Klasifikasi Pengenalan Huruf Sistem Isyarat Bahasa Indonesia (SIBI) Menggunakan Algoritma Attention dan VGG-16

Nur Hikma Amir

22917032

Tesis diajukan sebagai syarat untuk meraih gelar Magister Komputer

Konsentrasi Data Sains

Program Studi Informatika Program Magister

Fakultas Teknologi Industri

Universitas Islam Indonesia

2025

Lembar Pengesahan Pembimbing

Klasifikasi Pengenalan Huruf Sistem Isyarat Bahasa Indonesia (SIBI) Menggunakan Algoritma Attention dan VGG-16

Nur Hikma Amir

22917032

Yogyakarta, 7 Mei 2025

Pembimbing 1	Pembimbing 2
Ir. Chandra Kusuma Dewa, S.Kom., M.Cs., Ph.D.	Dr. Ahmad Luthfi, S.Kom., M.Kom.

Lembar Pengesahan Penguji

Klasifikasi Pengenalan Huruf Sistem Isyarat Bahasa Indonesia (SIBI) Menggunakan Algoritma Attention dan VGG-16

Nur Hikma Amir

22917032

Yogyakarta, 7 Mei 2025

Tim Penguji,

Ir. Chandra Kusuma Dewa, S.Kom., M.Cs., Ph.D.
Ketua

Dr. Ahmad Luthfi, S.Kom., M.Kom.
Anggota I

Dr. Feri Wijayanto, S.T., M.T.
Anggota II

Mengetahui,
Ketua Program Studi Informatika Program Magister
Fakultas Teknologi Industri
Universitas Islam Indonesia

Irving Vitra Paputungan, ST., M.Sc., Ph.D.

Abstrak

Diversitas gaya komunikasi non-verbal bagi penyandang disabilitas di Indonesia mendesak solusi teknologi yang mendukung fasilitas kebutuhan disabilitas dalam lingkungan kerja dan aspek sosial. Penelitian bertujuan menawarkan pendekatan baru dalam meningkatkan kinerja jaringan saraf menggunakan arsitektur sederhana untuk pengenalan huruf SIBI (M dan N) melalui metode *keypoint detection*, dengan memperhatikan variasi ukuran tangan (17–22 cm). Empat model dievaluasi: YOLOv5 berbasis *image detection*, serta VGG-16, Attention, dan Multi-Layer Perceptron (MLP) yang dikembangkan menggunakan *keypoint detection*. Evaluasi dilakukan secara real-time dengan mempertimbangkan aksesori seperti cincin, jam tangan, dan sarung tangan, serta variasi intensitas pencahayaan untuk mensimulasikan kondisi nyata di lingkungan pengguna. Kebaruan terletak pada integrasi *keypoint detection* ke dalam arsitektur sederhana, yang terbukti meningkatkan akurasi dan ketahanan terhadap gangguan visual (noise). Model MLP menunjukkan performa terbaik dengan akurasi 94% (M) dan 93% (N), mengungguli pendekatan kompleks seperti YOLOv5 yang turun drastis dalam kondisi terganggu. Integrasi VGG-16 dengan Attention justru menghasilkan *underfitting*, menegaskan bahwa kompleksitas tidak selalu berkorelasi dengan efektivitas. Temuan ini menyoroti potensi model ringan untuk digunakan memperluas aksesibilitas teknologi bagi komunitas disabilitas di berbagai konteks sosial dan profesional.

Kata Kunci: Disabilitas; *Keypoint Detection*; huruf M dan N; SIBI; Bahasa Isyarat;

Abstract

The diversity of non-verbal communication styles among persons with disabilities in Indonesia highlights the urgent need for technological solutions that support accessibility in both workplace settings and social contexts. This study proposes a novel approach to improving neural network performance through the use of simple architectures for recognizing Indonesian Sign Language (SIBI) letters M and N, by applying keypoint detection while accounting for hand size variations (17–22 cm). Four models were evaluated: YOLOv5 based on image detection, as well as VGG-16, Attention, and Multi-Layer Perceptron (MLP) developed using keypoint detection. The evaluation was conducted in real-time, taking into account accessories such as rings, watches, and gloves, as well as varying lighting intensities to simulate real-world user environments. The novelty lies in the integration of keypoint detection into lightweight architectures, which significantly improves accuracy and resilience against visual disturbances (noise). The MLP model achieved the best performance, with an accuracy of 94% for M and 93% for N, outperforming more complex approaches such as YOLOv5, which showed a significant drop in accuracy under disturbed conditions. The integration of VGG-16 with Attention resulted in underfitting, emphasizing that complexity does not always correlate with effectiveness. These findings underscore the potential of lightweight models to enhance technological accessibility for the disabled community across various social and professional domains.

Keywords: Disability; Keypoint Detection; Letter M and N; SIBI; Sign Language

Pernyataan Keaslian Tulisan

Dengan ini saya menyatakan bahwa tesis ini merupakan tulisan asli dari penulis, dan tidak berisi material yang telah diterbitkan sebelumnya atau tulisan dari penulis lain terkecuali referensi atas material tersebut telah disebutkan dalam tesis. Apabila ada kontribusi dari penulis lain dalam tesis ini, maka penulis lain tersebut secara eksplisit telah disebutkan dalam tesis ini.

Dengan ini saya juga menyatakan bahwa segala kontribusi dari pihak lain terhadap tesis ini, termasuk bantuan analisis statistik, desain survei, analisis data, prosedur teknis yang bersifat signifikan, dan segala bentuk aktivitas penelitian yang dipergunakan atau dilaporkan dalam tesis ini telah secara eksplisit disebutkan dalam tesis ini.

Segala bentuk hak cipta yang terdapat dalam material dokumen tesis ini berada dalam kepemilikan pemilik hak cipta masing-masing. Apabila dibutuhkan, penulis juga telah mendapatkan izin dari pemilik hak cipta untuk menggunakan ulang materialnya dalam tesis ini.

Yogyakarta, 7 Mei 2025



Nur Hikma Amir, S.Kom.,MTA

Daftar Publikasi

<u>Publikasi yang menjadi bagian dari tesis</u>

Amir, Nur Hikma., Dewa Chandra K, Luthfi Ahmad. (2025). "Refining the Performance of Neural Networks *with* Simple Architectures for Indonesian Sign Language System (SIBI) Letter Recognition Using Keypoint Detection". *Jurnal ILKOM, Vol.17, No. 1, April 2025*

Kontributor	Jenis Kontribusi
Nur Hikma Amir	Mendesain eksperimen (60%) Menulis *paper* (70%)
Chandra Kusuma Dewa	Mendesain eksperimen (20%) Menulis dan mengedit *paper* (15%)
Ahmad Luthfi	Mendesain eksperimen (20%) Menulis dan mengedit *paper* (15%)

Halaman Kontribusi

Dalam tesis ini, terdapat beberapa kontribusi dari pihak lain, diantaranya:

1. Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi Republik Indonesia (Kemendikbud Ristek) yang telah mendanai penelitian ini, melalui Program Hibah Penelitian Pascasarjana-Penelitian Tesis Magister (PPS-PTM), sesuai dengan Surat Keputusan Nomor 0459/E5/PG.02.00/2024 dan Surat Perjanjian/Kontrak Nomor 0609.1/LL5-INT/AL.04/2024. Penulis mengucapkan terima kasih yang sebesar-besarnya kepada Kemendikbud Ristek atas dukungan diberikan finansial, memungkinkan penulisan untuk melakukan penelitian dan penulisan publikasi artikel dan tesis secara mendalam.
2. Keluarga Besar di Desa Benteng Gantarang ucapan terima kasih yang mendalam saya sampaikan kepada keluarga besar di Desa Benteng Gantarang, yang telah membantu dalam proses pengambilan data dan dokumentasi gerakan tangan untuk penelitian ini. Dukungan dan partisipasi mereka memberikan kontribusi besar terhadap keakuratan dataset yang digunakan dalam pengembangan model klasifikasi huruf SIBI.
3. Seluruh dosen Fakultas Ilmu Komputer Universitas Muslim Indonesia (FIKOM UMI), khususnya kepada Ibunda Lilis Nur Hayati, S.Kom.,M.Eng.,MTA., atas bimbingan, dukungan, dan ilmu yang sangat berharga. Ucapan terima kasih juga kami sampaikan kepada mahasiswa Program Kreativitas Mahasiswa (PKM) angkatan 2016 - 2024 yang telah turut serta dalam berbagai diskusi tengah malam dan pengembangan tesis ini, termasuk Achmal, Almar, Rial, dan mahasiswa lainnya yang telah berkontribusi dalam berbagai aspek, mulai dari diskusi dan ide pengembangan, hingga pengujian dan penyusunan laporan. Kontribusi yang diberikan sangat berarti dalam keberhasilan tesis ini, dan saya berharap hasil kolaborasi ini dapat memberikan manfaat luas serta menjadi pijakan bagi penelitian dan pengembangan selanjutnya. Terima kasih kepada semua pihak yang telah berkontribusi dalam tesis ini.

Halaman Persembahan

Dengan penuh rasa syukur dan terima kasih yang mendalam, karya ini saya persembahkan kepada pihak-pihak yang telah memberikan cinta, doa, dukungan, serta semangat selama perjalanan akademik di Program Studi Informatika, Program Magister Universitas Islam Indonesia.

Pertama, penulis ingin menyampaikan penuh kasih dan rasa hormat kepada kedua orang tua tercinta, H. Muh Amir dan H. Suhaena, yang dengan penuh kasih sayang, doa, dan pengorbanan tanpa henti selalu menjadi cahaya dalam setiap langkah perjalanan penulis. Dukungan dan restu kalian adalah kekuatan terbesar dalam menggapai impian ini.

Selanjutnya, kepada saudara-saudaraku tersayang, Nadia Napis Amir, Sri Mulyani Amir, dan Muh Rafif Amir, yang selalu memberikan motivasi, semangat, dan kebersamaan dalam setiap perjalanan hidup. Terima kasih atas kehangatan dan dukungan tanpa batas yang membuat setiap tantangan menjadi lebih ringan.

Kepada Lilis Nur Hayati, S.Kom., M.Eng., MTA beserta keluarga, yang telah menjadi panutan dalam perjalanan akademik ini. Bimbingan, ilmu, serta inspirasi yang diberikan telah menjadi bekal berharga bagi penulis dalam menempuh perjalanan studi. Untuk sahabat seperjuangan selalu ada dalam suka maupun duka, Nur Almar'atussaliha, yang sedang melanjutkan studi yang sama, terima kasih atas kebersamaan, dukungan, serta semangat tak pernah pudar. Perjalanan ini terasa lebih bermakna karena ada teman seperjuangan yang saling menguatkan. Tak lupa, kepada Waode Tanti dan Irsal Arda, teman-teman angkatan Pixel 2016, yang selalu menjadi bagian dari perjalanan ini. Terima kasih atas setiap momen kebersamaan, perjuangan, serta semangat yang telah dibangun bersama.

Semoga setiap langkah yang telah dilalui membawa keberkahan dan manfaat, serta menjadi awal dari perjalanan baru yang lebih baik. Terima kasih atas segala cinta, doa, dan dukungan yang tiada henti.

Kata Pengantar

Assalamu Alaikum Warahmatullahi Wabarakatuh.

Bismillahir rohmaani rohiim, Alhamdulillah Segala pujian hanya milik Allah *Subhanahu Wata'ala*, yang atas rahmat, hidayah, taufiq dan karunia-Nya, penulis dapat menyelesaikan Tesis yang berjudul "Klasifikasi Pengenalan Huruf Sistem Isyarat Bahasa Indonesia (SIBI) Menggunakan Algoritma *Attention* dan VGG-16". Tesis ini disusun guna melengkapi sebagian syarat untuk memperoleh gelar Magister di Program Studi Informatika, Fakultas Teknik Industri, Universitas Islam Indonesia.

Dalam penyusunan dan penyelesaian tesis ini penulis telah menerima banyak bantuan dan dukungan dari berbagai pihak. Oleh karena itu, penulis menyampaikan ucapan terima kasih yang tulus kepada:

1. Bapak Prof. Fathul Wahid, S.T., M.Sc., Ph.D., selaku Rektor Universitas Islam Indonesia, atas dukungan dan arahannya selama studi saya di universitas ini.
2. Bapak Irving Vitra Paputungan, S.T., M.Sc., Ph.D., selaku Ketua Program Studi Magister Informatika Universitas Islam Indonesia, yang telah memberikan bimbingan dan fasilitas selama proses studi.
3. Bapak Ir. Chandra Kusuma Dewa, S.Kom., M.Cs., Ph.D., dan Bapak Dr. Ahmad Luthfi, S.Kom., M.Kom selaku dosen pembimbing, yang telah memberikan banyak saran/usulan, motivasi, ide, serta kritik yang membangun selama penyusunan tesis ini.
4. Dosen Program Studi Magister Informatika Universitas Islam Indonesia yang telah memberikan ilmu, bimbingan dan bantuan selama masa studi. Semoga ilmu yang diberikan menjadi amal jariyah yang bermanfaat.
5. Staff Akademik Program Studi Magister Informatika Universitas Islam Indonesia, yang telah membantu dalam segala urusan administrasi.
6. Ibu Lilis Nur Hayati, S.Kom.,M.Eng.,MTA selaku Sekretaris Program Studi Sistem Informasi Universitas Muslim Indonesia, atas dukungan, arahan, dan saran dalam penyelesain pendidikan magister ini.
7. Sahabat dan Teman yang tidak dapat penulis sebutkan satu persatu, atas segala dukungan dan bantuan dalam bentuk apapun.

Penulis menyadari bahwa tesis ini masih memiliki kekurangan dan kelemahan. Oleh karena itu, penulis sangat terbuka terhadap kritik dan saran yang sifatnya membangun agar tesis ini dapat menjadi lebih baik di masa mendatang.

Wassalamu'alaikum Warahmatullahi Wabarakatuh,

Daftar Isi

Lembar Pengesahan Pembimbing i
Lembar Pengesahan i
Lembar Pengesahan Penguji ii
Abstrak iii
Abstract iv
Pernyataan Keaslian Tulisan v
Daftar Publikasi vi
Halaman Kontribusi vii
Halaman Persembahan viii
Kata Pengantar ix
Daftar Isi x
Daftar Tabel xiii
Daftar Gambar xiv
Glosarium xv
BAB 1 Pendahuluan 1
1.1 Latar Belakang 1
1.2 Rumusan Masalah 3
1.3 Tujuan Penelitian 3
1.4 Batasan Masalah 3
1.5 Manfaat Penelitian 4
1.6 Sistematika Penulisan 5
BAB 2 Tinjauan Pustaka 6
2.1 *State of The Art* 6
2.2 Sistem Bahasa Isyarat Indonesia (SIBI) 11
2.3 Skenario Penelitian 12
2.3.1 YOLOv5 12
2.3.2 VGG-16 13
2.3.3 VGG-16 *with* Attention 14
2.3.4 MLP (*Multi-Layer Perceptron*) 14
2.4 Objek Detection 15
2.4.1 *Image Detection* 15
2.4.2 *Keypoint Detection* 15

2.5 *Confusion Matrix* 16
2.6 K-Fold Cross validation 17
BAB 3 Metodologi 18
3.1 Tahapan Penelitian 18
3.1.1 Identifikasi Masalah 18
3.1.2 Studi Literatur 19
3.1.3 Pengumpulan Data 19
3.1.4 *Pra-Processing* 20
3.1.5 Model Eksperimen 21
3.1.6 Evaluasi Model 24
3.1.7 Representasi Hasil 24
3.1.8 Kesimpulan 25
3.2 Desain Penelitian 25
3.2.1 Skenario Pertama : YOLOv5 25
3.2.2 Skenario Kedua : VGG-16 27
3.2.3 Skenario Ketiga : VGG-16 *with* Attention 28
3.2.4 Skenario Keempat : MLP 28
3.3 Metode Penelitian 29
3.3.1 Waktu dan Lokasi Penelitian 29
3.3.2 Sumber Data 29
3.3.3 Pemilihan Sampel 29
3.3.4 Skenario Eksperimen 30
3.3.5 Teknik Augmentasi 30
BAB 4 Hasil dan Pembahasan 31
4.1 Hasil Skenario Penelitian 31
4.1.1 Hasil Eksperimen YOLOv5 31
4.1.2 Hasil Eksperimen VGG-16 33
4.1.3 Hasil Eksperimen VGG-16 *with* Attention 35
4.1.4 Hasil Eksperimen MLP 36
4.2 Pembahasan 37
BAB 5 Kesimpulan dan Saran 42
5.1 Kesimpulan 42
5.2 Saran 43
Daftar Pustaka 44

Lampiran A 47
Lampiran B 51

Daftar Tabel

Tabel 2.1 Ulasan Krisis Tema Penelitian 6
Tabel 2.2 *Confusion matrix* 17
Tabel 3.1 Arsitektur *Multi-Layer Perceptron* 23
Tabel 4.1 Hasil Skenario Pertama YOLOv5 32
Tabel 4.2 Hasil Skenario Kedua VGG-16 33
Tabel 4.3 *Cross Validation* K=5: VGG-16 34
Tabel 4.4 Hasil Eksperimen Keempat MLP 36
Tabel 4.5 *Cross Validation* K=5: MLP 36
Tabel 4.6 Perbandingan Performa Model Pengenalan Huruf SIBI (Huruf M dan N) 37
Tabel 4.7 Perbandingan Arsitektur 39

Daftar Gambar

Gambar 2.1 Contoh Huruf SIBI M dan N 11
Gambar 2.2 Arsitektur dan Peta Jalan VGG-16 13
Gambar 2.3 Arsitektur *VGG-16 with Attention* 14
Gambar 2.4 Titik Koordinat *Hand Landmark* Model 16
Gambar 2.5 Contoh *MediaPipe landmark* 16
Gambar 2.6 *K-fold cross validation* 17
Gambar 3.1 Tahapan Penelitian 18
Gambar 3.2 Model Eksperimen YOLOv5 21
Gambar 3.3 Arsitektur VGG-16 22
Gambar 3.4 Arsitektur VGG-16 *with* Attention 23
Gambar 3.6 Diagram Alir Perancangan Images *Detection* 25
Gambar 3.7 Diagram Alir Perancangan *Keypoint Detection* 27
Gambar 3.8 *Loss Function danPerformance Metrics* 32
Gambar 4.1 Hasil *Confusion Matrix* 31
Gambar 4.2 Hasil *Precision - Recall* dan *F1-Score* 32
Gambar 4.3 Hasil dari *Bounding Box* 33
Gambar 4.4 Model Evaluasi dan *Confusion Matrix* Model VGG-16 34
Gambar 4.5 Pengujian Dalam Ruangan (Normal dan Sarung Tangan) 34
Gambar 4.6 Model Evaluation Model VGG-16 *with* Attention 35
Gambar 4.7 *Confusion Matrix* Model MLP 36
Gambar 4.8 Kondisi Pengujian M dan N 38
Gambar 4.9 Contoh Visualisasi Parameter 40

Glosarium

SIBI	- Sistem Isyarat Bahasa Indonesia
BISINDO	- Bahasa Isyarat Indonesia
VGG	- *Visual Geometry Group*
ASL	- *American Sign Language*
ISL	- *Indian Sign Language*
LIS	- *Lingua dei Segni Italiana*
TID	- *Türk İşaret Dili*
YOLO	- *You Only Look Once*
MLP	- *Multi-Layer Perceptron*
ResNet	- *Residual Network*
SPP	- *Spatial Pyramid Pooling*
SELayer	- *Squeeze-and-Excitation Layer*
PaNet	- *Path Aggregation Network*
NLP	- *Natural Language Processing*
ANN	- *Artificial Neural Network*
CNN	- *Convolutional Neural Network*
R-CNN	- *Region-Based Convolutional Neural Network*
SSD	- *Single Shot MultiBox Detector*
TP	- *True Positive*
FP	- *False Positive*
FN	- *False Negative*
TN	- *True Negative*
AWLDA	- *Attention-based Weighted Linear Discriminant Analysis*

BAB 1 Pendahuluan

1.1 Latar Belakang

Diversitas gaya berkomunikasi dan berinteraksi sering kali dianggap hanya melibatkan komunikasi verbal dan tertulis (Suardi, 2023). Namun, ada sejumlah besar individu mengalami kesulitan dalam menggunakan bahasa secara efektif akibat keterbatasan fisik atau mental (Lilis Nur Hayati et al. 2023). Menurut laporan Badan Pusat Statistik (BPS) menunjukkan bahwa jumlah pekerja disabilitas di Indonesia mencapai 720,748 orang pada tahun 2022 atau setara 0.53% dari total penduduk bekerja di Indonesia yang berjumlah 131,05 juta (Cindy Mutia Annur, 2023). Fakta tersebut mengilustrasikan pentingnya memperhatikan dan memfasilitasi kebutuhan komunikasi kelompok penyandang disabilitas, bukan hanya di lingkungan kerja tetapi juga dalam beragam aspek kehidupan sosial (Yinatan, 2023).

