion matrix pada Data Validation dan Data Train menunjukkan hasil yang sangat berbeda. Untuk Data Validation, distribusi prediksi sangat menyebar dan cenderung acak. Hal ini mengindikasikan bahwa model kesulitan mengenali gestur dengan benar pada data validasi, yang kemungkinan memiliki kondisi berbeda, seperti latar belakang, pencahayaan, atau posisi tangan yang tidak sesuai dengan pola yang dipelajari model. Untuk Data Train, meskipun distribusinya masih cukup menyebar yang menunjukkan bahwa model dapat menghafal atau mengenali data pelatihan lebih baik, meskipun belum sempurna. Pola ini mengindikasikan potensi overfitting, di mana model bekerja sangat baik pada data yang mirip dengan pelatihan, namun buruk pada data dengan karakteristik berbeda. Oleh karena itu, diperlukan perbaikan dalam diversifikasi dataset dan strategi pelatihan agar model lebih robust terhadap variasi.

Tabel 4. 1 *Classification Report* MobileNet

Data Test			
	<i>precision</i>	<i>recall</i>	<i>f1-score</i>
0	1.00	1.00	1.00

1	0.99	0.98	0.98
2	0.96	0.99	0.97
3	0.99	0.97	0.98
4	1.00	0.99	1.00
5	1.00	1.00	1.00
6	0.99	1.00	0.99
7	0.99	0.99	0.99
8	0.99	0.99	0.99
9	0.99	0.98	0.99
10	1.00	0.99	1.00
<i>accuracy</i>			0.99

#### Data Validation

	<i>precision</i>	<i>recall</i>	<i>f1-score</i>
0	0.05	0.06	0.05
1	0.11	0.11	0.11
2	0.12	0.11	0.11
3	0.10	0.10	0.10
4	0.08	0.08	0.08
5	0.10	0.10	0.10
6	0.10	0.10	0.10
7	0.10	0.10	0.10
8	0.11	0.11	0.11
9	0.06	0.06	0.06
10	0.13	0.13	0.13
<i>accuracy</i>			0.10

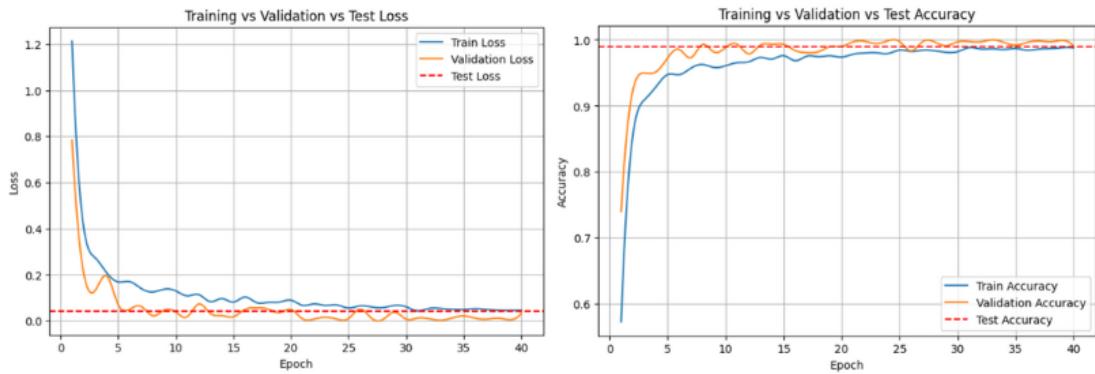
#### Data Train

	<i>precision</i>	<i>recall</i>	<i>f1-score</i>
0	0.09	0.10	0.09
1	0.11	0.11	0.11

2	0.09	0.09	0.09
3	0.10	0.10	0.10
4	0.08	0.08	0.08
5	0.09	0.09	0.09
6	0.09	0.09	0.09
7	0.08	0.08	0.08
8	0.09	0.09	0.09
9	0.09	0.09	0.09
10	0.09	0.09	0.09
<i>accuracy</i>			0.09

Tabel 4.1 menunjukkan hasil *classification report* dari model klasifikasi citra yang menunjukkan performa sangat baik. Nilai-nilai metrik utama seperti *precision*, *recall*, dan *f1-score* mendekati sempurna di hampir semua kelas, dengan sebagian besar berada di angka 0.99 hingga 1.00. Beberapa kelas seperti 0, 10, 4, dan 5 bahkan mencapai skor sempurna (1.00) pada ketiga metrik, yang berarti semua gambar dalam kelas tersebut berhasil dikenali dengan benar oleh model.

Meskipun ada sedikit penurunan performa pada kelas 2 dan 3, dengan *f1-score* masing-masing sebesar 0.97 dan 0.98, namun nilai tersebut masih tergolong sangat tinggi. Secara keseluruhan, model menghasilkan akurasi total sebesar 0.99, menunjukkan bahwa dari 2200 gambar uji, hampir seluruhnya diklasifikasikan dengan benar. Nilai *macro average* dan *weighted average* yang juga mencapai 0.99 mengindikasikan bahwa model mampu mempertahankan performa yang konsisten di seluruh kelas tanpa bias terhadap kelas tertentu.