Hal ini penting untuk memastikan bahwa mereka dapat berpartisipasi secara penuh dan efektif dalam masyarakat, yang pada gilirannya akan membantu menciptakan lingkungan yang lebih inklusif dan adil bagi semua warga negara. Pemerintah telah menegaskan pada UUD No.8 Tahun 2016 Pasal 41 (1) Pemerintah dan Pemerintah Daerah dalam menyelenggarakan dan memfasilitasi pendidikan inklusif dan pendidikan khusus sebagaimana dimaksud dalam pasal 40 ayat (2) wajib memfasilitasi penyandang disabilitas untuk mempelajari keterampilan dasar yang dibutuhkan untuk kemandirian dan partisipasi penuh dalam menempuh pendidikan dan pengembangan sosial.

Setiap gerakan tangan, posisi jari, dan ekspresi wajah dalam bahasa isyarat memiliki makna spesifik, sehingga konsistensi dalam penggunaannya menjadi kunci agar pesan tidak disalahartikan. Upaya meningkatkan akurasi pengenalan huruf SIBI peneliti mengidentifikasi empat faktor permasalahan utama. Pertama, **faktor lingkungan** berdasarkan hasil sosialisasi dilakukan di SLB 02 Makassar pada tahun 2019 (Konten Redaksi Rakyatku.Com, 2019), (Umar et al., 2019), ditemukan bahwa peserta didik mengalami kesulitan dalam membedakan kedua huruf SIBI (M dan N), yang mengindikasikan adanya hambatan dalam aspek lingkungan belajar dan perlunya pendekatan teknologi lebih presisi. Kesulitan serupa ditemukan dalam penggunaan huruf ASL (*American Sign Language*) (Qin et al., 2023), ISL (*Indian Sign Language*)

(Sajeena et al., 2020), ASL (*Arab Sign Language*) (M. J. AL Moustafa et al., 2024), LIS (*Lingua dei Segni Italiana*) (Halder & Tayade, 2021), TID (*Türk İşaret Dili*) (Sevli & Kemaloglu, 2020), dan BISINDO (Bahasa Isyarat Indonesia) (Nur Hayati Lilis, Nur Handayani Anik, Wahyu Sakti Gunawan Iriantoa, Rosa Andrie Asmara, Dolly Indra, 2023), menunjukkan bahwa faktor lingkungan lokal berperan penting dalam keberhasilan sistem pengenalan bahasa isyarat. Kedua, berdasarkan **faktor literatur view** huruf M dan N mendapatkan akurasi pengenalan yang rendah, berbagai arsitektur *machine learning* (SVM, KNN, Random Forest (Ewe et al., 2024), *Decision Tree* dan *Naïve Bayes*) dan *deep learning* (ANN, MLP (Bagaskoro et al., 2023), YOLOv5) telah digunakan tetap menunjukkan kecenderungan kesalahan klasifikasi akibat kemiripan konfigurasi gestur tangan. Ketiga, **faktor eksternal** yang ditemukan melalui pengujian mencakup kondisi pencahayaan, sudut kamera, serta keberadaan aksesoris (cincin, jam tangan, sarung tangan) dapat menurunkan kualitas citra dan mengganggu sistem pendeteksian. Keempat, **faktor intrinsik** berasal dari variasi fisiologis seperti ukuran dan bentuk tangan serta keterbatasan motorik individu dengan disabilitas mempengaruhi konsistensi bentuk gestur pada akurasi model pembelajaran.

Kombinasi keempat faktor tersebut menggarisbawahi pentingnya pengembangan sistem pengenalan bahasa isyarat yang tidak hanya cerdas secara teknis, tetapi juga adaptif terhadap kondisi nyata di lapangan. Penelitian terdahulu lebih banyak berfokus pada faktor eksternal seperti pencahayaan, sudut kamera, dan gangguan latar yang mempengaruhi kualitas citra dalam sistem pengenalan bahasa isyarat. Pendekatan ini sering mengabaikan faktor intrinsik berkaitan langsung dengan variasi ukuran tangan dan bentuk gestur antar individu. Padahal, perbedaan fisiologis tersebut memiliki dampak signifikan terhadap akurasi klasifikasi, terutama pada huruf dengan konfigurasi serupa seperti huruf M dan N.

Kebaruan penelitian terletak pada penggunaan *keypoint detection* berbasis kerangka tangan dengan 21 titik *landmark* tangan menggunakan *MediaPipe*. Setiap titik *landmark* direpresentasikan sebagai koordinat dalam ruang tiga dimensi (x, y, z) mencerminkan topologi tangan secara presisi. Tidak seperti metode berbasis pixel intensity yang bergantung pada struktur citra penuh seperti *image detection* dan semantic segmentation, pendekatan ini memanfaatkan hubungan spasial antar titik sebagai vektor fitur utama. Representasi mengurangi variabilitas akibat latar, pencahayaan, dan tekstur terhadap gestur dengan konfigurasi mirip.

Maka atas dasar itu, peneliti merancang empat skenario eksperimen untuk mengevaluasi efektivitas berbagai arsitektur jaringan saraf dalam mengklasifikasikan huruf SIBI (M dan N) rentan kesalahan klasifikasi. Pendekatan ini melibatkan perbandingan antara model berbasis *image detection* (YOLOv5) (Alayed, 2024; Sajeena et al., 2020; Tazyeen Fathima et al., 2024), arsitektur konvolusional (VGG-16), integrasi Attention, dan MLP dengan pendekatan berbasis *keypoint detection*. Tujuan utama penelitian adalah untuk menyempurnakan kinerja *neural networks* dengan arsitektur sederhana untuk pengenalan huruf sistem isyarat Bahasa Indonesia (SIBI) dengan konfigurasi kompleks menggunakan representasi spasial tangan, tanpa ketergantungan pada fitur visual menyeluruh. Penelitian berkontribusi mengusulkan berbasis *keypoint detection* mampu menormalisasi skala dataset pelatihan, sehingga meningkatkan akurasi penganalan huruf SIBI berbagai ukuran tangan, suatu pendekatan yang belum banyak diterapkan pada pengenalan sistem bahasa isyarat.

1.2 Rumusan Masalah

Adapun rumusan masalah pada permasalahn tersebut, sebagai berikut:

1. Bagaimana performa eksperimen untuk mengenali setiap huruf dalam SIBI?
2. Apakah kombinasi VGG-16 dan metode Attention dapat meningkatkan performa model untuk mengenali huruf tertentu (huruf M dan N)?

1.3 Tujuan Penelitian

Adapun tujuan yang ingin dicapai dalam penelitian ini adalah:

1. Menganalisis performa eksperimen dalam mengenali setiap huruf dalam Sistem Isyarat Bahasa Indonesia (SIBI) dengan menggunakan metode *deep learning*.
2. Menguji apakah kombinasi arsitektur VGG-16 dan Attention dapat meningkatkan akurasi model dalam mengenali huruf M dan N dibandingkan dengan arsitektur sederhana seperti MLP.

1.4 Batasan Masalah

Adapun batasan masalah dalam penelitian ini sebagai berikut:

1. Skenario eksperimen terdiri dari empat yaitu : (1) YOLOv5 (2) VGG-16 (3) VGG-16 *with* Attention, dan (4) MLP.
2. Fokus eksperimen menggunakan *keypoint detection* yaitu model VGG-16, VGG-16 *with* Attention dan MLP dengan khusus huruf M dan N. Alasan, karena YOLOv5

sudah dilakukan oleh peneliti terdahulu menggunakan *image detection* dengan 26 huruf SIBI.

3. Eksperimen menggunakan model YOLOv5 pada 26 kelas huruf SIBI, terdiri dari 1872 gambar mentah yang di augmentasi menjadi 4492 gambar dengan teknik seperti flip, pengaturan kecerahan (-35% hingga +35%), dan *shear* dengan sudut ±10°. Eksperimen dilakukan untuk *objek detection*.
4. Eksperimen dengan model *VGG-16*, *VGG-16 with Attention* dan MLP digunakan dengan 2 kelas huruf SIBI (huruf M dan N) menggunakan 346 gambar yang sudah diformat dalam *.csv*, mencakup titik-titik *keypoint detection* dari *MediaPipe Landmark*.
5. Pengujian dilakukan baik di dalam ruangan maupun di luar ruangan.
6. Pengujian dilakukan dengan penggunaan aksesori seperti cincin, jam tangan, dan sarung tangan.
7. Sumber data terkait huruf SIBI berasal dari observasi lapangan di masyarakat Desa Benteng Gantarang.

1.5 Manfaat Penelitian

1. Bagi Peneliti

   Dapat mengimplementasikan pengetahuan, ilmu, dan skill yang telah di dapatkan di bangu perkuliahan, sehingga menjadi mahasiswa aktif, kreatif dan inovatif. Serta dapat peka terhadap masalah yang dihadapi lingkungan sekitar.
2. Bagi Penyandang Disabilitas dan Non-Disabilitas

   Penelitian ini bermanfaat bagi penyandang disabilitas, khususnya tunarungu, dengan meningkatkan aksesibilitas komunikasi melalui sistem otomatis yang mengenali huruf dalam Sistem Isyarat Bahasa Indonesia (SIBI), sehingga mereka dapat berkomunikasi lebih mandiri dan mudah. Bagi non-disabilitas, penelitian ini dapat mempermudah interaksi dengan pengguna bahasa isyarat, mendukung edukasi SIBI, serta mendorong pengembangan teknologi inklusif yang lebih adaptif dan bermanfaat bagi masyarakat luas.
3. Bagi Universitas

   Menambah referensi dan literatur view sebagai bahan penelitian lanjutan yang lebih mendatang pada masa yang akan datang.

1.6 Sistematika Penulisan

Sistematika Penulisan Laporan Tesis ini dijelaskan sebagai berikut.

BAB 1 Pendahuluan

Berisi tentang latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, sistematika penulisan.

BAB 2 Tinjauan Pustaka

Berisi tentang kajian teori dan konsep yang relevan dengan penelitian, seperti pemahaman tentang *State of the Art*, Sistem Bahasa Isyarat Indonesia (SIBI), serta teori dan teknologi digunakan dalam skenario penelitian, seperti YOLOv5, VGG-16, VGG-16 with Attention, dan MLP. Selain itu, juga dibahas mengenai *object detection*, *keypoint detection*, *confusion matrix*, serta metode validasi *K-Fold Cross Validation*..

BAB 3 Metodologi

Berisi tentang tahapan penelitian meliputi identifikasi masalah, studi literatur, pengumpulan data, *pre-processing*, model eksperimen (YOLOv5, VGG-16, VGG-16 *with* Attention, dan MLP), evaluasi model, representasi hasil, dan kesimpulan awal. Bab ini juga menjelaskan desain penelitian berdasarkan skenario, serta metode penelitian mencakup waktu dan lokasi, sumber data, pemilihan sampel, skenario eksperimen, dan teknik augmentasi.

BAB 4. Hasil dan Pembahasan

Berisi tentang hasil eksperimen dari masing-masing model/arsitektur digunakan (YOLOv5, VGG-16, VGG-16 *with* Attention, dan MLP), serta pembahasan yang mendalam mengenai perbandingan performa masing-masing model berdasarkan hasil yang diperoleh..

BAB 5 Kesimpulan dan Saran

Berisi tentang kesimpulan dari hasil penelitian yang telah dilakukan serta saran yang dapat diberikan untuk pengembangan penelitian lebih lanjut.

BAB 2 Tinjauan Pustaka

2.1 *State of The Art*

Perkembangan teknologi kecerdasan buatan, khususnya dalam bidang *deep learning*, telah memberikan kemajuan signifikan dalam sistem pengenalan bahasa isyarat. Pengenalan huruf dalam Sistem Isyarat Bahasa Indonesia (SIBI) maupun bahasa isyarat lainnya, seperti *American Sign Language* (ASL) dan Bahasa Isyarat Indonesia (BISINDO), menjadi salah satu tantangan utama dalam pengembangan sistem komunikasi berbasis teknologi bagi penyandang tunarungu. Metode berbasis (CNN), terutama model YOLOv5, telah banyak digunakan dalam penelitian untuk mendeteksi dan mengenali isyarat tangan secara real-time dengan tingkat akurasi yang cukup tinggi.

Salah satu pendekatan yang banyak digunakan dalam pengenalan bahasa isyarat adalah model berbasis deteksi objek real-time, seperti YOLOv5, yang telah terbukti efektif dalam mengekstraksi fitur dari citra tangan. Berbagai penelitian telah membuktikan bahwa model ini mampu mengenali huruf-huruf dalam bahasa isyarat dengan tingkat presisi yang tinggi. Namun, dalam beberapa kasus, model ini masih mengalami kesulitan dalam membedakan huruf-huruf yang memiliki bentuk tangan yang mirip, seperti huruf M dan N, atau huruf dengan perbedaan gerakan minimal. Oleh karena itu, pengembangan lebih lanjut, seperti integrasi dengan Attention Mechanism atau arsitektur VGG-16, dapat menjadi solusi yang memungkinkan peningkatan performa model dalam menangani kompleksitas bahasa isyarat.

Untuk memahami perkembangan terbaru dalam klasifikasi huruf bahasa isyarat menggunakan *deep learning*, tabel berikut menyajikan beberapa penelitian terkini yang telah dilakukan dalam bidang ini. Tabel ini mencakup informasi mengenai model algoritma yang digunakan, jenis huruf yang diuji, dataset yang digunakan, serta kontribusi dan hasil yang dicapai dalam setiap penelitian.

Berikut menyajikan ulasan kritis terhadap beberapa penelitian sebelumnya menggunakan berbagai algoritma untuk pengenalan bahasa isyarat, dengan fokus pada jenis huruf yang diuji, dataset digunakan, kontribusi penelitian, serta hasil diperoleh, guna membandingkan efektivitas model dalam berbagai skenario. Dapat dilihat pada tabel 2.1 sebagai berikut:

Tabel 2.1 Ulasan Krisis Tema Penelitian

No	Judul dan Nama Peneliti	Algorithm / Model	Jenis Huruf	Dataset	Pengujian Huruf	Kontribusi	Hasil
1.	*Real Time Sign Language Recognition Using Deep learning* (Sanket Bankar, Tushar Kadam, Vedant Korhale, 2023)	YOLOv5	ASL	1752 gambar	- Huruf A : O, T - Huruf M : N, T - Huruf N : M - Huruf V : W - Huruf W : K, V	Membandingkan kinerja dan waktu pengujian dengan model lain, dan menemukan bahwa model yang diusulkan lebih berhasil mengekstraksi fitur dengan gerakan tangan realtime.	Hasil akurasi 88,4%, presisi 76,6% dan recall 81,2% dengan mendeteksi objek dan gerakan tangan dari video secara *realtime*.
2.	*Sign Language Detection System Using YOLOv5 Algorithm to Promote Communication Equality People with Disabilities* (Ningsih et al., 2024)	YOLOv5	BISINDO	47 video, durasi 50 menit dengan resolusi 1280 x 720 pixel. Total dataset image generated 2388 dataset, 47 class terdiri dari huruf dan kata	-	Model tidak hanya mengklasifikasikan class huruf dan kata BISINDO tetapi juga dapat memprediksi kondisi kompleks secara konsisten.	Hasil model mencapai mAP sebesar 97,2% dan Recall sebesar 99,6%
3.	*Using YOLOv5 Algorithm to Detect and Recognize American Sign Language* (Dima & Ahmed, 2021)	*YOLOv5*	ASL	2515 gambar berwarna jarak dekat, 36 gambar berlatar hitam, terdiri dari 26 class		Model *pre-trained* hanya 167MB memori yang cukup ringan. Fps dari YOLOv5 cukup tinggi.	Penelitian mencapai 95% Precision, 97% recall, 98% F1-score, 98% map@0.5 dan 98% map@0.5=0.95 score menandai untuk mengenali

				huruf dan 10 class angka			gerakan gesture secara real time.
4.	*Alphabet Classification of Sign System Using Convolutional Neural Network with Contrast Limited Adaptive Histogram Equalization and Canny Edge Detection* (Raharjo & Sugiharti, 2023)	Arsitektur VGG-19 dan ResNet50 dikombinasikan *Contrast Limited Adaptive Histogram Equalization* (CLAHE)	ASL	26 kelas berdasarkan jumlah abjad di mana setiap kelas memiliki 3.000 gambar	-	meningkatkan kualitas detail gambar dan *Canny Edge Detection* untuk menghasilkan gambar yang fokus pada objek di dalamnya. Hasil yang difokuskan adalah nilai akurasi. Studi ini menggunakan dataset abjad ASL dari Kaggle.	Berdasarkan hasil uji, terdapat tiga hasil akurasi terbaik. Yang pertama adalah menggunakan arsitektur ResNet50, CLAHE, dan ukuran gambar 128 x 128 piksel dengan akurasi 99,9%, diikuti oleh arsitektur ResNet50, CLAHE + *Canny Edge Detectio*n, dan ukuran gambar 128 x 128 piksel dengan akurasi 99,82%, dan yang ketiga adalah arsitektur VGG-19, CLAHE, dan ukuran gambar 128 x 128 piksel dengan akurasi 98,93%.
5.	*WiASL: American Sign Language writing recognition system using commercial WiFi devices* (Qin et al., 2023)	*Attention-based Weighted Linear Discriminant Analysis (AWLDA) algorithm and spatiotemporal*	ASL	26 kelas	- 20% huruf U salah di klasifikasikan sebagai huruf V - 18 % huruf O salah diklasifikasikan sebagai huruf C 0% huruf J dan Z	WiASL memiliki dua keterbatasan (1) pengguna tetap menjaga tubuh tetap tenang dan tidak boleh ada aktivitas manusia lain di dalam ruangan, (2) tahap reset WiASL	Dalam WiASL setiap inputan 25,81 huruf per menit secara rata-rata, dan akurasi deteksi serta akurasi pengenalannya 96,16% dan 92,96%

		artificial neural network					
6.	Classification of Sign-language Using VGG16 (Abu-Naser, 2022)	*Model CNN, VGG-16 Fine-Tuned*	ASL	43500 images, 87000 images total. 29 class (A to Z, space, delete, and nothing) the	- Perbandingan performance Klasifikasi setiap huruf. Huruf E, G, H, I, Q,U,W : 99%	Klasifikasi bahasa isyarat merupakan tugas yang sangat penting dalam bidang komunikasi tunarungu di masyarakat.	Hasil tingkat akurasi dari penelitian mencapai 100% hingga dari 20 Epoch.
7.	*Real-time sign language detection: Empowering the disabled community* (Kumar et al., 2024)	VGG-16 dengan Attention	ISL (Indian Sign Language)	23 class (A, B, C, D, E, F, G, I, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Z)	- Huruf K 90% - Huruf O dan T 80% - Dan huruf lainnya 100%	Mengaktifkan pengenalan bahasa isyarat dengan akurat dengan bantuan model VGG-16 dengan Attention menggunakan mekanisme pelatihan	Hasil menunjukkan efektivitas transfer learning untuk ISL, mencapai akurasi 97,5% dengan VGG-16 dan 99.8% dengan VGG-16 plus Attention.