Gambar 4. 12 Grafik *Accuracy* dan *Loss*

Berdasarkan grafik pada Gambar 4.12, Grafik hasil pelatihan menunjukkan bahwa model klasifikasi citra gestur berbasis MobileNetV1 memiliki performa yang sangat baik. Pada grafik sebelah kiri, terlihat bahwa nilai *training loss* dan *validation loss* menurun tajam hingga stabil mendekati nol, menandakan proses pembelajaran yang efektif tanpa indikasi *overfitting*. Nilai *test loss* yang divisualisasikan sebagai garis datar juga sangat rendah, menunjukkan bahwa model dapat mempertahankan performa tinggi saat diuji dengan data baru. Pada grafik sebelah kanan, baik *training accuracy* maupun *validation accuracy* menunjukkan tren kenaikan yang cepat dan stabil mendekati 100%. Nilai *test accuracy* juga sangat tinggi dan konsisten dengan hasil validasi, menandakan kemampuan generalisasi model yang sangat baik terhadap data yang belum pernah dilihat sebelumnya.

#### 4.5.5. Evaluasi Data Baru

Setelah proses pelatihan model klasifikasi selesai dan model terbaik berhasil diperoleh, tahap selanjutnya adalah mengevaluasi performa model terhadap data baru dari pengguna berbeda, yang tidak pernah digunakan dalam proses pelatihan sebelumnya. Evaluasi ini bertujuan untuk menguji kemampuan generalisasi model dalam mengenali gestur angka pada Bahasa Isyarat Indonesia (BISINDO) ketika diberikan input dari subjek yang berbeda-beda. Pengujian ini dilakukan secara terpisah dalam lingkungan pengujian offline, namun tetap menggunakan data rekaman asli dari pengguna baru untuk melihat sejauh mana model dapat mengenali gestur secara akurat di luar data latih.

Tabel 4. 2 Hasil *accuracy* menggunakan data baru berdasarkan subjek

Subjek	<i>Accuracy</i>
1	57,27%
2	16,36%
3	39,09%
4	59,09%
5	70,91%
6	47,27%
Rata-rata	48,33%

Pengujian dilakukan terhadap enam partisipan yang berbeda yang seluruhnya merupakan subjek baru (tidak terlibat dalam pelatihan). Masing-masing subjek diminta untuk memperagakan seluruh kelas gestur yang digunakan dalam pelatihan, sebanyak lima kali untuk setiap kelas, dengan metode pengumpulan data yang seragam seperti saat pelatihan. Hasil evaluasi pada Tabel 4.2 menunjukkan bahwa performa model bervariasi antar subjek. Secara keseluruhan, rata-rata akurasi dari seluruh partisipan adalah sebesar 48,33%. Hasil ini menunjukkan bahwa meskipun model mampu mengenali sebagian besar gestur dengan cukup baik pada subjek tertentu, masih terdapat tantangan dalam hal konsistensi performa, terutama pada subjek yang memiliki gaya gerakan atau bentuk tangan yang berbeda dari data pelatihan.

Perbedaan akurasi yang cukup besar antar subjek dapat disebabkan oleh beberapa faktor, di antaranya adalah variasi bentuk tangan, kecepatan gerakan, hingga konsistensi dalam memperagakan gestur. Hal ini menunjukkan bahwa model masih memiliki keterbatasan dalam menghadapi keragaman alami dari pengguna baru, terutama jika model dilatih dengan data dari subjek yang terbatas. Selain itu, kemungkinan lain adalah kurangnya fitur penting yang terdeteksi akibat posisi tangan atau sudut pengambilan gambar yang kurang optimal. Evaluasi ini menegaskan pentingnya memperluas cakupan data pelatihan agar mencakup lebih banyak variasi pengguna, sehingga model dapat belajar mengenali gestur dengan lebih adaptif dan akurat terhadap data baru.

Tabel 4. 3 Hasil *accuracy* menggunakan data baru berdasarkan gestur

Gestur	n	Persentase
0	60	72%
1	60	22%
2	60	27%
3	60	25%
4	60	43%
5	60	92%
6	60	60%
7	60	77%
8	60	38%
9	60	12%
10	60	65%
Rata-rata		48%

Seperti yang dapat dilihat pada Tabel 4.3, performa model klasifikasi terhadap masing-masing gestur menunjukkan variasi yang cukup signifikan. Gestur angka 5 mencatat persentase akurasi tertinggi sebesar 92%, diikuti oleh gestur angka 7 sebesar 77%, angka 0 sebesar 72%, dan angka 10 sebesar 65%. Hal ini menunjukkan bahwa gestur-gestur tersebut memiliki pola visual yang lebih konsisten dan mudah dikenali oleh model, baik dari segi bentuk tangan maupun kejelasan kontur. Sebaliknya, gestur angka 9 memperoleh persentase akurasi terendah hanya sebesar 12%, diikuti oleh angka 1 (22%), angka 3 (25%), dan angka 2 (27%). Rendahnya akurasi pada gestur-gestur ini kemungkinan besar disebabkan oleh kemiripan bentuk tangan antar kelas atau kurangnya perbedaan fitur yang signifikan, sehingga membingungkan model dalam melakukan klasifikasi yang akurat.

Dengan rata-rata akurasi keseluruhan sebesar 48%, dapat disimpulkan bahwa masih terdapat ruang untuk peningkatan performa model, terutama dalam membedakan gestur-gestur yang kompleks atau memiliki kemiripan visual. Beberapa faktor yang turut memengaruhi akurasi antara lain adalah kualitas pencahayaan, latar belakang saat

pengambilan data, serta posisi dan konsistensi bentuk tangan. Gestur dengan bentuk yang lebih jelas, kontras yang baik terhadap latar, serta gerakan yang konsisten cenderung memberikan hasil klasifikasi yang lebih akurat. Oleh karena itu, penting untuk memperhatikan aspek teknis dalam akuisisi data serta mempertimbangkan augmentasi atau pelatihan ulang pada gestur-gestur dengan performa rendah guna meningkatkan kinerja model secara keseluruhan.

#### 4.6. Diskusi

Berdasarkan hasil evaluasi akurasi model terhadap masing-masing subjek, dapat disimpulkan bahwa performa model secara umum masih tergolong kurang memuaskan. Hal ini ditunjukkan oleh rendahnya rata-rata akurasi di antara keenam subjek, di mana hanya tiga subjek yang mampu mencapai akurasi di atas 50%, yaitu Subjek 5 dengan 70,91%, Subjek 4 dengan 59,09%, dan Subjek 1 dengan 57,27%. Sementara itu, Subjek 2 hanya mencapai akurasi sebesar 16,36%, dan Subjek 3 serta Subjek 6 masing-masing berada di angka 39,09% dan 47,27%. Perbedaan signifikan ini mengindikasikan adanya faktor eksternal yang memengaruhi kemampuan model dalam mengenali dan mengklasifikasikan gestur angka dengan benar.



Gambar 4. 13 Kumpulan sampel subjek

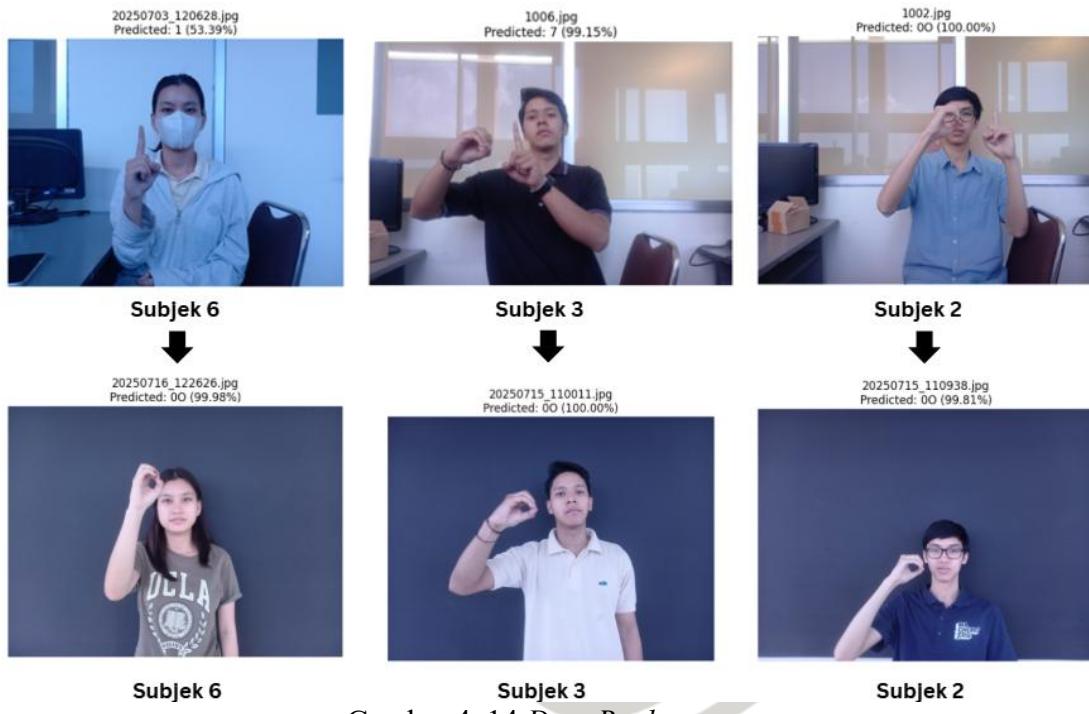
Seperti yang dapat dilihat pada Gambar 4.14, salah satu faktor utama yang diduga mempengaruhi hasil akurasi adalah perbedaan warna latar belakang (*background*) saat pengambilan data citra gestur. Pada subjek-subjek dengan akurasi tinggi seperti Subjek 5, Subjek 1, dan Subjek 6, citra diambil dengan latar belakang yang memiliki kontras tinggi terhadap warna kulit tangan, sehingga batas tepi (*edge*) tangan terlihat lebih jelas dan mudah diproses oleh model. Kejelasan kontur tangan ini mempermudah *MobileNetV1* sebagai *feature extractor* dalam menangkap pola-pola spasial yang relevan dari gerakan tangan.

Sebaliknya, pada subjek-subjek dengan akurasi rendah seperti Subjek 2, Subjek 3, dan Subjek 4, latar belakang yang digunakan memiliki warna yang hampir serupa dengan warna kulit, sehingga menyebabkan kesulitan dalam membedakan tangan dari latar belakang. Akibatnya, model mengalami kesulitan dalam mengekstraksi fitur yang relevan karena informasi visual yang kabur atau tidak kontras, yang pada akhirnya menurunkan akurasi prediksi. Hal ini menunjukkan bahwa aspek teknis seperti pencahayaan dan pemilihan *background* sangat berpengaruh terhadap efektivitas model dalam tugas klasifikasi citra gestur tangan, dan perlu diperhatikan dalam pengumpulan dataset maupun pengujian sistem secara real time.