Penelitian yang dilakukan oleh (Sanket Bankar, Tushar Kadam, Vedant Korhale, 2023) menggunakan YOLOv5 untuk pengenalan huruf dalam American Sign Language (ASL) menunjukkan akurasi yang relatif rendah (88.4%), dengan presisi 76.6% dan recall 81.2%, yang mengindikasikan masih adanya kesalahan dalam mengklasifikasikan huruf-huruf tertentu. Secara khusus, penelitian ini mengungkap bahwa huruf M hanya dikenali dengan akurasi 0.25 dan lebih sering diklasifikasikan sebagai huruf N dan T, sementara huruf N memiliki akurasi 0.25 dan lebih sering diklasifikasikan sebagai huruf M (0.62), menunjukkan keterbatasan YOLOv5 dalam menangani huruf dengan karakteristik visual yang mirip. Hal ini sejalan dengan temuan pada eksperimen pertama penelitian ini, di mana YOLOv5 mengalami kesulitan dalam membedakan huruf tertentu dalam bahasa isyarat, yang menegaskan perlunya pendekatan berbeda selain *image detection* contohnya *keypoint detection* atau metode berbasis fitur spasial yang lebih detail untuk meningkatkan akurasi dalam tugas pengenalan huruf SIBI atau ASL.

Penelitian yang dilakukan oleh (Ningsih et al., 2024) menggunakan YOLOv5 untuk mendeteksi huruf dan kata dalam Bahasa Isyarat Indonesia (BISINDO) dengan hasil yang sangat tinggi, mencapai mAP 97.2% dan recall 99.6%, namun penelitian ini tidak membahas secara spesifik pengaruh *noise* atau kesalahan klasifikasi antar huruf dan kata dalam BISINDO. Meskipun model dapat memprediksi kondisi kompleks secara konsisten, pendekatan ini lebih berfokus pada evaluasi akurasi secara keseluruhan, tanpa menganalisis bagaimana model menangani huruf-huruf yang serupa atau sulit dibedakan. Begitupun dengan penelitian (Dima & Ahmed, 2021) dan (Raharjo & Sugiharti, 2023) tidak menjelaskan secara spesifik noise pada huruf ASL. Oleh karena itu, meskipun penelitian ini menunjukkan performa tinggi dalam klasifikasi secara luas, masih terdapat kebutuhan untuk mengkaji tingkat kesalahan model pada huruf atau kata tertentu, terutama dalam konteks penggunaan bahasa isyarat di lingkungan dunia nyata.

Penelitian oleh (Qin et al., 2023) menggunakan algoritma *Attention-based Weighted Linear Discriminant Analysis* (AWLDA) dan *spatiotemporal artificial neural network* untuk pengenalan huruf ASL, dengan akurasi mencapai 96.16% dan 92.96%, namun masih mengalami kesalahan klasifikasi pada huruf tertentu, seperti huruf U yang salah diklasifikasikan sebagai V (20%) dan huruf O yang salah dikenali sebagai C (18%). Meskipun pendekatan berbasis *attention* diterapkan, penelitian ini tidak mengkaji lebih lanjut studi kasus pada huruf yang memiliki tingkat kemiripan

tinggi, serta memiliki keterbatasan lingkungan, di mana pengguna harus tetap diam tanpa aktivitas lain dalam ruangan agar deteksi optimal. Oleh karena itu, meskipun metode *attention-based* terbukti efektif dalam mengenali huruf tertentu.

Penelitian yang dilakukan oleh (Abu-Naser, 2022) menggunakan VGG-16 yang telah di-*fine-tuned* untuk pengenalan bahasa isyarat ASL, dengan dataset besar sebanyak 87.000 gambar yang mencakup 29 kelas, menghasilkan akurasi mencapai 100% setelah 20 epoch. Model VGG-16 dan VGG-19 menunjukkan kinerja yang jauh lebih baik dibandingkan YOLOv5, terutama dalam klasifikasi huruf F, G, H, I, Q, U, dan W yang mencapai 99% akurasi, namun tidak dijelaskan bagaimana model menangani huruf yang lebih sulit dibedakan atau mengalami kesalahan identifikasi. Hasil ini mendukung temuan pada eksperimen kedua dalam penelitian ini, yang juga menggunakan VGG-16.

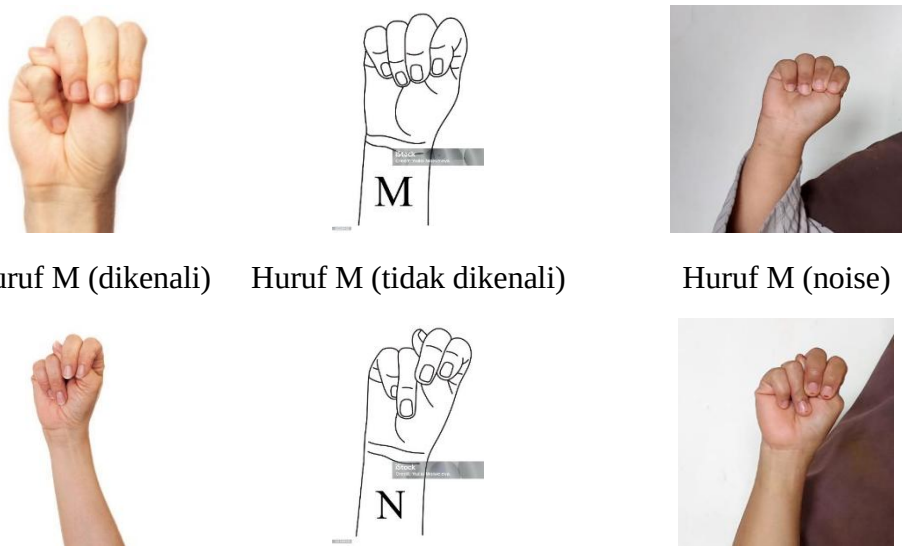
Penelitian oleh (Kumar et al., 2024) menggunakan VGG-16 *with* Attention untuk mengenali bahasa isyarat *Indian Sign Language* (ISL) menunjukkan efektivitas *transfer learning* dengan akurasi 97.5% untuk VGG-16 dan meningkat menjadi 99.8% setelah ditambahkan *Attention Mechanism*. Meskipun model VGG-16 *with* Attention berhasil meningkatkan akurasi dalam mengenali 23 kelas huruf, terdapat beberapa huruf seperti O dan T yang hanya dikenali dengan akurasi 80%, serta huruf K yang memiliki akurasi 90%, yang menunjukkan masih adanya tantangan dalam membedakan huruf-huruf yang memiliki karakteristik visual serupa. Dengan dasar pendekatan yang sama, penelitian ini dapat menjadi referensi dalam penerapan VGG-16 *with* Attention untuk pengenalan huruf dalam SIBI, dengan harapan model dapat mencapai tingkat akurasi yang serupa jika diadaptasi dengan dataset dan karakteristik bahasa isyarat yang sesuai.

Berdasarkan kajian terhadap penelitian sebelumnya, diterapkan **empat skenario eksperimen** untuk mengevaluasi efektivitas berbagai arsitektur dalam pengenalan huruf SIBI khusus (huruf M dan N) sering mengalami kesalahan klasifikasi. Eksperimen pertama: YOLOv5, bertujuan untuk menguji apakah model ini mengalami kesalahan klasifikasi akibat *noise* sebagaimana ditemukan dalam penelitian (Sanket, dkk, 2023) yang menunjukkan bahwa huruf M dan N sering tertukar. Eksperimen kedua : VGG-16, dengan fokus pada huruf M dan N karena pada penelitian (Abu-Naser, 2022) model ini tidak mengalami kendala signifikan dalam mengklasifikasikan huruf tersebut, sehingga dapat digunakan sebagai *baseline* untuk evaluasi. Eksperimen ketiga: VGG-16 *with* Attention, untuk menilai apakah model dapat meningkatkan

akurasi dalam mengenali huruf M dan N, seperti dalam penelitian (Kumar et al., 2024) menerapkan pendekatan serupa pada bahasa ISL. Eksperimen keempat menerapkan MLP sebagai model yang lebih sederhana untuk membandingkan apakah model berbasis *keypoint detection* dapat memberikan performa lebih baik dibandingkan model berbasis *image detection* dengan efisiensi lebih tinggi dan kemampuan adaptasi

2.2 Sistem Bahasa Isyarat Indonesia (SIBI)

Di Indonesia terdapat dua bahasa isyarat yang umum digunakan, yaitu Sistem Bahasa Isyarat Indonesia (SIBI) sebagai bahasa resmi dan Bahasa Isyarat Indonesia (BISINDO) sebagai bahasa tidak resmi. SIBI lebih terstruktur dan formal, sering digunakan dalam pendidikan dan penyiaran berita, sedangkan BISINDO lebih fleksibel dan ekspresif, mirip dengan penggunaan bahasa lisan sehari-hari (Indra et al., 2019). Penelitian ini lebih berfokus pada 2 huruf lebih ke gestur tangan yaitu Huruf N dan M, alasan klasifikasi huruf N dan M yaitu dalam perkenalkan nama lengkap seseorang nyaris mayoritas menggunakan nama “Nur” dan “Muhammad”. Selain itu menurut peneliti terdahulu huruf M dan N tidak dikenali.



(a) Huruf M (dikenali) Huruf M (tidak dikenali) Huruf M (noise)

(b) Huruf N (dikenali) Huruf N (dikenali) Huruf N (dikenali/noise)

Gambar 2.1 Contoh Huruf SIBI M dan N

Gambar 2.1 menggambarkan contoh huruf SIBI yang dikenali dan tidak dikenali sehingga akan mengakibatkan noise pada sistem yang dibangun. Pengembangan penelitian dalam pengenalan bahasa isyarat SIBI. Fokus penelitian pada faktor eksternal seperti pencahayaan, sudut, kamera dan penggunaan aksesoris (Nurhayati et al., 2022; Suardi, 2023; Suharjito et al., 2021). Sedangkan untuk penelitian berjalan, memadai transisi ke penelitian yang mempertimbangkan faktor intrinsik seperti variasi

ukuran tangan yang bertujuan untuk mengurangi noise dan meningkatkan akurasi sistem dengan pemanfaatan arsitektur attention dan VGG-16. Dan penelitian keberlanjutan diharapkan dapat mewujudkan sistem pengenalan huruf SIBI yang beroperasi secara *real-time* dengan menggunakan *computer vision*.

2.3 Skenario Penelitian

2.3.1 YOLOv5

Arsitektur YOLOv5 terdiri dari tiga komponen utama: *Backbone*, *Neck* (PaNet), dan *Output Head*. *Backbone* bertanggung jawab untuk mengekstraksi fitur dari gambar input menggunakan blok konvolusional seperti *Focus layer*, CSPNet (C3), dan SPP (*Spatial Pyramid Pooling*), Neck (PaNet) berfungsi untuk memperkaya representasi fitur dengan proses upsampling, concatenation, dan konvolusi tambahan, sementara Output Head menghasilkan prediksi akhir dengan deteksi *bounding box* dan klasifikasi objek, memungkinkan sistem untuk secara efisien mengenali objek dalam gambar secara *real-time*.

a. *Backbone* (Ekstraksi Fitur)

Backbone bertugas mengekstraksi fitur dari gambar masukan menggunakan serangkaian lapisan konvolusi:

1. Focus layer: Mengonversi gambar input menjadi fitur awal.

$$F(x) = Conv(x, W) + B$$

Keterangan

x : Gambar input

W : Bobot filter konvolusi

B : Bias

2. *C3-2 Block*: *Blok* residual yang menggabungkan informasi dari beberapa layer untuk menghindari hilangnya fitur penting.
3. SPP (*Spatial Pyramid Pooling*): Memperkaya representasi fitur dengan berbagai ukuran skala.
4. SELayer (*Squeeze-and-Excitation Layer*): Memperkuat fitur yang lebih penting untuk klasifikasi.

b. *Neck* (*PaNet - Path Aggregation Network*)

Bagian ini digunakan untuk meningkatkan representasi fitur dan memperbaiki deteksi objek dengan resolusi multi-skala. *Upsampling* dapat meningkatkan resolusi fitur untuk menangkap detail objek yang lebih kecil.*Concat* (*Concatenation*) dapat menggabungkan informasi dari beberapa

lapisan fiturC3 Module dapat digunakan untuk meningkatkan efisiensi ekstraksi fitur.

c. *Head* (*Output Layer*)

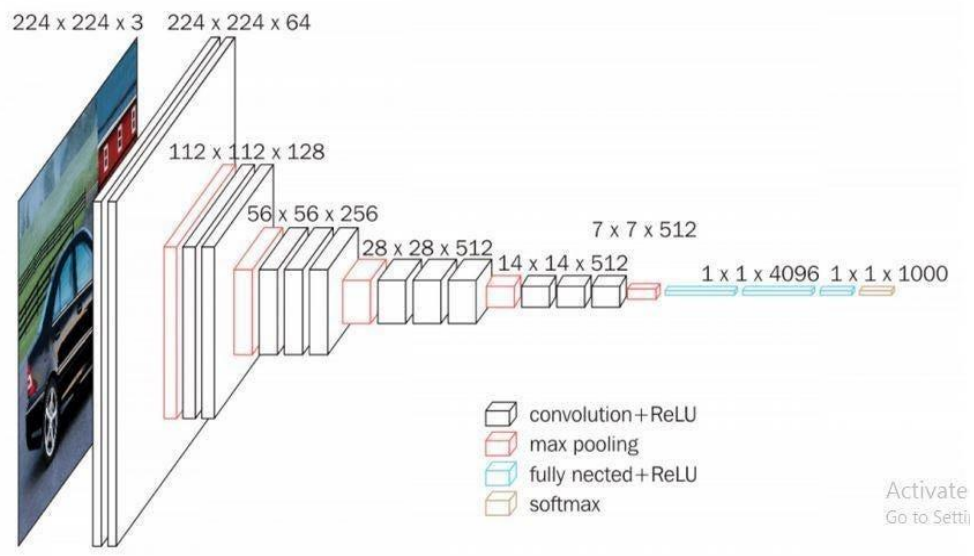
Bagian ini bertugas untuk mendeteksi objek dan memberikan koordinat *bounding box* serta skor klasifikasi. Dimana *Lapisan Conv2D* menghasilkan tiga *output* utama untuk setiap objek yang terdeteksi yaitu (1) Koordinat *Bounding box* (x, y, w, h), (2) *Confidence Score* atau seberapa yakin model terhadap deteksi objek, dan (3) *Class Probability* dimana probabilitas setiap kelas huruf yang terdeteksi.

Bounding box menggunakan persamaan :

$$x = \sigma(t_x) + C_x, y = \sigma(t_y) + C_y, w = p_{w}e^{t_w}, h = p_{h}e^{t_h}$$

Dimana σ adalah fungsi sigmoid t_x, t_y, t_w, t_hadalah parameter keluaran, dan C_x, C_y, p_w, p_h adalah nilai referensi dari *anchor box*.

2.3.2 VGG-16

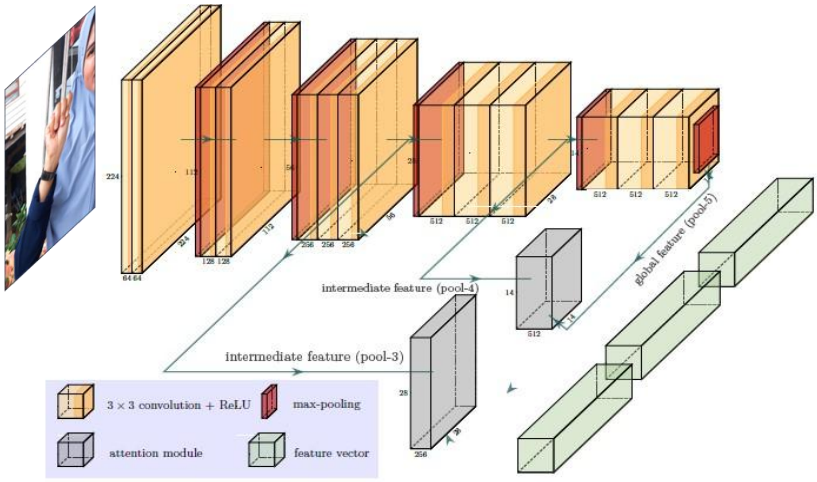


Gambar 2.2 Arsitektur dan Peta Jalan VGG-16

Arsitektur VGG-16 adalah model *convolutional neural network* (CNN) yang memiliki 16 *layers*, terdiri dari *convolutional layers* dan *fully connected layers*. Salah satu ciri utama VGG-16 adalah penggunaan *convolutional filters* berukuran 3×3 dengan arsitektur yang *deep* serta *stride* sebesar 1. *Pooling layers* menggunakan konfigurasi 2×2 dengan *stride* 2, serta menerapkan *same padding*. Ukuran *input image* yang diproses oleh *network* ini secara *default* adalah 224×224 *pixels*. Sebelum masuk ke *fully connected layers*, *feature map* berukuran 7×7 dengan 512 *channels* dikonversi menjadi *vector* dengan 25,088

channels (7×7×512) yang berfungsi sebagai *final feature representation* (Rajab et al., 2024).

2.3.3 VGG-16 *with* Attention



Gambar 2.3 Arsitektur *VGG-16 with Attention*

Attention adalah mekanisme dalam *deep learning* yang berfokus pada input data penting untuk dikerjakan seperti dalam *Natural Language Processing* (NLP) dan *Computer Vision*. Konsep ini akan meningkatkan efisiensi dan akurasi model yang lebih signifikan.

Mekanisme *attention* bekerja dengan cara menghitung bobot setiap bagian *input* dan memfokuskan model pada yang memiliki bobot lebih tinggi. Bobot ini dihitung berdasarkan relevansi fitur tertentu dengan keluaran yang diharapkan.

$$attention\ (Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right) V$$

Keterangan

Q : *Query*

K : *Key*

V : *Velue*

d_k : Dimensi dari *Key*

$softmax$: fungsi *softmax* untuk mengubah skor kesamaan antara *query* dan *key*

2.3.4 MLP (*Multi-Layer Perceptron*)

Multi-Layer Perceptron (MLP) adalah salah satu jenis jaringan saraf tiruan (*Artificial Neural Network*/ANN) yang terdiri dari lapisan *input*, lapisan tersembunyi, dan lapisan output. Setiap neuron dalam MLP terhubung sepenuhnya ke *neuron* di lapisan berikutnya (*fully connected layers*) dan

memproses data menggunakan kombinasi bobot, bias, serta fungsi aktivasi seperti *ReLU*, *Sigmoid*, atau *Tanh*. Proses pembelajaran dalam MLP dilakukan melalui *forward propagation* untuk menghasilkan *output*, lalu *backpropagation* digunakan untuk mengoptimalkan bobot berdasarkan kesalahan yang dihitung oleh fungsi *loss*.

2.4 Objek Detection

2.4.1 *Image Detection*

Image detection adalah proses dalam visi komputer yang bertujuan untuk mengidentifikasi dan menentukan lokasi objek dalam sebuah gambar dengan menggunakan teknik kecerdasan buatan, khususnya *deep learning*. Metode *image detection* bekerja dengan mendeteksi objek melalui ekstraksi fitur visual dan menerapkan model seperti *Convolutional Neural Networks* (CNN) untuk mengenali pola dalam gambar. Algoritma deteksi objek yang populer meliputi YOLO (*You Only Look Once*), Faster R-CNN, dan SSD (*Single Shot MultiBox Detector*), dapat mendeteksi dan memberi label pada berbagai objek dalam gambar secara *real-time*. Prosesnya melibatkan segmentasi gambar, pembuatan *bounding box*, serta klasifikasi untuk memastikan objek dikenali dengan akurat. Teknologi *image detection* banyak digunakan dalam berbagai aplikasi.