#### 4.6.1. Evaluasi Data Baru ke-2

Untuk meningkatkan akurasi model klasifikasi gerakan angka dalam BISINDO, dilakukan proses pengambilan ulang data (data *replacement*) terhadap subjek yang sebelumnya menunjukkan performa pengenalan yang rendah, yakni subjek 2, subjek 3, dan subjek 6 seperti yang dapat dilihat dari Gambar 4.15. Berdasarkan evaluasi awal, rendahnya akurasi pada ketiga subjek tersebut disebabkan oleh faktor eksternal seperti warna kulit tangan yang cenderung menyatu dengan latar belakang, serta pencahayaan yang kurang mendukung. Untuk mengatasi permasalahan tersebut, pengambilan ulang dilakukan dengan menggunakan latar belakang berwarna hitam dan pencahayaan yang merata agar kontras antara tangan dan *background* menjadi lebih jelas. Hal ini bertujuan untuk meningkatkan deteksi fitur tangan oleh sistem, khususnya pada tahap ekstraksi ciri visual menggunakan model CNN berbasis *MobileNetV1*. Dengan kondisi pencahayaan yang lebih baik dan *background* yang seragam, diharapkan data baru

yang dihasilkan dapat memberikan representasi visual yang lebih optimal sehingga mendukung performa klasifikasi yang lebih akurat.



Gambar 4. 14 *Data Replacement*

Tabel 4. 4 Hasil accuracy menggunakan data baru ke-2 berdasarkan subjek

Subjek	Accuracy
1	57,27%
2	67,27%
3	57,27%
4	59,09%
5	70,91%
6	57,27%
Rata-rata	61,52%

Setelah data tersebut diproses kembali melalui pipeline klasifikasi yang sama, akurasi model mengalami peningkatan yang signifikan. Subjek 2, yang sebelumnya memiliki akurasi 16,36%, meningkat menjadi 67,27%. Subjek 3 dan subjek 6 masing-masing juga menunjukkan peningkatan akurasi dari 39,09% dan 47,27% menjadi

57,27%. Hasil ini menunjukkan bahwa kualitas latar belakang dan pencahayaan saat pengambilan gambar memiliki pengaruh besar terhadap keberhasilan proses klasifikasi citra, khususnya dalam mendukung ekstraksi fitur yang optimal oleh model.

Seperti yang ditampilkan pada Tabel 4.4, rata-rata akurasi model meningkat secara signifikan setelah penggunaan data baru dengan latar belakang yang seragam dan pencahayaan yang baik. Hal ini memperkuat bukti bahwa kualitas latar atau background memiliki pengaruh besar terhadap tingkat akurasi prediksi model. Salah satu penyebab utama dari rendahnya akurasi sebelumnya adalah ketidakmampuan model dalam mengenali kontur atau edge tangan secara jelas, terutama ketika tangan tidak kontras dengan latar belakang.

Selain itu, warna kulit tangan yang serupa dengan warna wajah atau pakaian turut menyebabkan penurunan performa model, karena sistem kesulitan dalam membedakan bagian tangan dari elemen visual lain dalam citra. Ketika fitur tangan tidak terdeteksi secara utuh, proses ekstraksi fitur menjadi tidak optimal, sehingga prediksi yang dihasilkan cenderung keliru atau bahkan gagal sepenuhnya. Oleh karena itu, pemilihan latar belakang yang tepat dan pencahayaan yang memadai menjadi aspek krusial dalam proses akuisisi data untuk klasifikasi citra berbasis gerakan tangan.

Tabel 4. 5 Hasil accuracy menggunakan data baru ke-2 berdasarkan gestur

Gestur	n	Persentase
0	60	82%
1	60	32%
2	60	55%
3	60	23%
4	60	72%
5	60	95%
6	60	83%
7	60	97%
8	60	52%

Gestur	n	Persentase
9	60	13%
10	60	73%
Rata-rata		62%

Pada Tabel 4.5, performa model klasifikasi terhadap masing-masing gestur dari 3 subjek yang telah diganti menunjukkan variasi yang lebih baik dibandingkan sebelumnya. Perubahan ini merupakan hasil dari proses pengambilan ulang data dengan latar belakang hitam dan pencahayaan yang lebih optimal, yang secara signifikan meningkatkan kualitas citra serta keterbacaan fitur visual seperti tepi tangan (edge). Dengan perbaikan tersebut, gestur yang sebelumnya sulit dikenali seperti angka 1, 2, 3, dan 9 kini menunjukkan peningkatan akurasi yang cukup mencolok. Sebagai contoh, gestur angka 1 yang sebelumnya hanya dikenali sebesar 22%, kini mencapai tingkat akurasi yang lebih baik pada subjek-subjek baru, menandakan bahwa visibilitas dan kontras warna antara tangan dan latar sangat berpengaruh terhadap keberhasilan prediksi model.