2.4.2 *Keypoint Detection*

Keypoint detection adalah teknik yang digunakan untuk mendeteksi titik-titik penting pada objek dalam gambar, seperti *landmark* wajah, tubuh, atau tangan. Pada konteks *MediaPipe Hand Landmark Detection*, metode *keypoint detection* digunakan untuk mendeteksi posisi dan orientasi dari tangan manusia dengan representasi 3D dari 21 titik-titik penting (*landmarks*) di tangan.

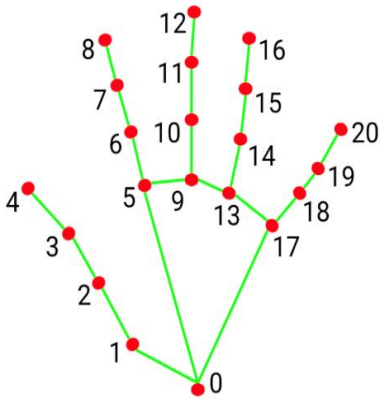
Estimasi *keypoint* dilakukan dengan melatih model yang mendeteksi posisi *landmark* p dalam bentuk (x, y, z). Dimana (x_1, y_1, z_1) dan (x_2, y_2, z_2) adalah koordinat *landmark* yang berbeda. Rumus tersebut akan membantu dalam menghitung jarak antara dua titik di tangan seperti pengenalan gerakan (*gesture recognition*).

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \quad (1)$$

Keterangan :

x dan y : koordinat dalam citra 2D

z : dimensi kedalaman (*depth*)

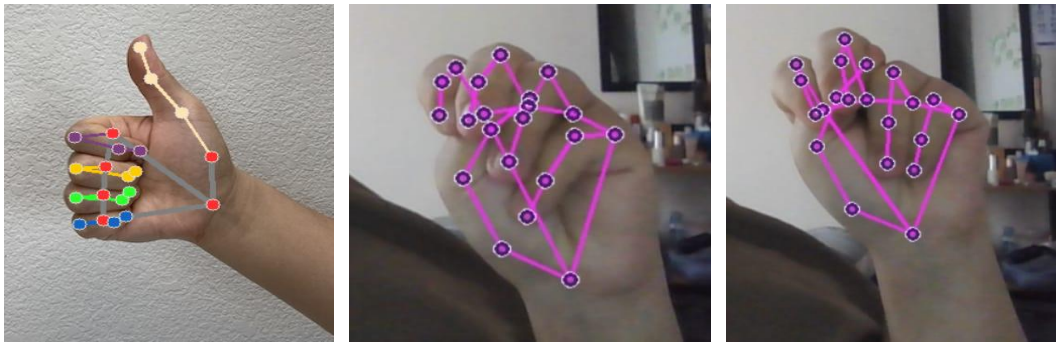


0. WRIST
1. THUMB_CMC
2. THUMB_MCP
3. THUMB_IP
4. THUMB_TIP
5. INDEX_FINGER_MCP
6. INDEX_FINGER_PIP
7. INDEX_FINGER_DIP
8. INDEX_FINGER_TIP
9. MIDDLE_FINGER_MCP
10. MIDDLE_FINGER_PIP
11. MIDDLE_FINGER_DIP
12. MIDDLE_FINGER_TIP
13. RING_FINGER_MCP
14. RING_FINGER_PIP
15. RING_FINGER_DIP
16. RING_FINGER_TIP
17. PINKY_MCP
18. PINKY_PIP
19. PINKY_DIP
20. PINKY_TIP

Gambar 2.4 Titik Koordinat *Hand Landmark* Model

Gambar 2.4 menggambarkan terdapat 21 titik acuan tangan, yang masing-masing terdiri dari koordinat x, y dan z. Koordinat x dan y dinormalisasi ke [0.0, 1.0] berdasarkan lebar dan tinggi gambar. Koordinat z tersebut mewakili kedalaman titik acuan, dengan kedalaman di pergelangan tangan sebagai titik asal. Semakin kecil nilainya, semakin dekat titik acuan tersebut dengan kamera. Besarnya z menggunakan skala yang hampir sama dengan x.

MediaPipe adalah *framework* yang memungkinkan deteksi *landmark* secara *real-time*. Algoritma *MediaPipe* dapat mengenali posisi tangan dan jari dalam sebuah citra atau video dengan tingkat akurasi yang tinggi dengan pengenalan gerakan tangan dan interaksi berbasis gerakan (*gesture recognition*).



Gambar 2.5 Contoh *MediaPipe landmark*

2.5 *Confusion Matrix*

Confusion matrix merupakan alat ukur kinerja metode prediksi dengan menghitung tingkat kebenaran dari proses klasifikasi. Akurasi diperoleh berdasarkan rasio prediksi yang benar terhadap seluruh data. Presisi diperoleh berdasarkan rasio prediksi benar positif dibandingkan dengan hasil prediksi positif keseluruhan. *Recall* diperoleh dari rasio prediksi positif yang benar dibandingkan dengan semua data yang benar.

Di dalam *confusion matrix*, ada empat kemungkinan keputusan yang mungkin terjadi dalam mendeteksi objek, seperti *True positive* (TP), *False positive* (FP), *False negative* (FN) dan *True negative* (TN). Berikut perhitungan Akurasi, Presisi dan *recall* menggunakan persamaan *confusion matrix* (1) – (3).

Tabel 2.2 *Confusion matrix*

		Actual Class	
		Positive (**P**)	Negative (**N**)
Predicted Class	Positive (**P**)	True Positive (**TP**)	False Positive (**FP**)
	Negative (**N**)	False Negative (**FN**)	True Negative (**TN**)

$$accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (2)$$

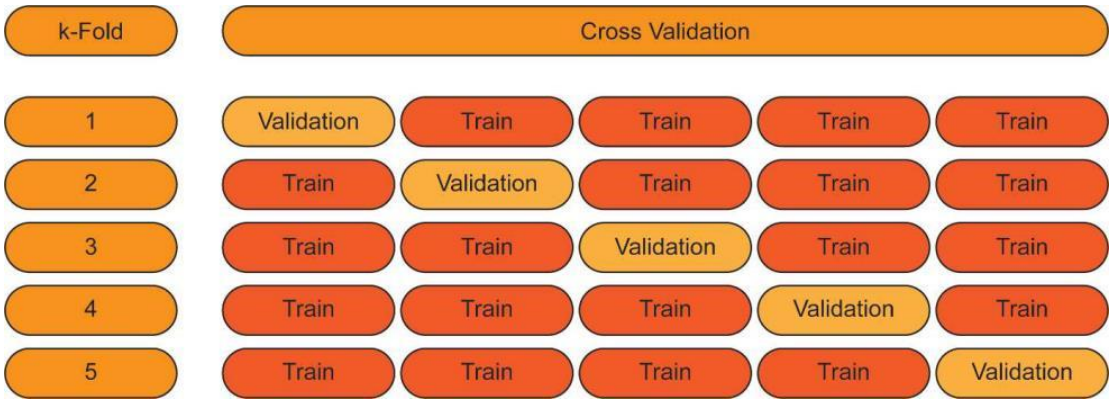
$$Precision = \frac{TP}{TP+FP} \quad (3)$$

$$recall = \frac{TP}{TP + FN} \quad (4)$$

$$F1\text{-}Score = 2 \times \frac{precision \times recall}{precision + recall} \quad (5)$$

2.6 K-Fold Cross validation

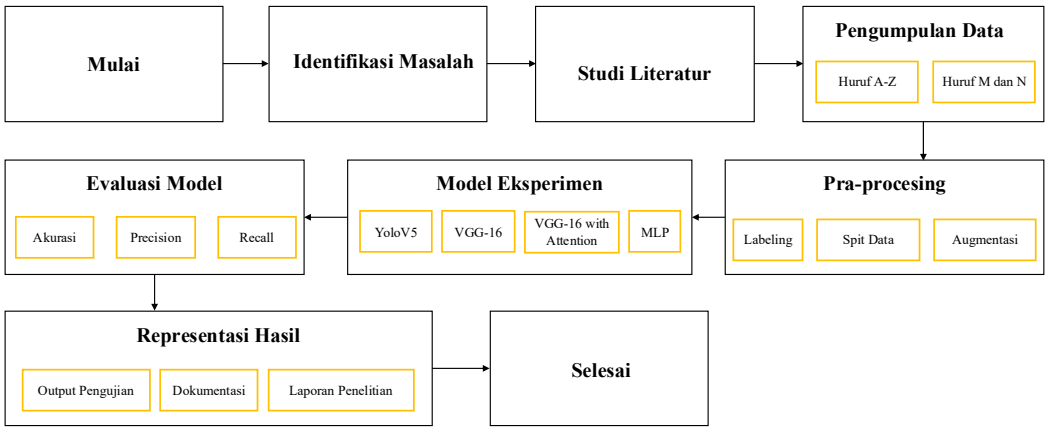
K-fold cross validation merupakan salah satu metode statistik yang diterapkan untuk mengevaluasi kinerja model atau algoritma yang telah dirancang. Pada tahap pelatihan dataset dibagi menjadi data train dan data *validation*. Model akan dilatih menggunakan data train dan divalidasi menggunakan data validation sebanyak *k-fold* kali.



Gambar 2.6 *K-fold cross validation*

BAB 3 Metodologi

3.1 Tahapan Penelitian



Gambar 3.1 Tahapan Penelitian

Gambar 3.1 menggambarkan tahapan penelitian dimulai dari identifikasi masalah, studi literatur, pengumpulan dan pra-pemrosesan data, eksperimen model (YOLOv5, VGG-16, VGG-16 *with* Attention, dan MLP), evaluasi model dengan metrik akurasi, *presisi*, dan *recall*, hingga representasi hasil melalui output pengujian, dokumentasi, dan laporan penelitian sebelum tahap penyelesaian.

3.1.1 Identifikasi Masalah

Permasalahan telah diselesaikan penelitian terdahulu dalam pengenalan Sistem Isyarat Bahasa Indonesia (SIBI), tetapi cenderung berfokus pada faktor eksternal mempengaruhi akurasi model, seperti ketidakkonsistenan pencahayaan, sudut kamera tidak ideal, dan hambatan fisik menghalangi pandangan tangan, tanpa mempertimbangkan secara mendalam aspek intrinsik dari masing-masing individu. Faktor intrinsik, seperti perbedaan ukuran, bentuk tangan, serta fleksibilitas jari, berkontribusi signifikan terhadap peningkatan *noise* dapat mengakibatkan kesalahan klasifikasi dalam model pengenalan huruf. Salah satu kasus sering terjadi adalah huruf M dan N, memiliki kemiripan gestur sehingga lebih rentan terhadap kesalahan klasifikasi. Huruf-huruf memiliki tingkat penggunaan yang tinggi dalam kehidupan sehari-hari, terutama dalam nama-nama umum digunakan di masyarakat, seperti "Muhammad" dan "Nur", sehingga penting untuk mengembangkan sistem lebih presisi dalam membedakan kedua huruf tersebut.

3.1.2 Studi Literatur

Peneliti menelusuri tinjauan pustaka untuk memberikan wawasan lebih dalam terhadap konsep-konsep yang dipakai dan dipakai dan solusi yang telah ditawarkan saat ini untuk menyelesaikan masalah yang diangkat. *MediaPipe*, *keypoint detection*, dan metode pengenalan pola berbasis *deep learning*. Studi literatur mengidentifikasi solusi telah ditawarkan dalam penelitian sebelumnya, termasuk kendala teknis seperti variabilitas ukuran tangan dan kesalahan identifikasi huruf serupa.

3.1.3 Pengumpulan Data

Pengumpulan data dilakukan dua gelombang dengan total anak muda berpartisipasi 26 orang. Pada gelombang pertama terdiri dari 14 orang anak muda dari usia 15-25 tahun. Gelombang kedua 15 orang yang berpartisipasi dengan 3 orang berpartisipasi juga di gelombang pertama. Masing-masing gambar diambil di rumah H.Subair, Suhaeda dan Hasanuddin yang memiliki dinding warna putih.

Data digunakan dalam eksperimen ini merupakan data primer, dimana data tersebut diperoleh melalui proses pengambilan gambar yang dilakukan secara mandiri. Gambar-gambar dihasilkan mencakup berbagai kondisi lingkungan, seperti variasi latar tempat : dalam ruangan (diambil di lingkungan dalam rumah berlatar polos), luar ruangan (diambil di lingkungan terbuka dengan latar alami) dan penggunaan aksesoris (cincin, gelang, jam tangan) (Nur Hayati Lilis, Nur Handayani Anik, Wahyu Sakti Gunawan Iriantoa, Rosa Andrie Asmara, Dolly Indra, 2023). **Dataset model YOLOv5** terdiri dari 1872 gambar mentah huruf-huruf SIBI yang di *augmentasi* menjadi 4492 gambar, menggunakan teknik augmentasi seperti *flip*, pengaturan kecerahan (-35% hingga +35%), dan *shear* dengan sudut ±10°(Al Ahmadi et al., 2024). Dataset ini digunakan untuk menguji kemampuan YOLOv5 dalam melakukan *object detection* pada huruf-huruf bahasa isyarat dalam berbagai kondisi lingkungan dan penggunaan aksesoris.

Dataset model VGG-16, Attention dan **MLP** terdiri dari 346 gambar mentah yang hanya berfokus pada area tangan untuk huruf M dan N dalam bahasa isyarat SIBI, tanpa *augmentasi* tambahan. Model ini menggunakan *keypoint detection* dengan *MediaPipe Landmark* untuk mendeteksi titik-titik penting pada tangan agar pengenalan gestur lebih presisi (Alvin et al., 2021),(Köpüklü et al., 2019).

3.1.4 *Pra-Processing*

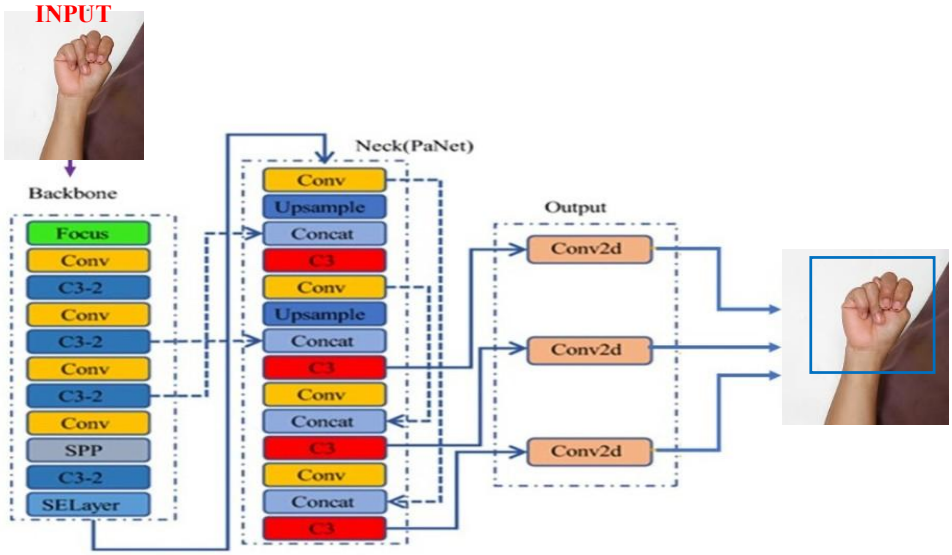
Pra-pemrosesan dalam penelitian ini dilakukan untuk memastikan bahwa data yang digunakan dalam pelatihan model memiliki kualitas yang baik dan seragam. Tahapan pertama adalah *labeling*, di mana semua gambar yang telah dikumpulkan harus diberi anotasi agar dapat digunakan dalam pelatihan model. Dalam eksperimen pertama yang menggunakan YOLOv5, pelabelan gambar dilakukan menggunakan *RoboFlow*, yang memfasilitasi proses anotasi dengan fitur-fitur tambahan seperti *preprocessing* dan *augmentasi*. Gambar yang digunakan diubah ukurannya dari 1920x1080 menjadi 640x640 menggunakan metode fit *white edge*, yang bertujuan untuk mempertahankan *rasio* aspek gambar tetap 1:1, mengurangi beban komputasi, serta menyamakan resolusi dengan *webcam* yang digunakan saat pengujian. Selanjutnya, dalam eksperimen **kedua, ketiga, dan keempat**, pelabelan dilakukan secara otomatis menggunakan *MediaPipe*, yang mengekstraksi 21 titik kunci (*landmark*) dari citra tangan untuk mempermudah proses klasifikasi. Dataset yang telah diberi label kemudian dikonversi ke dalam format *biner*, di mana setiap sampel diberi kategori berdasarkan huruf yang ditampilkan.

Setelah proses pelabelan, tahapan *split data* dilakukan untuk membagi dataset menjadi dua bagian utama: data latih (80%) dan data uji (20%). Pemisahan ini penting agar model dapat diuji dengan data yang belum pernah dilihat sebelumnya, sehingga hasil evaluasi lebih objektif dan dapat mengukur tingkat generalisasi model. Selanjutnya, dilakukan *augmentasi data* untuk meningkatkan jumlah variasi dalam dataset dan mengurangi risiko *overfitting*. Pada eksperimen YOLOv5, augmentasi dilakukan menggunakan *RoboFlow*, dengan menambah variasi citra dari 1560 gambar menjadi 4054 gambar menggunakan teknik *flip horizontal*, perubahan kecerahan dari -35% hingga +35%, serta *shear* dengan ±10° secara *horizontal* dan vertikal. Sedangkan pada eksperimen yang menggunakan *MediaPipe*, augmentasi dilakukan dengan menyesuaikan titik koordinat dari *landmark* yang diekstraksi, sehingga model dapat belajar dari variasi posisi tangan dalam berbagai kondisi pencahayaan dan sudut pengambilan gambar. Proses augmentasi ini bertujuan untuk memastikan bahwa model dapat mengenali huruf dengan lebih baik dan lebih adaptif terhadap variasi dalam lingkungan nyata.

3.1.5 Model Eksperimen

a. YOLOv5

Eksperimen pertama menggunakan model YOLOv5 (*You Only Look Once version 5*) sebagai metode deteksi huruf bahasa isyarat. Model YOLOv5 beroperasi dalam tiga tahap utama: input data, pemrosesan fitur menggunakan arsitektur YOLOv5, dan prediksi output dengan *bounding box* serta klasifikasi huruf.

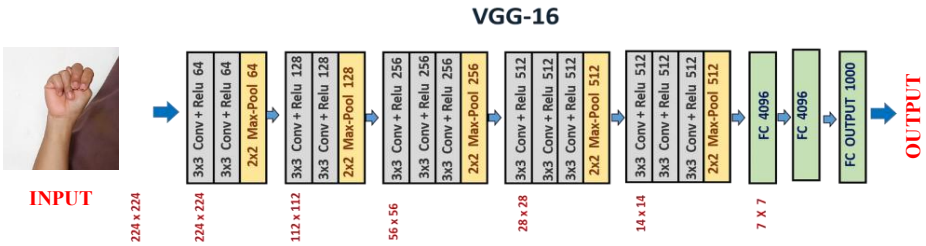


Gambar 3.2 Model Eksperimen YOLOv5

Proses dimulai dari input data, di mana gambar tangan dengan gestur huruf dikumpulkan dan diubah ukurannya menjadi 640 × 640 *piksel* untuk memastikan konsistensi dalam pelatihan model. Setelah itu, gambar melalui tahap *preprocessing* seperti normalisasi dan *augmentasi* guna meningkatkan variasi data dan mencegah *overfitting*. Setelah itu, gambar melalui tahap preprocessing seperti normalisasi dan augmentasi guna meningkatkan variasi data dan mencegah *overfitting*. Model YOLOv5 sendiri terdiri dari tiga komponen utama, yaitu *Backbone*, *Neck* (*PaNet*), dan *Head* (*Output Layer*). *Backbone* bertugas mengekstraksi fitur dari gambar menggunakan serangkaian lapisan konvolusi, termasuk *Focus layer*, *C3-2 Block*, *Spatial Pyramid Pooling* (SPP), dan *SELayer*, untuk meningkatkan representasi fitur. Kemudian, fitur yang diekstrak dikirim ke *Neck* (*PaNet*), yang memperkaya representasi fitur melalui teknik *upsampling*, *concatenation*, dan konvolusi tambahan untuk menangkap informasi dari berbagai skala objek.