Secara keseluruhan, rata-rata akurasi dari gestur-gestur yang diuji meningkat, dengan banyak gestur menunjukkan konsistensi hasil yang lebih baik di antara ketiga subjek pengganti. Hal ini memperkuat temuan bahwa kualitas akuisisi data, termasuk pemilihan latar belakang dan pencahayaan, merupakan faktor penting dalam proses klasifikasi citra berbasis gestur tangan. Peningkatan akurasi setelah penggantian data subjek menunjukkan bahwa model *MobileNetV1* yang digunakan memiliki potensi yang baik dalam mengenali gestur BISINDO, asalkan data latih dan uji memiliki kualitas visual yang mendukung. Temuan ini juga memberikan implikasi penting terhadap proses akuisisi data di masa mendatang, yaitu perlunya standarisasi lingkungan pengambilan gambar guna meminimalkan variabilitas dan memaksimalkan performa model klasifikasi.

## Bab V

### Penutup

#### 5.1. Simpulan

Berdasarkan rumusan masalah dan tujuan yang telah ditetapkan dalam penelitian ini, dapat disimpulkan bahwa sistem klasifikasi citra untuk mengenali gerakan angka dalam Bahasa Isyarat Indonesia (BISINDO) berhasil dibangun menggunakan arsitektur *Convolutional Neural Network* (CNN) berbasis *MobileNetV1*. Model yang dikembangkan menunjukkan performa yang sangat baik pada data pengujian internal, dengan akurasi mencapai 99% serta nilai *precision*, *recall*, dan *f1-score* yang rata-rata berada pada kisaran 0.99. Hasil *confusion matrix* juga menunjukkan bahwa sebagian besar kelas berhasil diklasifikasikan dengan benar, dengan sangat sedikit kesalahan prediksi antarkelas.

Selain itu, model yang dibangun telah diuji pada data baru dari enam subjek berbeda guna mengevaluasi kemampuan generalisasi model terhadap input yang belum pernah dilihat sebelumnya. Hasil pengujian ini menunjukkan bahwa performa model bervariasi antar subjek, dengan akurasi berkisar antara 16,36% hingga 70,91%, dan rata-rata akurasi keseluruhan sebesar 48,33%. Hal ini menunjukkan bahwa meskipun model cukup efektif dalam mengenali pola dari data pelatihan, masih terdapat tantangan dalam hal kualitas maupun kuantitas terhadap variasi individu pengguna, posisi tangan, serta gaya gerakan yang berbeda-beda. Dengan demikian, arsitektur *MobileNetV1* terbukti efisien dan mampu menghasilkan model yang ringan dan akurat, namun untuk meningkatkan performa terhadap data baru, dibutuhkan perluasan dataset pelatihan dan mungkin integrasi teknik augmentasi atau personalisasi model lebih lanjut.

Berdasarkan hasil eksperimen dan analisis terhadap data baru, dapat disimpulkan bahwa kualitas latar belakang dan pencahayaan memiliki pengaruh yang signifikan terhadap akurasi model klasifikasi gerakan tangan berbasis CNN *MobileNetV1*. Peningkatan akurasi yang terjadi setelah dilakukan pengambilan ulang data dari subjek dengan performa rendah membuktikan bahwa ke tidak-jelasan kontur tangan serta kemiripan warna tangan dengan latar belakang atau bagian tubuh lain,

seperti wajah dan pakaian, menjadi faktor utama penyebab penurunan performa model. Penggunaan latar belakang hitam dan pencahayaan yang baik terbukti meningkatkan keterbacaan fitur visual oleh model, sehingga akurasi prediksi meningkat secara konsisten. Temuan ini menegaskan bahwa kondisi lingkungan saat proses akuisisi data perlu diperhatikan secara serius dalam pengembangan sistem klasifikasi citra berbasis bahasa isyarat.

## 5.2. Saran

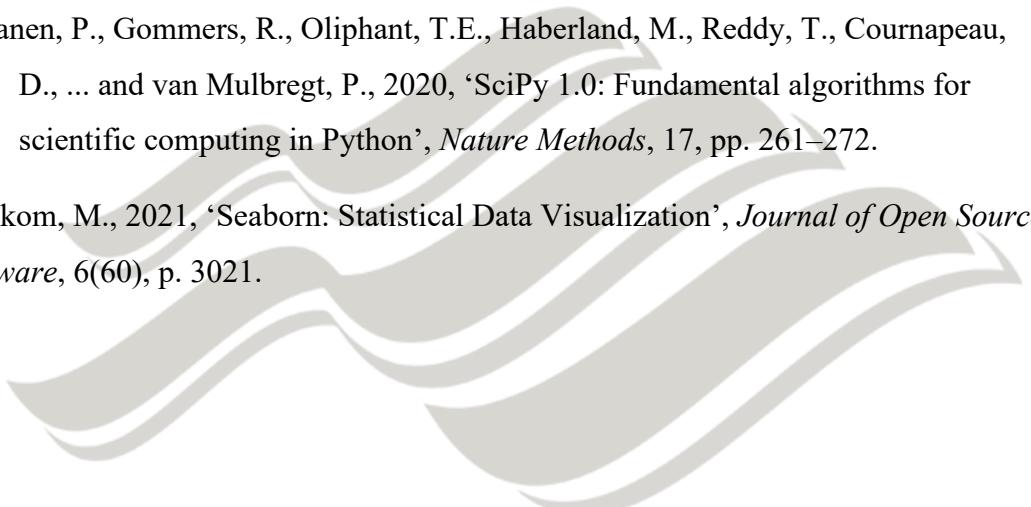
Untuk meningkatkan performa model dalam mengenali gerakan angka pada Bahasa Isyarat Indonesia (BISINDO), terutama terhadap data baru atau pengguna yang berbeda, disarankan untuk memperluas dan memperkaya *dataset* pelatihan dengan melibatkan lebih banyak partisipan dari latar belakang yang beragam. Hal ini bertujuan agar model dapat belajar berbagai variasi bentuk tangan, ukuran tangan, sudut pandang kamera, serta perbedaan gaya dalam melakukan gestur. Selain itu, penerapan augmentasi data yang lebih kompleks, seperti rotasi 3D atau simulasi pencahayaan dan latar belakang yang berbeda, juga dapat membantu memperkuat generalisasi model.