Pada tahap *Head* (*Output Layer*), model menghasilkan prediksi dalam bentuk *bounding box* yang mengelilingi tangan dan menentukan kelas huruf yang terdeteksi. Prediksi ini dilakukan dengan menghitung koordinat *bounding box* (x,y,w,h), tingkat *confidence score*, dan probabilitas klasifikasi menggunakan fungsi *Softmax*. *Bounding box* diperoleh melalui persamaan transformasi non-linear berbasis fungsi *sigmoid*, yang memastikan model dapat mendeteksi posisi dan ukuran objek dengan lebih akurat. Setelah melalui proses klasifikasi, model memberikan hasil berupa *bounding box* yang mengelilingi tangan, label huruf yang terdeteksi, serta tingkat kepercayaan deteksi. Dengan pendekatan ini, YOLOv5 mampu mengenali huruf SIBI secara *real-time* dengan efisiensi tinggi, menjadikannya pilihan yang ideal untuk aplikasi berbasis visi komputer dalam bahasa isyarat.

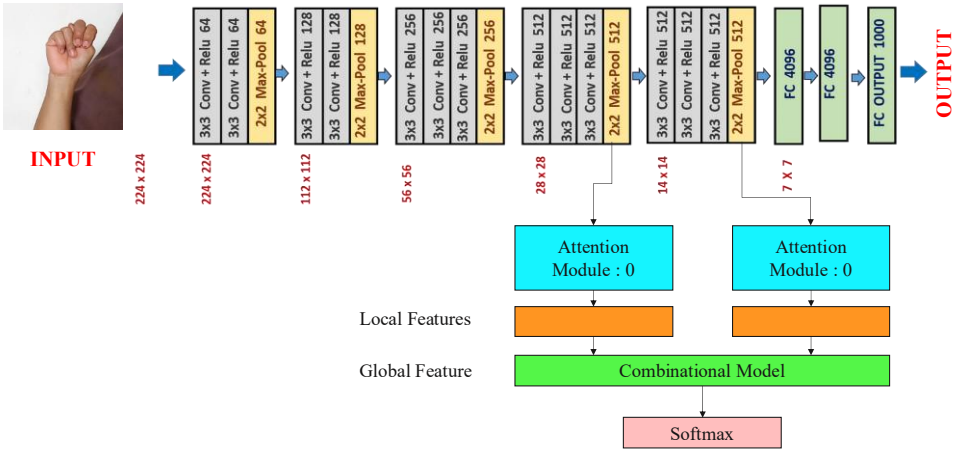
b. VGG-16



Gambar 3.3 Arsitektur VGG-16

Gambar 3.3 menggambarkan arsitektur VGG-16 (*Visual Geometry Group 16-Layers*) dengan menerima citra input berukuran 224 × 224 *piksel*, kemudian diproses melalui serangkaian lapisan konvolusi (*Conv*) 2 × 2 dengan aktivasi ReLU (*Rectified Linear Unit*). Setelah setiap beberapa lapisan konvolusi, diterapkan *max-pooling* 2 × 2 untuk mengurangi dimensi spasial dan mempertahankan informasi penting. Proses ini terjadi secara bertahap, dimulai dengan 64 *filter* konvolusi, kemudian meningkat menjadi 128, 256, dan 512 *filter* dalam lapisan lebih dalam. Setelah ekstraksi fitur melalui konvolusi bertingkat, hasilnya dilewatkan ke lapisan *fully connected* (FC) dengan 4096 *neuron*, bertugas mengubah fitur yang telah dipelajari menjadi representasi dapat digunakan untuk klasifikasi. Lapisan terakhir terdiri dari *fully connected output layer* dengan 1000 neuron. *Code* program dapat dilihat pada lampiran B : Arsitektur VGG-16

c. VGG-16 *with* Attention



Gambar 3.4 Arsitektur VGG-16 *with* Attention

Gambar 3.4 menggambarkan arsitektur VGG-16 *with* Attention, di mana citra input (224 × 224 *piksel*) diproses melalui serangkaian lapisan konvolusi 2×2 yang diikuti oleh aktivasi ReLU dan lapisan *max-pooling* 2×2 untuk ekstraksi fitur secara bertahap, dari 64 *filter* pada tahap awal hingga 512 *filter* pada tahap akhir. Model ini dibagi menjadi dua jalur utama, yaitu *Local Features* dan *Global Features*, yang masing-masing melalui *Attention Module* untuk menyoroti area penting dalam citra sebelum digabungkan dalam *Combinational* Model untuk menghasilkan fitur representasi. Hasil akhir dari kombinasi ini kemudian diteruskan ke lapisan *fully connected* (FC) dengan 4096 *neuron*, lalu diproses melalui *Softmax Layer* dengan klasifikasi *output*, memungkinkan model dapat lebih akurat mengenali pola kompleks bahasa isyarat berbasis gestur tangan.

d. MLP (*Multi-Layer Perceptron*)

Tabel 3.1 Arsitektur *Multi-Layer Perceptron*

Layer Type	***Size***	***Output Shape***	***Parameter***
Input	21 x 2 *features*	42	0
Dropout	*Rate* = 0.2	42	0
Dense + ReLU	30 *neurons*	30	1,290
Dropout	*Rate* = 0.2	30	0
Dense + ReLU	60 *neurons*	60	1,860
Dense + Softmax	2 *neurons*	2	122

Tabel 3.2 menunjukkan arsitektur MLP dengan total parameter sebesar 3.272, yang menunjukkan bahwa model ini bersifat ringan dan efisien. Model menerima input dengan 42 *fitur*, kemudian melewati *dropout* layer untuk mengurangi risiko *overfitting*, sebelum masuk ke lapisan *dense* pertama (30 *neuron*) dengan aktivasi ReLU menangkap pola *non-linear* dalam data. Setelah melalui lapisan *dropout* kedua, model dilanjutkan ke *dense layer* kedua (60 *neuron*) untuk meningkatkan kapasitas pemetaan fitur, sebelum akhirnya diklasifikasikan ke dalam 2 kelas *output* melalui *dense layer* terakhir (2 *neuron*) yang kemungkinan menggunakan aktivasi *softmax* atau *sigmoid*, tergantung pada jenis klasifikasi yang digunakan (*biner* atau *multi-class*).

3.1.6 Evaluasi Model

Evaluasi model dilakukan menggunakan *K-Fold Cross Validation* untuk menguji performa model secara lebih menyeluruh dengan membagi dataset 80:20. Dalam setiap iterasi, 1 bagian digunakan sebagai data validasi, sementara 4 bagian lainnya digunakan sebagai data pelatihan, dan proses ini diulang sebanyak 5 kali untuk memastikan bahwa setiap bagian dataset mendapatkan kesempatan sebagai data validasi maupun data pelatihan. Pendekatan membantu dalam mengurangi bias evaluasi, meningkatkan generalisasi model, serta memastikan bahwa model tidak hanya beradaptasi dengan subset tertentu dari data pelatihan.

3.1.7 Representasi Hasil

Setelah melalui proses pelatihan dan evaluasi model, hasil yang diperoleh disajikan dalam bentuk *output* pengujian, dokumentasi, dan laporan penelitian. Representasi hasil mencakup *bounding box* pada gambar yang menunjukkan lokasi tangan, klasifikasi huruf yang terdeteksi, serta matrik evaluasi seperti akurasi, *precision*, *recall*, dan *F1-score* untuk mengukur performa model. Selain itu, dilakukan analisis terhadap tingkat keberhasilan model dalam mengenali huruf dengan berbagai variasi posisi, pencahayaan, dan kondisi tangan. Dokumentasi penelitian juga mencakup perbandingan antara model yang digunakan, seperti YOLOv5, VGG-16, VGG-16 *with* Attention, dan MLP, guna melihat kelebihan dan kelemahan masing-masing pendekatan. Hasil pengujian ini kemudian dirangkum dalam laporan penelitian yang berisi analisis performa model serta rekomendasi untuk pengembangan lebih lanjut.

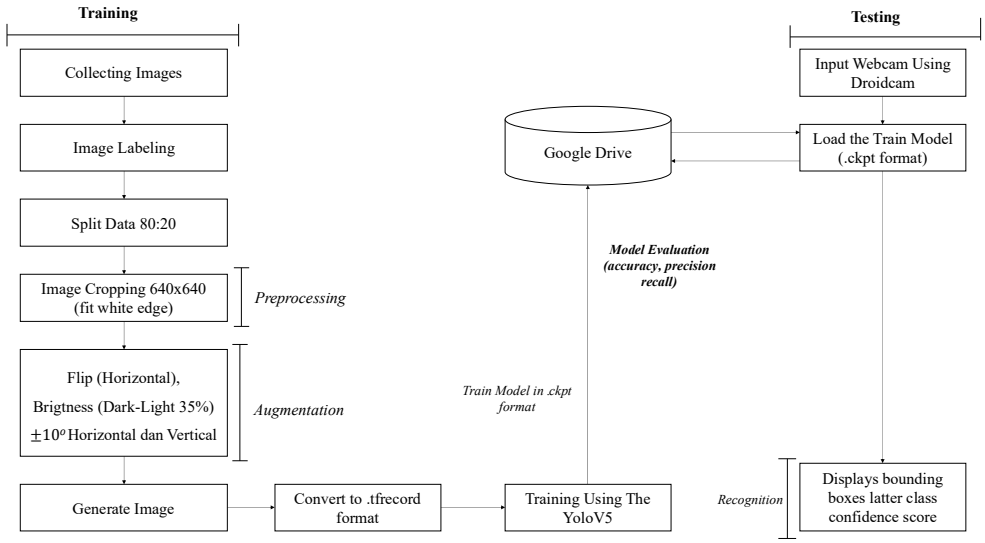
3.1.8 Kesimpulan

Berdasarkan eksperimen yang telah dilakukan, dapat disimpulkan bahwa penggunaan YOLOv5 dan arsitektur berbasis CNN mampu mengenali huruf dalam Sistem Isyarat Bahasa Indonesia (SIBI) dengan tingkat akurasi yang tinggi. Model VGG-16 dengan Attention menunjukkan peningkatan dalam membedakan huruf dengan bentuk tangan yang mirip, seperti M dan N, dibandingkan dengan model sederhana seperti MLP. Selain itu, hasil penelitian ini juga mengindikasikan bahwa augmentasi data berperan penting dalam meningkatkan generalisasi model terhadap berbagai kondisi lingkungan.

3.2 Desain Penelitian

Perancangan eksperimen dilakukan dengan empat skenario utama untuk mengevaluasi akurasi pengenalan huruf dalam bahasa isyarat SIBI menggunakan model berbasis visi komputer

3.2.1 Skenario Pertama : YOLOv5



Gambar 3.5 Diagram Alir Perancangan Images *Detection*

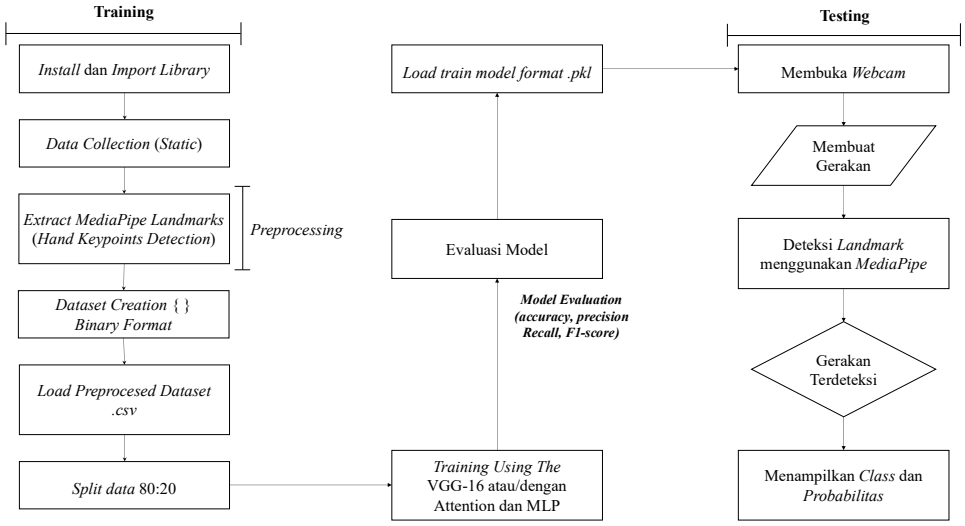
Gambar 3.6 menggambarkan perancangan model yang terdiri dari Proses pertama, *Collecting Images* yaitu mengumpulkan data citra berdasarkan 26 kelas huruf SIBI. Total keseluruhan data *citra satelit* yang dikumpulkan berjumlah 4492. Gambar diambil menggunakan kamera *handphone* dan aktor beberapa masyarakat dengan variasi ukuran tangan (17-21 cm). Posisi aktor untuk pemotretan adalah 70 cm menghadap ke depan kamera. Citra yang diambil bervariasi dari latar belakang berupa kaos hitam dan dinding bermotif abu-abu

putih, perspektif berupa tangan aktor saja dan juga dari kepala aktor hingga tubuh serta pencahayaan dalam ruangan yang redup dan terang. Kedua, *Image Labeling* yaitu tahap Pelabelan atau anotasi. Gambar yang dikumpulkan akan diberi anotasi menggunakan *RoboFlow*. *RoboFlow* memiliki beberapa fitur yang berguna untuk memudahkan proses pembuatan dataset seperti *preprocessing*, augmentasi dan salah satunya adalah pelabelan data gambar agar dapat digunakan pada tahap pelatihan. Ketiga, dilakukan *preprocessing* gambar dengan mengubah ukuran gambar dari 1920x1080 menjadi 640x640 (*fit white edge*) untuk memastikan bahwa gambar memiliki ukuran yang sama dan mengurangi daya komputasi dan membuat tepi putih dilakukan untuk memastikan gambar memiliki *rasio* yang sama 1: 1, *preprocessing* dilakukan dengan aplikasi *RoboFlow*. Alasan para peneliti memilih 640x640 adalah untuk menyamakan resolusi webcam yang akan digunakan selama pengujian. Keempat, Setelah proses *pre-processing*, penulis melakukan image augmentation menggunakan *RoboFlow* dari citra 1560 menjadi 4492 dengan memberikan variasi data yang akan digunakan oleh model sehingga dapat mengurangi *overfitting*, menambah jumlah data, mengurangi potensi kebutuhan data baru, dan tentunya memperluas cakupan data dengan banyak variasi data yang digunakan. *Augmentasi* yang digunakan adalah *flip*, kecerahan dari -35% hingga +35%, dan juga *shear* dengan ±10° *Horizontal*, ±10° *Vertical*.

Generate dataset, Proses kelima pembangkitan dilakukan dalam aplikasi *RoboFlow* dengan mengubah dataset menjadi file format ". *tfrecord*", yang merupakan format file yang digunakan oleh YOLOv5 untuk pelatihan dan evaluasi dan disimpan ke *Google Drive*. Keenam, *Training Using* YOLOv5 model yang telah dilatih menggunakan mekanisme YOLOv5 untuk memperbaiki fokus dan ekstraksi fitur dalam pengenalan citra. Model yang dilatih disimpan dalam format *.ckpt*, yang memungkinkan penggunaan ulang yang efisien untuk tahap pengujian. Selama fase pengujian, model yang telah dilatih di-*load* dari *Google Drive*, dan dengan bantuan *DroidCam*, webcam menyediakan input visual secara real-time untuk dianalisis oleh model. Model kemudian menampilkan *bounding boxes*, klasifikasi huruf, dan skor kepercayaan untuk setiap isyarat yang dikenali, memberikan umpan balik yang kritis untuk evaluasi kinerja model secara langsung dalam skenario nyata.

Catatan :

Skenario dua, tiga dan empat merupakan eksperimen yang diusulkan dengan harapan dapat mengembangkan dan meningkatkan akurasi per-huruf (huruf M dan N) dalam bahasa isyarat. Serta menentukan algoritma/ model yang tepat untuk huruf (M dan N) yang baik.



Gambar 3.6 Diagram Alir Perancangan *Keypoint Detection*

3.2.2 Skenario Kedua : VGG-16

Gambar 3.7 menggambarkan bagan alir pengenalan huruf Sistem Isyarat Bahasa Indonesia (SIBI) dengan menggunakan teknologi *MediaPipe* untuk ekstraksi fitur berupa titik-titik kunci (*landmark*) dari *citra statis*. Pada tahapan awal, dilakukan meng-*Install* dan meng-*Import* semua *Library* yang diperlukan untuk pemprosesan citra, ekstraksi *landmark* tangan dan pengembangan model. Selanjutnya, mengumpulkan data berupa citra *static* yang berisi gambar gestur tangan dari dua kategori huruf SIBI (M dan N) dengan jumlah *dataset* 346 gambar. Data citra tersebut kemudian diproses menggunakan *MediaPipe* untuk mengekstraksi 21 titik kunci pada tangan, yang diwakili oleh koordinat (x, y, z). Setelah ekstraksi selesai, dataset dibentuk dalam format *biner*, di mana setiap sampel dilabeli sesuai dengan kategorinya. Dataset ini kemudian disimpan dalam format *.csv*, dan selanjutnya dibagi menjadi data latih (80%) dan data uji (20%) untuk melatih model. Model dikembangkan menggunakan arsitektur **VGG-16** dan/atau algoritma **Attention** untuk meningkatkan akurasi klasifikasi gerakan tangan.

Tahap pengujian dilakukan dengan membuat model yang telah dilatih sebelumnya, yang disimpan dalam format *.pkl*, dan digunakan untuk mendeteksi gerakan tangan secara *real-time* melalui *webcam*. Subjek membuat gerakan tangan di depan *webcam*, yang kemudian dideteksi oleh *MediaPipe* untuk mengekstraksi titik kunci tangan. Apabila gerakan terdeteksi, model akan memproses hasil ekstraksi tersebut untuk melakukan klasifikasi berdasarkan pelatihan yang telah dilakukan. Hasil klasifikasi ditampilkan dalam bentuk kelas gerakan tangan (M_SIBI atau N_SIBI) beserta probabilitas keakuratan nya. Evaluasi kinerja model dilakukan dengan menggunakan metrik seperti akurasi, presisi, recall dan *F1-score* untuk memastikan bahwa sistem memiliki performa yang optimal dalam mengenali dan mengklasifikasikan gerakan tangan.

3.2.3 Skenario Ketiga : VGG-16 *with* Attention

Skenario ketiga menggabungkan model VGG16 dengan mekanisme Attention untuk meningkatkan akurasi pengenalan. *Mekanisme Attention* dirancang untuk membantu model fokus pada area tertentu dari gambar yang paling penting untuk klasifikasi, seperti posisi jari dan telapak tangan. Dengan pendekatan ini, diharapkan masalah yang dihadapi dalam skenario sebelumnya, seperti kesulitan mengenali huruf dengan akurasi rendah, dapat teratasi. *Mekanisme Attention* memungkinkan model untuk lebih adaptif terhadap variasi ukuran tangan dan kondisi lingkungan yang berbeda pada akhirnya meningkatkan kinerja sistem secara keseluruhan dalam pengenalan bahasa isyarat SIBI.

3.2.4 Skenario Keempat : MLP

Skenario keempat menggunakan *Multi-Layer* sebagai arsitektur yang lebih sederhana untuk membandingkan efektivitasnya dengan model lebih kompleks dalam tugas pengenalan huruf SIBI (M dan N). Model MLP menerima 42 fitur *input*, kemudian diproses melalui dua lapisan *fully connected* (*dense layers*) dengan 30 dan 60 *neuron*, masing-masing diaktifkan menggunakan ReLU (*Rectified Linear Unit*) untuk menangkap pola *non-linear* dalam data. Meningkatkan generalisasi dan mengurangi *overfitting*, model menerapkan *dropout layer* sebelum akhirnya diklasifikasikan ke dalam dua kelas *output* (huruf M dan N) menggunakan *dense layer* terakhir dengan 2 *neuron*, yang kemungkinan besar menggunakan aktivasi softmax atau sigmoid sesuai dengan kebutuhan klasifikasi *biner*.

3.3 Metode Penelitian

3.3.1 Waktu dan Lokasi Penelitian

Pengumpulan data dilaksanakan rentang waktu Desember 2023 hingga Juni 2024, guna memastikan keterwakilan variasi ukuran tangan yang diperoleh. Penelitian berpusat di Desa Benteng Gantarang (diantaranya: Desa Borong Loe dan Desa Borong Nangka) yang secara administratif terletak di Kabupaten Bulukumba, Sulawesi Selatan, Indonesia. Pemilihan lokasi didasarkan pada faktor keberagaman partisipan anak muda. Tujuan memilih keberagaman jenis ukuran tangan, sehingga memungkinkan akuisisi data lebih autentik dan kontekstual..

3.3.2 Sumber Data

Proses akuisisi data dilakukan secara bertahap dalam dua gelombang pengambilan gambar untuk memastikan variasi dalam dataset. Gelombang pertama diambil pada bulan Desember 2023-Januari 2024 difokuskan pada huruf M dan N, menghasilkan sebanyak 199 gambar yang dikumpulkan dari partisipan dengan rentang ukuran tangan tertentu. Gelombang kedua, bulan Februari-Juni 2024 mencakup keseluruhan 26 huruf A-Z dalam SIBI, menghasilkan 1.673 gambar yang diperagakan oleh individu berbeda untuk meningkatkan keragaman data.

3.3.3 Pemilihan Sampel

Partisipan penelitian ini tidak hanya berasal dari Desa Benteng Gantarang, mencakup dua desa di dalamnya, yaitu Borong Loe Tengah dan Borong Nangka. Keterlibatan lebih dari satu lokasi bertujuan untuk memperkaya variasi dataset serta memastikan bahwa model dikembangkan memiliki *robustness* dalam menghadapi perbedaan karakteristik gestur dan lingkungan pencahayaan.

Selain itu, eksperimen dirancang dengan mempertimbangkan faktor gangguan visual berupa penggunaan aksesori tangan, seperti cincin, jam tangan, dan sarung tangan, guna menguji keandalan model mengidentifikasi huruf meskipun terdapat elemen *non-biometrik* dapat mengaburkan pola tangan. Pengambilan gambar dilakukan dalam dua kondisi utama, yakni di luar ruangan dengan pencahayaan alami dan di dalam ruangan dengan pencahayaan buatan, untuk mengevaluasi seberapa sensitif model terhadap perubahan lingkungan eksternal.

3.3.4 Skenario Eksperimen

Skenario eksperimen dirancang dalam empat tahap untuk mengevaluasi performa berbagai arsitektur model dalam mengenali huruf SIBI. **Eksperimen pertama**: YOLOv5, berfokus pada deteksi objek untuk mengidentifikasi apakah terbukti terdapat penurunan akurasi pada beberapa huruf SIBI?. **Eksperimen kedua**: VGG-16, diharapkan dapat memberikan peningkatan akurasi dalam pengenalan huruf M dan N dengan pendekatan berbasis *keypoint detection*. **Eksperimen ketiga**: VGG-16 *with* Attention untuk melihat apakah dapat meningkatkan ketajaman model dalam mengenali karakteristik spesifik huruf M dan N?. Terakhir, eksperimen keempat membandingkan arsitektur yang lebih kompleks dengan model yang lebih sederhana, yaitu *Multi-Layer Perceptron* (MLP), untuk menilai efektivitas dalam pengenalan huruf dengan menggunakan *koordinat keypoint* tanpa ekstraksi fitur yang kompleks. Keempat skenario ini bertujuan untuk menemukan pendekatan yang paling optimal, efisien, dan adaptif dalam sistem pengenalan bahasa isyarat berbasis *deep learning*.

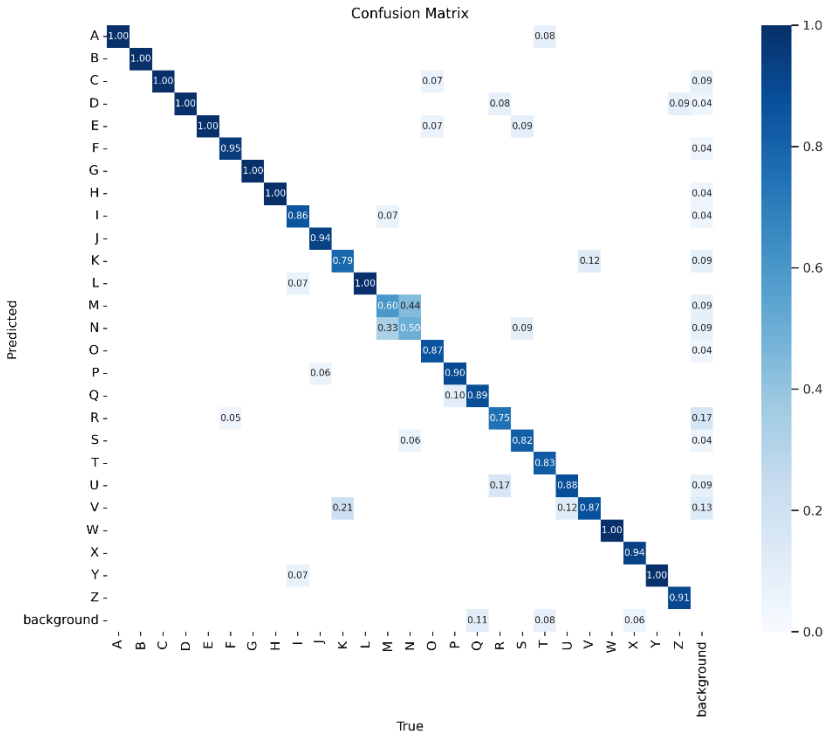
3.3.5 Teknik Augmentasi

Teknik augmentasi diterapkan pada Arsitektur YOLOv5 mencakup *flipping,* penyesuaian kecerahan (-35% hingga +35%), serta *shearing* dengan sudut ±10°, bertujuan untuk meningkatkan ketahanan model terhadap variasi kondisi pencahayaan dan sudut pandang. Eksperimen pertama difokuskan pada *object detection*. Sementara itu, model VGG-16, Attention, dan MLP menggunakan pendekatan *keypoint detection*, dengan dataset yang diformat dalam .csv dan mencakup koordinat *MediaPipe Landmark,* sehingga memungkinkan analisis yang lebih terperinci terhadap pergerakan jari dan struktur gestur isyarat.

BAB 4 Hasil dan Pembahasan

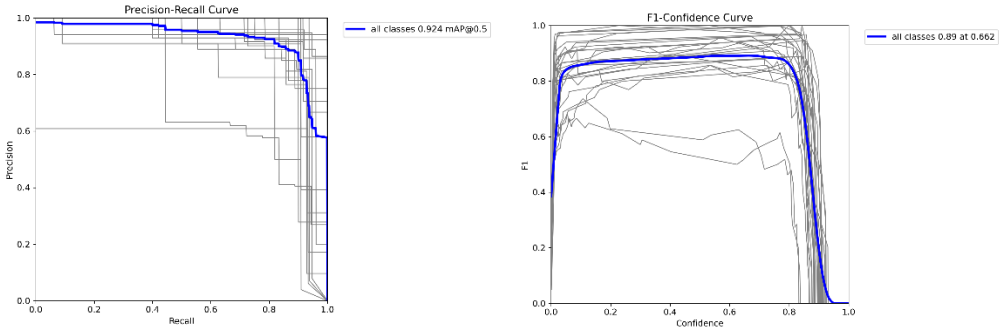
4.1 Hasil Skenario Penelitian

4.1.1 Hasil Eksperimen YOLOv5



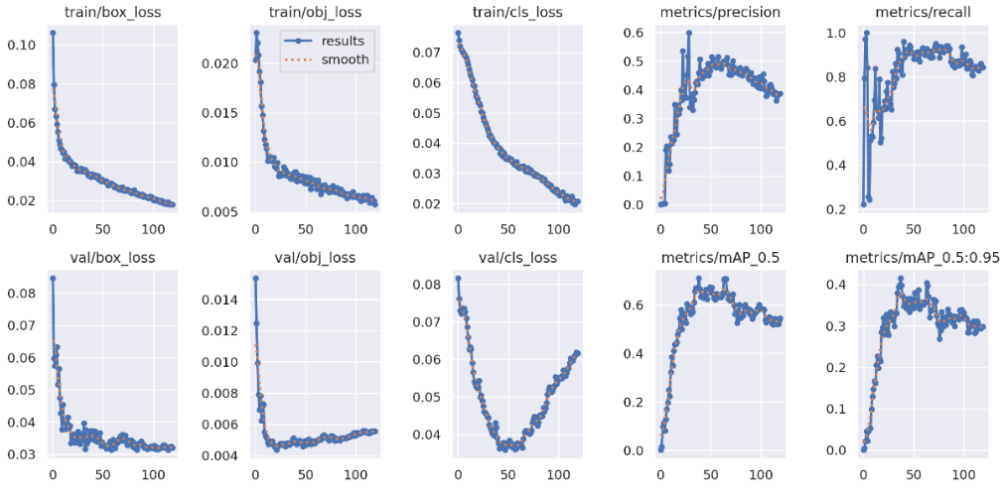
Gambar 4.1 Hasil *Confusion Matrix*

Pada Gambar 4.1, dapat dilihat bahwa pengenalan setiap huruf untuk setiap data uji memiliki hasil yang cukup baik. Beberapa kali terjadi kesalahan pengenalan pada huruf M, N, K dan Q. Kesalahan pengenalan tersebut terjadi karena fitur yang dibentuk oleh masing-masing huruf memiliki tingkat kemiripan yang sangat tinggi. Kesalahan pengenalan tersebut terjadi karena fitur yang dibentuk oleh masing-masing huruf memiliki tingkat kemiripan yang sangat tinggi seperti A, C, E, G, M, N, S and Q menggunakan objek detection (Bagaskoro et al., 2023). Beberapa huruf yang di dikenali sangat baik dengan menggunakan *keypoint detection* yaitu huruf A, B, C, D, E, F, G, H, I, J, K, L, O, P, R, S, T, U, V, W, X, Y dan Z. Terbukti dengan Tabel 2 dengan tingkat precision, accuracy dan recall yang besar, sehingga hasil dari skenario pengujian yang telah diujikan dapat menghasilkan tingkat akurasi sebesar 85,46% yang didapat dari metode YOLOv5.



Gambar 4.2 Hasil *Precision - Recall* dan *F1-Score*

Pada Gambar 4.2 menggambarkan kurva *Precision-Recall* dan *F1-Score* menunjukkan kinerja model dalam membedakan huruf berdasarkan data uji digunakan. Kurva *Precision-Recall* tetap tinggi di berbagai ambang batas mengindikasikan bahwa model mampu mempertahankan performa baik untuk kebanyakan huruf, dengan sedikit penurunan pada huruf-huruf tertentu memiliki fitur serupa.



Gambar 3.7 *Loss Function danPerformance Metrics*

Tabel 4.1 Hasil Skenario Pertama YOLOv5

No	YOLOv5 Model			
	Skenario	***Akurasi***	***Presisi***	***Recall***
1.	Outdoor Testing *With*out Accessories	0.954	0.912	1
2.	Outdoor Testing *With* Accessories (Ring)	0.889	0.831	0.947
3.	Outdoor Testing *With* Accessories (Watch)	0.948	0.901	1
4.	Outdoor Testing *With* Accessories (Gloves)	0.957	0.930	1
5.	Indoor Testing *With*out Accessories	0.957	0.917	1
6.	Indoor Testing *With* Accessories (Ring)	0.959	0.928	0.987
7.	Indoor Testing *With* Accessories (Watch)	0.933	0.875	1
8.	Indoor Testing *With* Accessories (Gloves)	0.924	0.853	0.965

Berdasarkan Tabel 4.1 menunjukkan hasil dari pengujian model YOLOv5 menunjukkan variasi akurasi, presisi, dan recall di berbagai skenario dalam dan luar ruangan, dengan atau tanpa menggunakan aksesoris seperti cincin dan jam tangan. Secara keseluruhan, akurasi tertinggi diperoleh pada pengujian di dalam ruangan tanpa aksesoris, dengan akurasi 0,957, sedangkan akurasi terendah tercatat pada pengujian di luar ruangan dengan menggunakan cincin, dengan akurasi 0,889.



Gambar 4.3 Hasil dari *Bounding Box*

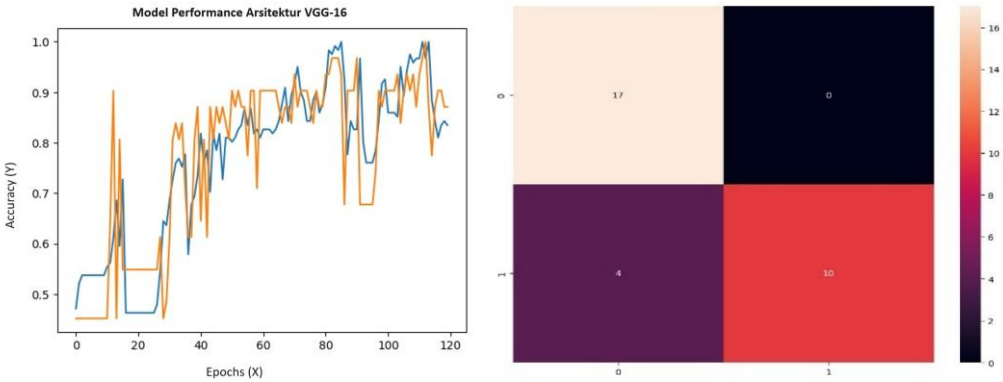
Gambar 4.3 menggambarkan hasil *bounding box* dari model YOLOv5 dalam mendeteksi huruf SIBI pada berbagai individu dengan latar belakang berbeda. Model mampu mengenali huruf dengan cukup baik pada sebagian besar kasus, namun terdapat beberapa deteksi kurang presisi terutama pada kondisi pencahayaan beragam. Hal ini menunjukkan bahwa meskipun model dapat bekerja dengan baik dalam lingkungan terkendali, masih diperlukan optimasi lebih lanjut agar model lebih adaptif, pencahayaan dan latar belakang bervariasi.

4.1.2 Hasil Eksperimen VGG-16

Tabel 4.2 Hasil Skenario Kedua VGG-16

Label	**VGG16**				
	Class	**Akurasi**	**Presisi**	**Recall**	**F1-Score**
0	Huruf M	0.87	0.81	1.00	0.89
1	Huruf N	0.87	1.00	0.71	0.83

Hasil pada Tabel 3 menunjukkan bahwa huruf M memiliki performa lebih stabil dengan recall 1.00, sementara huruf N menunjukkan performa lebih rendah dengan *recall* 0.71, meskipun memiliki presisi tinggi (1.00). Performa ini menunjukkan model lebih cenderung mengenali huruf M dengan lebih baik dibandingkan huruf N.



Gambar 4.4 Model Evaluasi dan *Confusion Matrix* Model VGG-16

Gambar 4.4 menggambarkan kksperimen berhasil mengklasifikasi huruf SIBI M dan N menggunakan algoritma VGG-16 dengan akurasi sekitar 88,89%. Grafik menunjukkan stabilitas pembelajaran, dan *confusion matrix* mencatat 17 prediksi benar untuk M dan 13 untuk N, meski ada 4 kesalahan untuk M. Hasil ini membuktikan efektivitas model dalam mengenali gestur, meskipun terbatas oleh ukuran dataset yang kecil (346 gambar). Ke depan, penelitian dapat diperluas dengan dataset lebih besar, augmentasi data, dan *mekanisme attention* untuk meningkatkan akurasi dan generalisasi model.



Gambar 4.5 Pengujian Dalam Ruangan (Normal dan Sarung Tangan)

Penelitian (Abu-Naser, 2022) tingkat akurasi mencapai 100% hingga 20 epoch dengan menggunakan objek deteksi sehingga, model VGG-16 dapat memprediksi dan mengklasifikasi huruf bahasa isyarat tanpa ada kesalahan. Jika dibandingkan dengan hasil saat ini (akurasi 87%) menggunakan *keypoint detection*, penelitian ini memiliki potensi peningkatan performa dengan menerapkan metode *transfer learning* tambahan, seperti data augmentasi yang lebih luas atau optimasi arsitektur.

Tabel 4.3 *Cross Validation* K=5: VGG-16

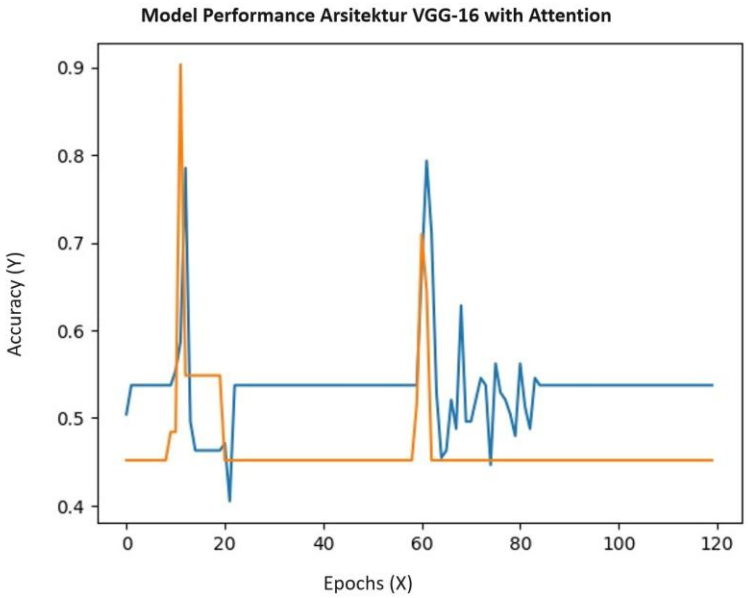
Cross Validation	***Precision***	***Recall***	***F1-Score***
Iterasi 1	0.63	0.58	0.58
Iterasi 2	0.77	0.77	0.77
Iterasi 3	1	1	1

Iterasi 4	1	1	1
Iterasi 5	0.77	0.77	0.77

Tabel 4.2 menunjukkan hasil evaluasi model VGG-16 menggunakan metode *K-Fold Cross Validation* (K=5), dengan metrik *Precision, Recall*, dan *F1-Score* setiap iterasi. Hasil menunjukkan bahwa performa model meningkat secara signifikan pada iterasi ke-3 dan ke-4 dengan nilai *Precision, Recall*, dan *F1-Score* mencapai 1, sementara iterasi lainnya memiliki fluktuasi yang lebih rendah, yang mengindikasikan bahwa model masih memiliki variabilitas dalam performanya tergantung pada subset data yang digunakan dalam validasi. Tetapi hasil akurasi 100% dengan *loss* yang sangat kecil, sebagaimana terlihat dari *plot training loss*, mengindikasikan kemungkinan *overfitting* terhadap data pelatihan.