Selanjutnya, penggunaan metode *fine-tuning* tambahan atau *transfer learning* dengan model yang lebih besar dapat dipertimbangkan untuk meningkatkan kemampuan representasi fitur. Penerapan teknik evaluasi tambahan seperti *cross-validation* juga disarankan untuk mendapatkan hasil evaluasi yang lebih stabil. Terakhir, penyesuaian antarmuka dan sistem umpan balik juga perlu dirancang agar sistem lebih mudah digunakan oleh pengguna dengan berbagai kebutuhan khusus.

## Daftar Pusaka

- Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... and Zheng, X., 2016, ‘TensorFlow: A system for large-scale machine learning’, *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, pp. 265–283.
- Alexander, N., Widodo, R.B., & Swastika, W. Penggunaan Machine Learning Dalam Klasifikasi Bahasa Isyarat BISINDO Menggunakan Kamera. *Prosiding Seminar Nasional Universitas Ma Chung(SENAM)*, 2023, (pp. 11-26).
- Bradski, G., 2000, ‘The OpenCV Library’, *Dr. Dobb's Journal of Software Tools*.
- Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., ... and Adam, H., 2017, ‘MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications’, *arXiv preprint*, arXiv:1704.04861.
- Hunter, J.D., 2007, ‘Matplotlib: A 2D graphics environment’, *Computing in Science & Engineering*, 9(3), pp. 90–95.
- Krizhevsky, A., Sutskever, I. and Hinton, G.E., 2012, ‘ImageNet classification with deep convolutional neural networks’, *Advances in Neural Information Processing Systems*, 25, pp. 1097–1105.
- LeCun, Y., Bengio, Y. and Hinton, G., 2015, ‘Deep learning’, *Nature*, 521(7553), pp. 436–444.
- McKinney, W., 2010, ‘Data structures for statistical computing in Python’, *Proceedings of the 9th Python in Science Conference*, pp. 51–56.
- Nugraheni, A.S., Husain, A.P. and Unayah, H., 2021, ‘Optimalisasi penggunaan bahasa isyarat dengan SIBI dan BISINDO pada mahasiswa difabel tunarungu di Prodi PGMI UIN Sunan Kalijaga’, *Jurnal Holistik*, 5(1), pp. 28–33.
- Oliphant, T.E., 2006, *A Guide to NumPy*, USA: Trelgol Publishing.

- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... and Duchesnay, E., 2011, ‘Scikit-learn: Machine learning in Python’, *Journal of Machine Learning Research*, 12, pp. 2825–2830.
- Syahputra, Y.E., Widagdo, A.K. and Hadi, S., 2021, ‘Analisis Komparatif Struktur Bahasa SIBI dan BISINDO dalam Pembelajaran Bahasa Indonesia’, *Jurnal Pendidikan Bahasa dan Sastra Indonesia Metalingua*, 6(2), pp. 128–138.
- Virtanen, P., Gommers, R., Oliphant, T.E., Haberland, M., Reddy, T., Cournapeau, D., ... and van Mulbregt, P., 2020, ‘SciPy 1.0: Fundamental algorithms for scientific computing in Python’, *Nature Methods*, 17, pp. 261–272.
- Waskom, M., 2021, ‘Seaborn: Statistical Data Visualization’, *Journal of Open Source Software*, 6(60), p. 3021.



UNIVERSITAS  
**MA CHUNG**

## Lampiran

### Lampiran A. Data Kegiatan PKL

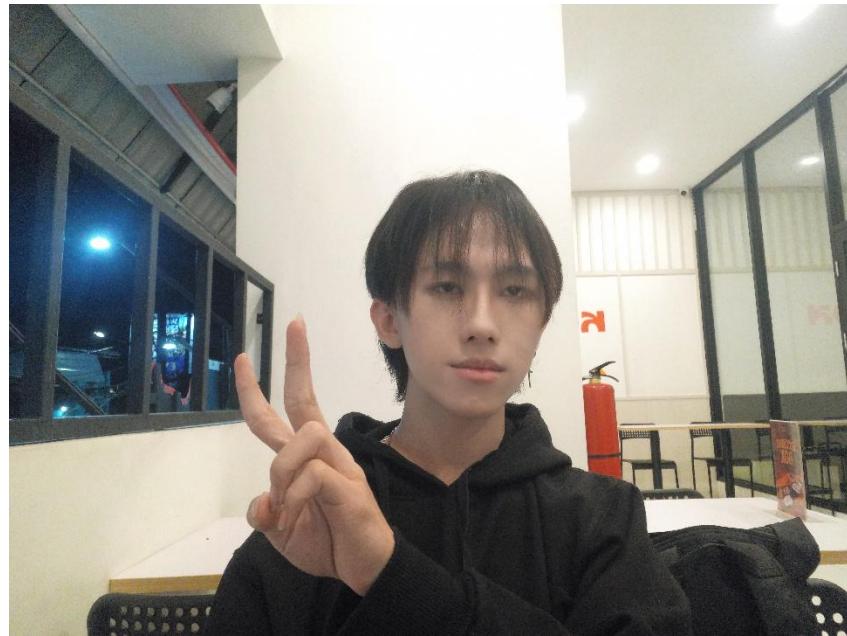
#### Lampiran A.1 Dokumentasi Pengambilan Data Subjek



Gambar A. 1 Gestur "0"



Gambar A. 2 Gestur "1"



Gambar A. 4 Gestur "2"



Gambar A. 3 Gestur "3"



Gambar A. 6 Gestur "4"



Gambar A. 5 Gestur "5"



Gambar A. 8 Gestur "6"



Gambar A. 7 Gestur "7"



Gambar A. 10 Gestur "8"



Gambar A. 9 Gestur "9"



Gambar A. 11 Gestur "10"

UNIVERSITAS  
**MA CHUNG**

**Lampiran B. Hasil Kegiatan PKL****Lampiran B.1 Tabel Persentase Pengujian Presisi menggunakan Data Baru**

Tabel B. 1 Tabel Persentase pengujian menggunakan Data Baru

Gestur	n	Persentase
0	60	72%
1	60	22%
2	60	27%
3	60	25%
4	60	43%
5	60	92%
6	60	60%
7	60	77%
8	60	38%
9	60	12%
10	60	65%
Rata-rata		48%

Tabel B. 2 Accuracy menggunakan data baru ke-2

Gestur	n	Persentase
0	60	82%
1	60	32%
2	60	55%
3	60	23%
4	60	72%
5	60	95%
6	60	83%
7	60	97%
8	60	52%

Gestur	n	Percentase
9	60	13%
10	60	73%
Rata-rata		62%

Tabel B. 3 Accuracy menggunakan data baru secara keseluruhan

Subjek I												
Angka	n	Pred iksi	Total Benar	Total Salah								
0	10	0	7	0	0	0	7	7	7	7	4	6
1	10	1	7	1	1	1	8	7	8	6	5	5
2	10	2	8	8	8	8	7	1	8	8	1	9
3	10	9	7	4	8	9	9	3	7	3	8	2
4	10	9	4	4	9	9	9	4	4	4	6	4
5	10	9	5	9	5	5	5	5	5	5	8	2
6	10	6	6	6	6	6	8	6	6	6	9	1
7	10	7	7	7	10	7	7	7	7	7	9	1
8	10	8	5	8	8	8	8	8	8	8	9	1
9	10	8	9	9	9	5	8	8	9	9	8	5
10	10	10	10	10	10	10	7	7	0	7	5	5
											Rata- rata	57,27 %

Subjek II												
Angka	n	Pred iksi	Total Benar	Total Salah								
		1	2	3	4	5	6	7	8	9	10	
0	10	0	0	0	0	7	7	0	7	7	0	6
1	10	0	0	7	0	0	7	5	0	7	7	0
2	10	7	7	0	7	0	7	7	7	7	0	10
3	10	7	0	5	0	0	0	0	10	0	0	10
4	10	7	7	7	0	0	5	7	5	5	0	10
5	10	5	9	5	5	5	5	5	5	5	9	1
6	10	5	7	7	7	7	7	7	7	7	0	10
7	10	7	7	7	7	7	7	7	7	7	10	0
8	10	7	7	7	0	7	7	0	0	0	0	10
9	10	7	5	7	7	7	7	7	7	7	0	10
10	10	0	0	0	7	7	0	7	7	10	2	8
											Rata- rata	16,36 %

Subjek III												
Angka	n	Prediksi	Total Benar	Total Salah								
		1	2	3	4	5	6	7	8	9	10	
0	10	5	7	0	0	7	0	0	0	0	7	3
1	10	7	7	7	0	0	7	7	7	7	0	10
2	10	7	4	7	4	4	10	4	3	7	7	0
3	10	5	3	3	7	4	3	4	3	3	3	6
4	10	7	7	7	7	4	7	7	7	7	4	2
5	10	5	5	5	5	5	5	5	5	5	10	0
6	10	7	7	7	5	7	7	7	7	7	0	10
7	10	7	7	7	7	7	7	7	7	7	10	0
8	10	7	7	7	7	7	7	7	7	7	0	10
9	10	5	7	7	7	7	7	7	7	7	0	10
10	10	10	10	10	7	7	10	10	10	10	8	2
											Rata-rata	39,09 %

Subjek IV												
Angka	n	Prediksi	Total Benar	Total Salah								
		1	2	3	4	5	6	7	8	9	10	
0	10	0	7	0	0	0	0	7	0	0	0	8
1	10	7	7	7	7	7	10	7	7	7	7	0
2	10	2	2	2	2	2	2	10	10	2	4	7
3	10	4	2	3	7	2	3	7	3	7	4	3
4	10	4	4	4	7	4	4	4	7	4	5	7
5	10	5	5	5	5	5	5	5	5	5	5	10
6	10	6	6	7	6	6	6	6	7	6	6	8
7	10	7	7	7	7	7	7	7	7	7	10	0
8	10	4	8	3	8	4	4	9	9	5	0	2
9	10	5	7	5	5	4	8	9	5	5	7	1
10	10	10	10	10	10	0	0	10	10	10	10	9
											Rata-rata	59,09 %