4.1.3 Hasil Eksperimen VGG-16 *with* Attention

Hasil dari pengujian algoritma VGG-16 *with* Attention mengalami **underfitting**, yang disebabkan kurangnya variasi dataset atau huruf M dan N ini tidak memerlukan arsitektur yang kompleks. Beberapa penelitian (Kumar et al., 2024) performa *transfer learning* untuk klasifikasi ISL (*Indian Sign Language*) menggunakan VGG-16 *with* Attention menunjukkan mencapai akurasi 97,5% dengan VGG-16 dan 99,8%. Huruf M dan N di deteksi 100% dikenali karena gerakan ISL dengan SIBI berbeda, sehingga dapat dibedakan dengan huruf-huruf yang lain.



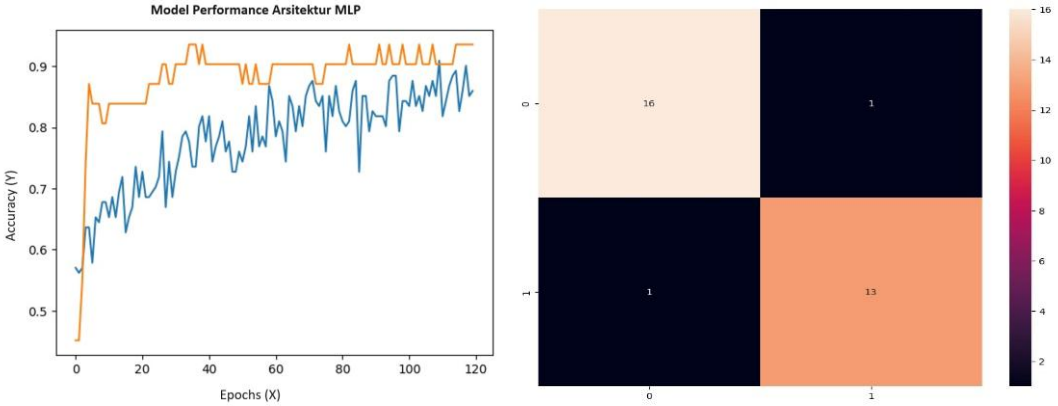
Gambar 4.6 Model Evaluation Model VGG-16 *with* Attention

4.1.4 Hasil Eksperimen MLP

Tabel 4.4 Hasil Eksperimen Keempat MLP

Label	MLP				
	Class	**Akurasi**	**Presisi**	**Recall**	**F1-Score**
0	Huruf M	0.94	0.94	0.94	0.94
1	Huruf N	0.94	0.93	0.93	0.93

Penelitian MLP menunjukkan bahwa metode dengan *keypoint detection* mencapai akurasi 94%, lebih tinggi dibandingkan penelitian (Bagaskoro et al., 2023) yang menggunakan *neural networks* (82.30%) dan MLP (84.54%). Hal ini membuktikan bahwa *keypoint detection* mampu meningkatkan akurasi pengenalan huruf dengan lebih baik, terutama dalam mengatasi kesalahan akibat kemiripan fitur antar huruf.



Gambar 4.7 *Confusion Matrix* Model MLP

Tabel 4.5 *Cross Validation* K=5: MLP

Crossvalidation	***Precision***	***Recall***	***F1-Score***
Iterasi 1	0.94	0.94	0.93
Iterasi 2	0.93	0.93	0.94
Iterasi 3	0.94	0.94	0.93
Iterasi 4	0.93	0.93	0.94
Iterasi 5	0.94	0.94	0.94

Tabel 4.5 menunjukkan *hasil cross-validation* dengan K=5 pada model MLP menunjukkan bahwa nilai *precision, recall*, dan *F1-score* relatif stabil di setiap *iterasi*, dengan kisaran *precision* antara 0.93 hingga 0.94, *recall* antara 0.93 hingga 0.94, dan *F1-score* antara 0.93 hingga 0.94, yang mengindikasikan bahwa model memiliki konsistensi performa dalam klasifikasi data. Konsistensi nilai metrik bahwa model MLP tidak mengalami variabilitas signifikan antara *iterasi*, sehingga dapat disimpulkan bahwa model memiliki stabilitas yang baik dalam proses validasi silang.

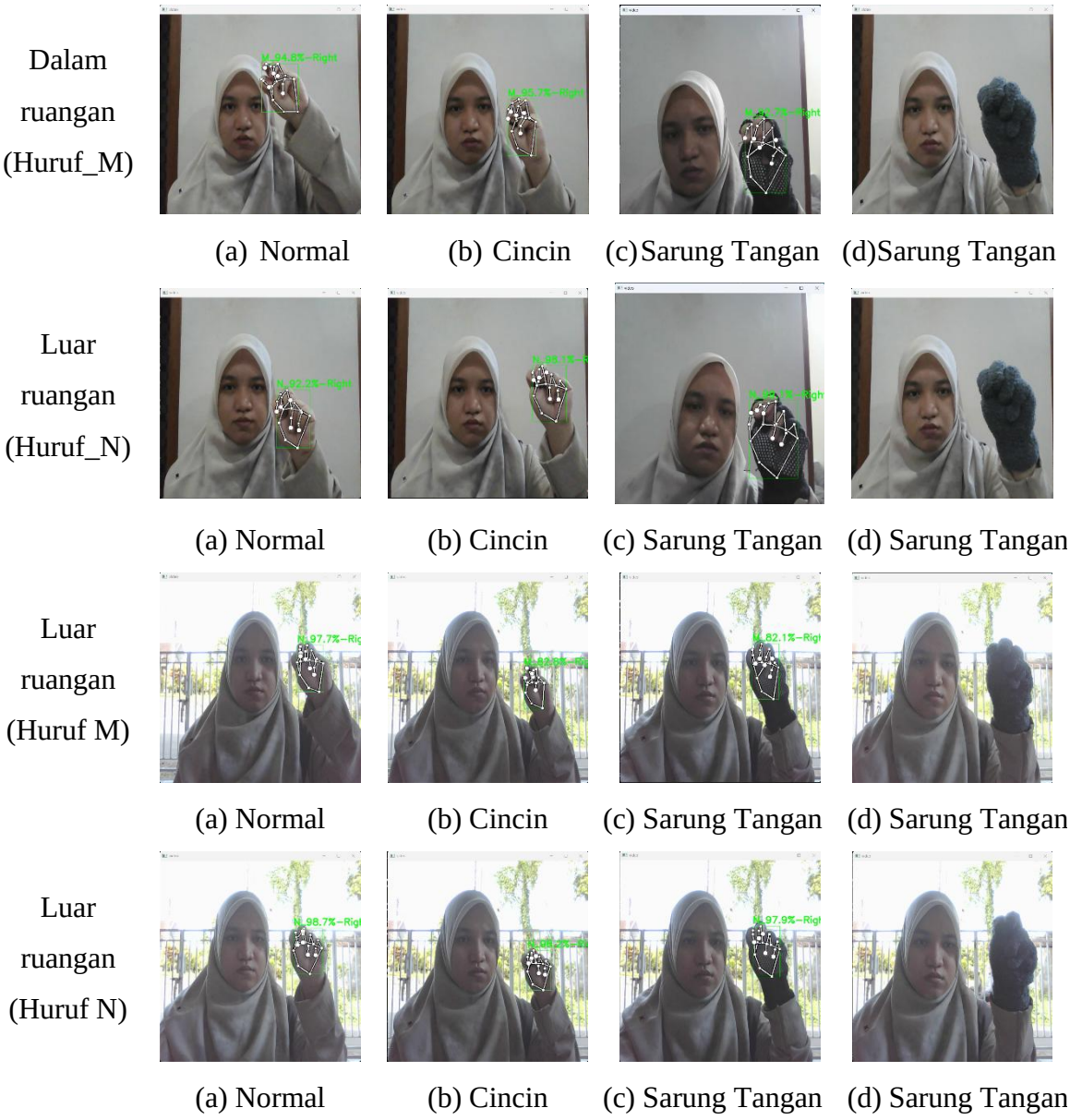
4.2 Pembahasan

Pengaruh faktor performa sistem pengenalan huruf SIBI dilakukan evaluasi berdasarkan empat kategori utama: faktor lingkungan (layak diimplementasikan secara *real-time*), literatur, eksternal dan intrinsik. Masing-masing skenario eksperimen dianalisis berdasarkan kemampuan mengenali huruf khusus M dan N dengan berbagai skenario misalnya pencahayaan, aksesoris dan ukuran tangan. Hasil evaluasi dapat dilihat pada Tabel 4.6 dengan kondisi pengujian secara real-time sebagai berikut:

Tabel 4.6 Perbandingan Performa Model Pengenalan Huruf SIBI (Huruf M dan N)

Permasalahan	**Keterangan**	**YOLO v5**	**VGG-16**	**VGG-16 *with* Attention**	**MLP**
Faktor Lingkungan	Mengenali huruf SIBI diatas akurasi 90%	Tidak	Hampir	Tidak	Iya
Faktor Literatur	Menggunakan *object detection*	*Image detection*	*Keypoint detection*	*Keypoint detection*	*Keypoint detection*
Faktor Eksternal	*Outdoor Testing Without Accessories*	0.60	0.81	0.53	0.97
	Outdoor Testing With Accessories (Ring)	0.50	0.80	0.53	0.94
	Outdoor Testing With Accessories (Watch)	0.55	0.72	0.52	0.95
	Outdoor Testing With Accessories (Gloves)	0.47	0.70	0.45	0.67
	Indoor Testing Without Accessories	0.62	0.82	0.53	0.98
	Indoor Testing With Accessories (Ring)	0.50	0.80	0.54	0.94
	Indoor Testing With Accessories (Watch)	0.57	0.69	0.50	0.90
	Indoor Testing With Accessories (Gloves)	0.49	0.80	0.30	0.68
Faktor Intrinsik	Ukuran tangan pendek 17 cm	53%	78%	57%	98%

Berdasarkan Tabel 4.6, hasil dari pengujian model YOLOv5 menunjukkan variasi akurasi, *presisi*, dan *recall* di berbagai skenario dalam dan luar ruangan, dengan atau tanpa menggunakan aksesoris seperti cincin dan jam tangan. Secara keseluruhan, akurasi tertinggi diperoleh pada pengujian di dalam ruangan tanpa aksesoris, dengan akurasi 0,957, sedangkan akurasi terendah tercatat pada pengujian di luar ruangan dengan menggunakan cincin, dengan akurasi 0,889. Hasil ini berbeda pada tabel 3 yang hanya khusus dua huruf (M dan N), sedangkan hasil tabel 4 terdiri 26 huruf (A-Z).



Gambar 4.8 Kondisi Pengujian M dan N

Gambar 4.8 menunjukkan variasi kondisi pengujian huruf M dan N pada Sistem Isyarat Bahasa Indonesia (SIBI) baik di dalam maupun luar ruangan, dengan empat skenario: normal, memakai cincin, sarung tangan 80%, dan sarung tangan 100%. Setiap variasi aksesori dan pencahayaan bertujuan untuk mengevaluasi ketahanan model terhadap *noise* visual yang umum terjadi dalam lingkungan nyata. Dari pengamatan visual ini, dapat dilihat bahwa kondisi aksesori seperti sarung tangan 100% secara signifikan mengaburkan struktur jari, berpotensi menurunkan akurasi pengenalan gestur oleh model berbasis citra penuh atau tidak mendeteksi huruf apapun.

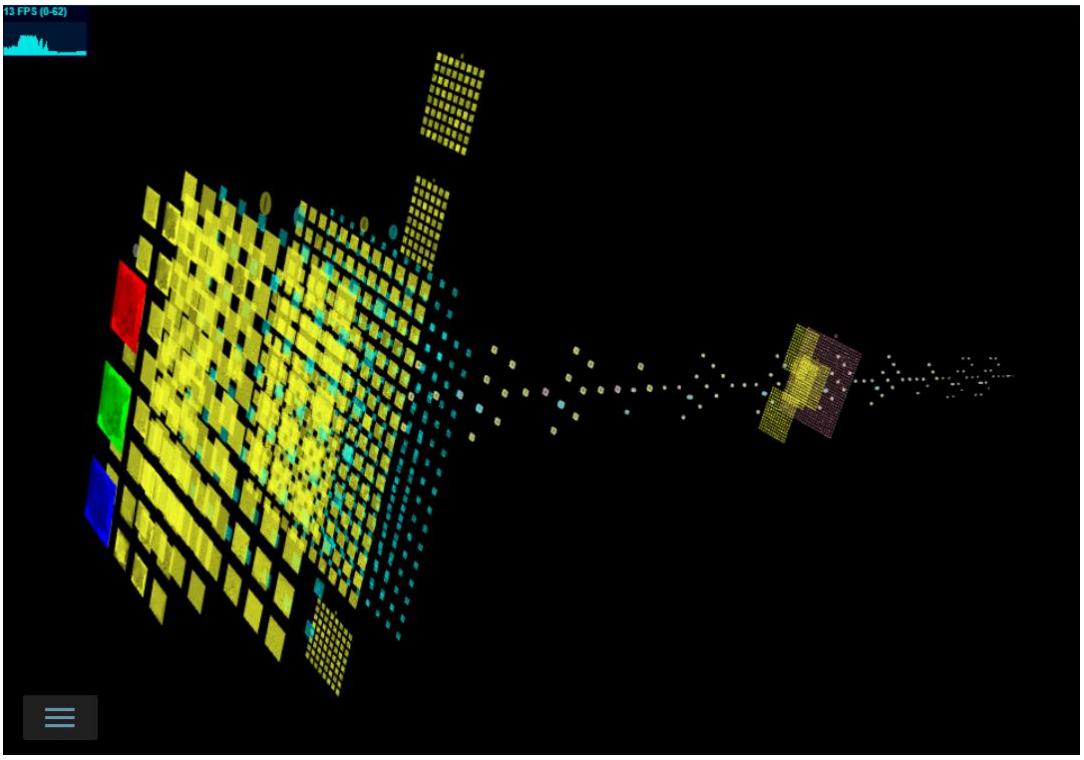
Meskipun VGG-16 *with* Attention menawarkan kapasitas tinggi untuk mengekstraksi fitur dari citra penuh, pendekatan ini menjadi kurang efektif ketika

digunakan pada input numerik berukuran kecil seperti 21 titik koordinat tangan. Kompleksitas jaringan tidak sebanding dengan jumlah fitur dan volume data, sehingga model kesulitan dalam mempelajari representasi secara optimal. Hal ini ditunjukkan oleh akurasi validasi yang rendah (53.72%) dan nilai loss yang tinggi (0.6906), mengindikasikan gejala *underfitting*. Sebaliknya, MLP dengan dua layer mampu menyesuaikan kompleksitas model dengan fitur spasial yang tersedia, sehingga menghasilkan klasifikasi lebih presisi pada huruf-huruf SIBI dengan gestur serupa.

Tabel 4.7 Perbandingan Arsitektur

No	**Komponen**	**YOLOv5**	**VGG-16**	**VGG-16 *with* Attention**	**MLP**
1.	Input	Citra 640x 640	42 *fiture* (21 *landmark* x,y)	42 *fiture* (21 *landmark* x,y)	42 *fiture* (21 *landmark* x,y)
2.	Lapisan awal	*Conv+Polling*	*Dense +ReLU*	Dense +Attention	*Danse layer* (30 *nodes*)
3.	Lapisan tengah	*CSP Bottleneck + Detect*	*Dense + Flatten*	*Flatten + Dense*	*Dense layer (*60 *nodes)*
4.	Output	26 *classes* (A-Z)	2 *Classes* (M/N)	2 *Classes* (M/N)	2 Classes (M/N)
5.	Jumlah parameter	>7M	138K	140K	3.2K
6.	Ketahanan terhadap noise	Rendah	Sedang	Rendah	Tinggi
7.	Akurasi huruf M	44%	87.10%	53.72%	94%
8.	Akurasi huruf N	50%	87.10%	53.72%	93%
9.	Masalah umum	*Noise* & latar belakang dominan	Masih kompleks untuk input kecil	Underfitting (fitur terlalu serupa)	Cocok untuk input numerik dan variasi ukuran tangan

Tabel 4.7 menunjukkan kompleksitas arsitektur yang tinggi pada YOLOv5 dan VGG-16 *with* Attention tidak memberikan keunggulan akurasi dalam klasifikasi huruf M dan N, justru menunjukkan performa lebih rendah dibandingkan MLP memiliki struktur sederhana dan efisien untuk data numerik. Hasil ini menegaskan bahwa pemilihan arsitektur harus disesuaikan dengan karakteristik *input*, di mana representasi spasial lebih optimal diproses oleh model ringan seperti MLP daripada arsitektur konvolusional kompleks.



Gambar 4.9 Contoh Visualisasi Parameter

Gambar 4.9 menggambarkan visualisasi parameter dari setiap model arsitektur seperti YOLOv5 dan VGG-16 *with* Attention memiliki kompleksitas tinggi dengan jumlah parameter yang mencapai lebih dari 7 juta dan 140 ribu masing-masing, karena dirancang untuk memproses citra resolusi penuh (640×640 piksel), mengandung 409,600 nilai *piksel*. Dalam proses ini, setiap lapisan konvolusi dan *Attention* akan mengeksekusi ratusan ribu fitur yang tidak hanya berasal dari tangan pengguna, tetapi juga dari latar belakang, pencahayaan, dan aksesori. Hal ini menyebabkan distorsi representasi terhadap gestur yang memiliki konfigurasi mirip seperti huruf M dan N, apalagi jika gambar mengandung noise atau pencahayaan ekstrem, sehingga model gagal membedakan fitur utama dan terjadi misclassification (akurasinya turun hingga 0.44 dan 0.50).

Sebaliknya, model MLP hanya menggunakan 42 fitur numerik dari 21 titik *landmark* (x,y) yang diekstraksi oleh *MediaPipe*, tanpa pengaruh visual eksternal. Dengan total parameter hanya sekitar 3.200, MLP beroperasi pada ruang fitur yang lebih sempit, namun lebih terstruktur dan relevan secara spasial, sehingga mampu mengidentifikasi pola koordinat gestur huruf M dan N dengan akurasi mencapai 94% dan 93%. Visualisasi perbandingan ini secara logis membuktikan bahwa arsitektur sederhana dengan input numerik langsung dari struktur anatomi tangan lebih efektif dibandingkan model visual yang padat namun rawan gangguan.

Selama proses eksperimen, penelitian ini menemukan bahwa kompleksitas arsitektur tidak selalu berbanding lurus dengan peningkatan akurasi, khususnya dalam kasus pengenalan huruf SIBI yang memiliki konfigurasi gestur sangat mirip seperti M dan N. Temuan menunjukkan bahwa model MLP, meskipun sederhana, mampu mengungguli model kompleks seperti YOLOv5 dan VGG-16 *with* Attention dalam kondisi pengujian nyata yang melibatkan noise visual, variasi ukuran tangan, serta penggunaan aksesori. Selain itu, integrasi *keypoint detection* terbukti lebih efisien dalam merepresentasikan fitur spasial tangan, mengurangi pengaruh latar belakang, dan mempercepat proses klasifikasi secara real-time, menjadikan pendekatan ini relevan untuk sistem pengenalan bahasa isyarat yang responsif dan adaptif terhadap lingkungan pengguna.