Subjek V												
Angka	n	Prediksi	Total Benar	Total Salah								
		1	2	3	4	5	6	7	8	9	10	
0	10	0	0	0	0	0	0	0	0	0	10	0
1	10	1	1	8	10	1	1	5	5	1	6	5
2	10	8	6	8	8	2	2	2	5	2	4	6
3	10	5	5	4	2	3	5	3	3	4	2	3
4	10	5	4	4	4	4	4	4	4	4	9	1
5	10	5	5	5	5	5	5	5	5	9	10	0

6	10	6	6	6	6	6	6	6	6	6	10	0	
7	10	7	7	7	7	7	7	7	7	7	10	0	
8	10	8	8	8	5	8	5	8	8	8	8	2	
9	10	5	8	9	5	5	5	8	5	5	1	9	
10	10	10	10	10	0	10	10	0	10	10	8	2	
												Rata-rata %	
<b>Subjek VI</b>												<b>70,91 %</b>	
Angka	n	Prediksi 1	Prediksi 2	Prediksi 3	Prediksi 4	Prediksi 5	Prediksi 6	Prediksi 7	Prediksi 8	Prediksi 9	Prediksi 10	Total Benar	Total Salah
0	10	0	0	0	0	0	8	0	0	8	0	8	2
1	10	1	6	0	0	0	5	0	8	1	1	3	7
2	10	2	2	2	0	0	8	2	8	8	8	4	6
3	10	5	3	2	2	2	2	2	5	5	2	1	9
4	10	9	3	5	4	5	2	4	5	8	2	2	8
5	10	5	5	5	1	5	5	9	5	5	5	8	2
6	10	6	6	6	6	6	6	1	6	6	6	9	1
7	10	5	7	5	5	7	7	7	7	5	6	6	4
8	10	8	2	5	8	2	8	8	2	2	2	4	6
9	10	5	5	5	5	4	5	5	5	8	0	10	
10	10	10	10	10	8	10	10	0	10	0	7	3	
												Rata-rata %	<b>47,27 %</b>

Tabel B. 4 Accuracy menggunakan data baru ke 2 secara keseluruhan

<b>Subjek II</b>													
Angka	n	Prediksi 1	Prediksi 2	Prediksi 3	Prediksi 4	Prediksi 5	Prediksi 6	Prediksi 7	Prediksi 8	Prediksi 9	Prediksi 10	Total Benar	Total Salah
0	0	0	0	10	0	0	0	0	0	0	9	1	0
1	0	1	0	1	10	0	10	1	1	1	5	5	0
2	2	2	2	3	2	2	2	4	2	2	8	2	2
3	4	2	4	3	4	3	3	2	3	3	5	5	4
4	4	4	4	4	4	4	4	5	4	4	9	1	4
5	5	5	5	5	5	5	5	5	5	5	10	0	5
6	1	6	1	6	6	6	6	6	6	6	8	2	1
7	7	7	7	7	7	7	7	7	7	7	10	0	7
8	0	2	5	2	5	5	2	2	5	8	1	9	0
9	3	5	5	3	5	5	5	5	5	5	0	10	3
10	10	10	10	10	10	10	10	10	2	9	1	10	
												Rata-rata %	<b>67,27 %</b>

Subjek  
III

Angka	n	Pred iksi	Total Benar	Total Salah								
		1	2	3	4	5	6	7	8	9	10	
0	10	0	0	0	0	0	6	0	0	0	9	1
1	10	5	6	6	6	5	6	5	5	6	0	10
2	10	2	2	2	2	5	2	2	1	2	2	8
3	10	1	2	2	2	1	3	5	4	2	2	1
4	10	4	4	4	2	4	4	4	5	3	7	3
5	10	5	5	5	5	5	5	5	5	5	10	0
6	10	6	6	6	6	6	6	5	5	6	7	3
7	10	7	7	7	7	7	7	7	7	7	10	0
8	10	1	5	5	8	8	8	8	5	8	8	6
9	10	5	5	5	8	4	5	5	5	5	0	10
10	10	10	0	0	10	0	6	0	0	10	10	5
											Rata- rata	57,27 %

Subjek  
VI

Angka	n	Pred iksi	Total Benar	Total Salah								
		1	2	3	4	5	6	7	8	9	10	
0	0	0	0	0	0	5	0	0	0	0	9	1
1	1	1	1	2	1	5	8	6	6	0	4	6
2	2	2	5	5	6	8	2	2	2	6	5	5
3	7	5	8	4	9	5	5	4	2	2	0	10
4	4	4	4	4	5	4	5	9	5	5	5	4
5	5	5	5	5	5	5	5	5	5	4	9	1
6	6	6	6	6	7	6	6	6	1	8	2	6
7	7	7	7	7	0	7	7	7	7	9	1	7
8	8	8	8	8	8	2	2	5	6	2	5	8
9	5	9	5	5	5	5	5	5	5	1	9	5
10	10	10	10	10	10	10	0	0	10	10	8	2
											Rata- rata	57,27 %