Hasil penelitian ini memberikan kontribusi penting baik dalam ranah akademik maupun sosial. Secara akademik, penelitian ini memperluas pemahaman terhadap efektivitas arsitektur sederhana seperti MLP dalam mengolah input numerik berbasis *keypoint detection*, terutama untuk kasus pengenalan gestur dengan konfigurasi serupa seperti huruf M dan N dalam sistem SIBI. Kontribusi ini membuka ruang bagi pengembangan model ringan yang efisien untuk diterapkan dalam studi lanjut mengenai pengenalan bahasa isyarat. Secara sosial, sistem yang diusulkan berpotensi besar untuk meningkatkan aksesibilitas komunikasi bagi penyandang disabilitas, khususnya dalam konteks penggunaan sehari-hari di lingkungan kerja dan pendidikan. Dengan kemampuan real-time dan ketahanan terhadap gangguan visual, model ini dapat diintegrasikan dalam perangkat portabel seperti smartphone, sehingga memperluas jangkauan teknologi bantu yang inklusif dan adaptif terhadap kebutuhan komunitas disabilitas di Indonesia.

BAB 5 Kesimpulan dan Saran

5.1 Kesimpulan

Kesimpulan penelitian diperoleh beberapa temuan penting terkait dengan penelitian klasifikasi pengenalan huruf Sistem Isyarat Bahasa Indonesia (SIBI) menggunakan Algoritma Attention dan VGG-16 sebagai berikut:

1. Performa metode *deep learning* dalam mengenali huruf-huruf pada analisis performa eksperimen menunjukkan bahwa tidak semua huruf dalam Sistem Isyarat Bahasa Indonesia (SIBI) dapat dikenali dengan baik menggunakan arsitektur *deep learning* yang kompleks seperti kombinasi arsitektur VGG-16 *with* Attention, terutama huruf M dan N memiliki kemiripan fitur tinggi sehingga sering terjadi kesalahan prediksi.
2. Eksperimen membuktikan bahwa VGG-16 mampu meningkatkan akurasi 0,8710, tetapi tidak dengan kombinasi VGG-16 *with* Attention mengalami *underfitting* dengan akurasi hanya 0,5372. Sementara pendekatan lebih sederhana seperti MLP dengan metode *keypoint detection* melalui *MediaPipe* mampu meningkatkan akurasi pengenalan huruf M dan N hingga 94%. Oleh karena itu, pemilihan arsitektur harus mempertimbangkan karakteristik fitur setiap huruf, di mana huruf dengan fitur unik dapat dikenali dengan baik menggunakan arsitektur kompleks seperti huruf huruf A, B, C, D, E, F, G, H, I, J, K, L, O, P, Q, R, S, T, U, V, W, X, Y, sementara huruf (M dan N) memiliki kemiripan tinggi lebih optimal menggunakan model sederhana seperti arsitektur MLP. Dengan demikian, strategi terbaik dalam sistem pengenalan huruf SIBI adalah menggabungkan berbagai pendekatan arsitektur sesuai dengan karakteristik setiap huruf untuk mencapai akurasi optimal.

Dapat disimpulkan bahwa menyempurnakan performa arsitektur sederhana dalam sistem pengenalan huruf pada Sistem Isyarat Bahasa Indonesia (SIBI) dengan memanfaatkan deteksi titik kunci (*keypoint detection*).

5.2 Saran

Berdasarkan kesimpulan diatas, ada beberapa saran yang dapat diberikan untuk penelitian selanjutnya, yaitu:

1. Penggunaan dataset yang lebih besar dan beragam dengan variasi ukuran tangan, kondisi pencahayaan, serta perangkat kamera untuk meningkatkan generalisasi model.
2. Pengembangan model yang dioptimalkan untuk aplikasi mobile atau embedded systems dapat meningkatkan implementasi praktis dalam komunikasi bahasa isyarat secara langsung.
3. mengintegrasikan pendekatan berbasis transformer atau hybrid *deep learning* guna meningkatkan akurasi pengenalan huruf yang lebih kompleks.

Daftar Pustaka

Abu-Naser, T. N. A.-J. S. S. (2022). Classification of Sign language Using VGG16. *ISAS 2023 - 7th International Symposium on Innovative Approaches in Smart Technologies, Proceedings*, *6*(6), 36–46. doi: 10.1109/ISAS60782.2023.10391673

Al Ahmadi, S., Mohammad, F., & Al Dawsari, H. (2024). Efficient YOLO-Based Deep Learning Model for Arabic Sign Language Recognition. *Journal of Disability Research*, *3*(4). doi: 10.57197/jdr-2024-0051

Alayed, A. (2024). Machine Learning and Deep Learning Approaches for Arabic Sign Language Recognition: A Decade Systematic Literature Review. *Sensors*, *24*(23). doi: 10.3390/s24237798

Alvin, A., Shabrina, N. H., Ryo, A., & Christian, E. (2021). Hand Gesture Detection for Sign Language using Neural Network *with MediaPipe*. *Ultima Computing : Jurnal Sistem Komputer*, *13*(2), 57–62. doi: 10.31937/sk.v13i2.2109

Bagaskoro, M. C., Prasojo, F., Handayani, A. N., Hitipeuw, E., Wibawa, A. P., & Liang, Y. W. (2023). Hand image reading approach method to Indonesian Language Signing System (SIBI) using neural network and multi layer perseptron. *Science in Information Technology Letters*, *4*(2), 97–108. doi: 10.31763/sitech.v4i2.1362

Cindy Mutia Annur. (2023). *Proporsi Penduduk Bekerja dengan Disabilitas Menurut Status Pekerjaan di Indonesia (2021-2022)*. Databoks. https://databoks.katadata.co.id/datapublish/2023/08/08/mayoritas-pekerja-disabilitas-di-indonesia-berstatus-wirausaha

Dima, T. F., & Ahmed, M. E. (2021). Using YOLOv5 Algorithm to Detect and Recognize American Sign Language. *2021 International Conference on Information Technology, ICIT 2021 - Proceedings*, 603–607. doi: 10.1109/ICIT52682.2021.9491672

Ewe, E. L. R., Lee, C. P., Lim, K. M., Kwek, L. C., & Alqahtani, A. (2024). LAVRF: Sign language recognition via Lightweight Attentive VGG16 *with* Random Forest. *PLoS ONE*, *19*(4 April), 1–22. doi: 10.1371/journal.pone.0298699

Halder, A., & Tayade, A. (2021). Real-time Vernacular Sign Language Recognition using *MediaPipe* and Machine Learning. *International Journal of Research Publication and Reviews*, *2*(5), 9–17. doi: 10.13140/RG.2.2.32364.03203

Indra, D., Purnawansyah, Madenda, S., & Wibowo, E. P. (2019). Indonesian sign language

recognition based on shape of hand gesture. *Procedia Computer Science*, *161*, 74–81. doi: 10.1016/j.procs.2019.11.101

Konten Redaksi Rakyatku.Com. (2019). *Mahasiswa UMI Buat Aplikasi Pembasiindo Untuk Belajar Bahasa Isyarat*. Konten Redaksi Rakyatku.Com. https://rakyatku.com/read/155065/mahasiswa-umi-buat-aplikasi-pembasiindo-untuk-belajar-bahasa-isyarat

Köpüklü, O., Gunduz, A., Kose, N., & Rigoll, G. (2019). Real-time hand gesture detection and classification using convolutional neural networks. *Proceedings - 14th IEEE International Conference on Automatic Face and Gesture Recognition, FG 2019*. doi: 10.1109/FG.2019.8756576

Kumar, S., Rani, R., & Chaudhari, U. (2024). Real-time sign language detection: Empowering the disabled community. *MethodsX*, *13*(August), 102901. doi: 10.1016/j.mex.2024.102901

M. J. AL Moustafa, A., Mohd Rahim, M. S., Khattab, M. M., Zeki, A. M., Matter, S. S., Soliman, A. M., & Ahmed, A. M. (2024). Arabic Sign Language Recognition Systems: a Systematic Review. *Indian Journal of Computer Science and Engineering*, *15*(1), 1–18. doi: 10.21817/indjcse/2023/v15i1/241501008

Ningsih, M. R., Nurriski, Y. J., Az, F., Sanjani, Z., Faris, M., Hakim, A., Unjung, J., & Muslim, A. (2024). Sign Language Detection System Using YOLOv5 Algorithm to Promote Communication Equality People *with* Disabilities. *Scientific Journal of Informatics*, *11*(2), 549–558. doi: 10.15294/sji.v11i2.6007

Nur Hayati Lilis, Nur Handayani Anik, Wahyu Sakti Gunawan Iriantoa, Rosa Andrie Asmara, Dolly Indra, M. F. (2023). Classifying BISINDO Alphabet using Tensorflow Object Detection API. *Jurnal ILKOM*, *15*(2), 358–364.

Nurhayati, O. D., Eridani, D., & Tsalavin, M. H. (2022). Sistem Isyarat Bahasa Indonesia (SIBI) Metode Convolutional Neural Network Sequential secara Real Time. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, *9*(4), 819. doi: 10.25126/jtiik.2022944787

Qin, Y., Pan, S., Zhou, W., Pan, D., & Li, Z. (2023). WiASL: American Sign Language writing recognition system using commercial WiFi devices. *Measurement: Journal of the International Measurement Confederation*, *218* (June). doi: 10.1016/j.measurement.2023.113125

Raharjo, A. S. G., & Sugiharti, E. (2023). Alphabet Classification of Sign System Using Convolutional Neural Network *with* Contrast Limited Adaptive Histogram Equalization and Canny Edge Detection. *Scientific Journal of Informatics*, *10*(3), 239–

250. doi: 10.15294/sji.v10i3.44137

Rajab, M. A., Abdullatif, F. A., & Sutikno, T. (2024). Classification of grapevine leaves images using VGG-16 and VGG-19 deep learning nets. *Telkomnika (Telecommunication Computing Electronics and Control)*, *22*(2), 445–453. doi: 10.12928/TELKOMNIKA.v22i2.25840

Sajeena, A., Sheeba, O., & Ajitha, S. S. (2020). Indian sign language recognition using YOLOV5. *AIP Conference Proceedings*, *2222*, 107–113. doi: 10.1063/5.0005665

Sanket Bankar, Tushar Kadam, Vedant Korhale, M. A. A. K. (2023). Real Time Sign Language Recognition Using Deep Learning. *International Journal for Research in Applied Science and Engineering Technology*, *11*(9), 193–199. doi: 10.22214/ijraset.2023.55621

Sevli, O., & Kemaloglu, N. (2020). Turkish Sign Language digits classification *with* CNN using different optimizers. *International Advanced Researches and Engineering Journal*, *4*(3), 200–207. doi: 10.35860/iarej.700564

Suardi, C. (2023). CNN architecture based on VGG16 model for SIBI sign language. *ETLTC-ICETM2023 INTERNATIONAL CONFERENCE PROCEEDINGS: ICT Integration in Technical Education & Entertainment Technologies and Management*, *2909*(120010). doi: 10.1063/5.0181956

Suharjito, Thiracitta, N., & Gunawan, H. (2021). SIBI Sign Language Recognition Using Convolutional Neural Network Combined *with* Transfer Learning and non-trainable Parameters. *Procedia Computer Science*, *179*(2019), 72–80. doi: 10.1016/j.procs.2020.12.011

Tazyeen Fathima, Ashif Alam, Ashish Gangwar, Dev Kumar Khetan, & Prof. Ramya K. (2024). Real-Time Sign Language Recognition and Translation Using Deep Learning Techniques. *International Research Journal on Advanced Engineering Hub (IRJAEH)*, *2*(02), 93–97. doi: 10.47392/irjaeh.2024.0018

Umar, A. A. Z., Satma, S., Mukarramah, R., Agung, R. D., Amir, N. H., Indra, D., & Hayati, L. N. (2019). Pembasiindo (Pembelajaran Bahasa Isyarat Indonesia) Besbasis Android Di Sekolah Luar Biasa Negeri 02 Makassar. *Jurnal Balireso: Jurnal Pengabdian Pada Masyarakat*, *4*(2), 135–144. doi: 10.33096/balireso.v4i2.124

Yinatan, A. Z. (2023). *Menilik Distribusi Sektor Pekerja Disabilitas Indonesia*. Data Goodstats. https://data.goodstats.id/statistic/agneszefanyayonatan/menilik-distribusi-sektor-pekerja-disabilitas-indonesia-TJbE3

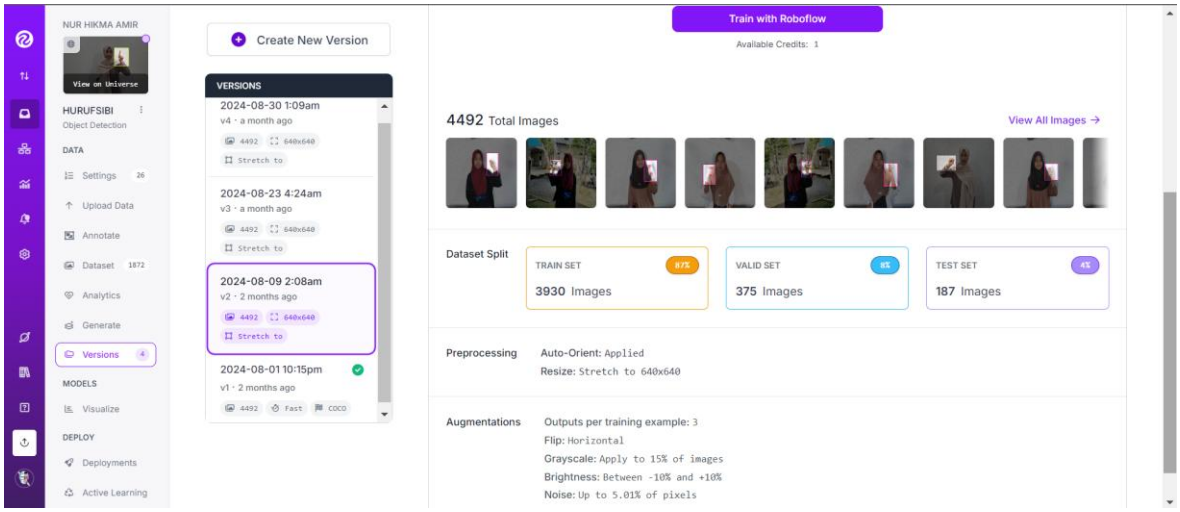
Lampiran A

Lampiran 1. Pengumpulan Dataset

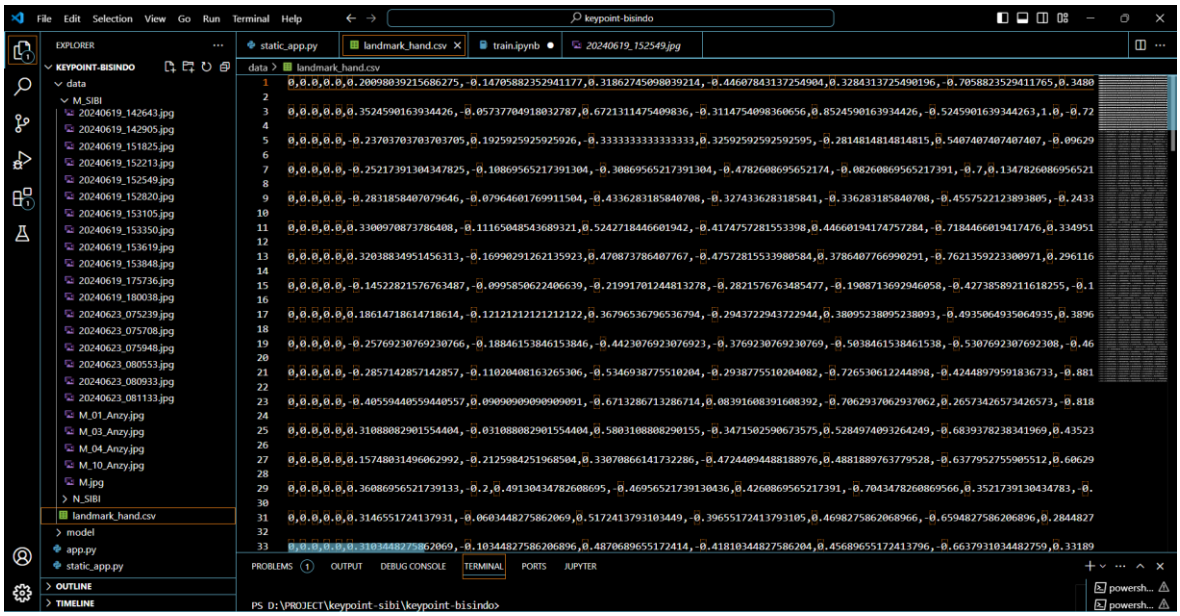
No	Contoh Jenis Dataset	Gambar
1.	M Dalam Ruangan Polos	
2.	M Dalam Ruangan Pakai Cincin	
3.	M Dalam Ruangan Pakai Jam Tangan	
4.	M Dalam Ruangan Sarung Tangan	
5.	M Luar Ruangan Polos	
6.	M Luar Ruangan Pakai Cincin	

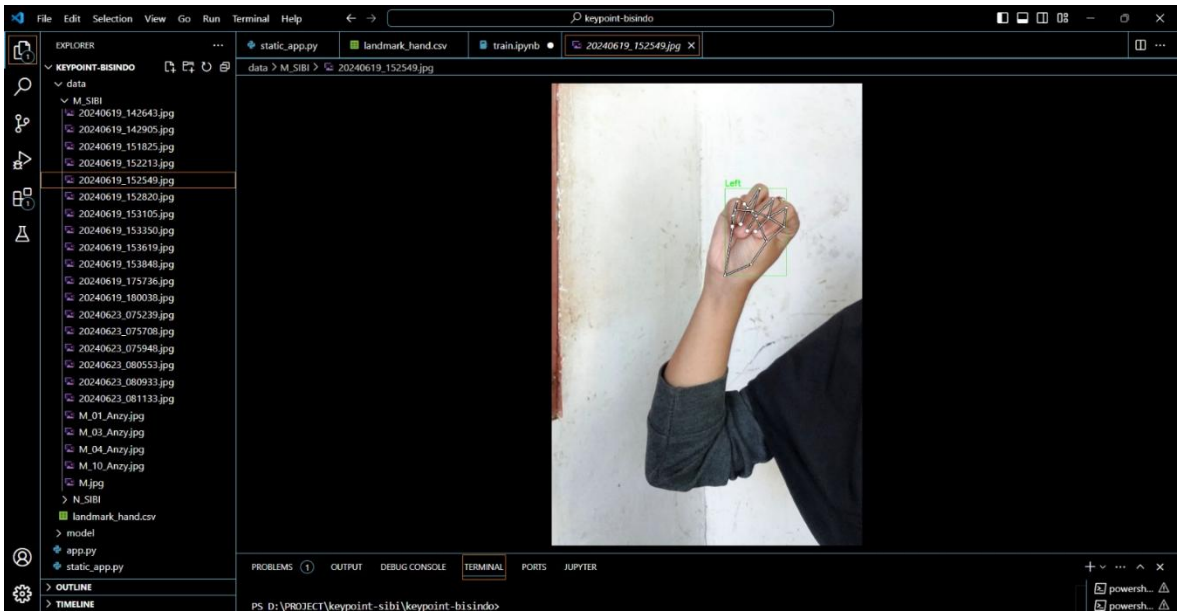
7.	M Luar Ruangan Pakai Jam Tangan	
8.	M Luar Ruangan Sarung Tangan	
9.	N Dalam Ruangan Polos	
10.	N Dalam Ruangan Pakai Cincin	
11.	N Dalam Ruangan Pakai Jam Tangan	
12.	N Dalam Ruangan Sarung Tangan	
13.	N Luar Ruangan Polos	

14.	N Luar Ruangan Pakai Cincin	
15.	N Luar Ruangan Pakai Jam Tangan	
16.	N Luar Ruangan Sarung Tangan	



Gambar diatas menunjukkan proses pelatihan model dengan total 4.492 gambar yang dibagi menjadi tiga set, yaitu 3.930 gambar untuk pelatihan (*train set*), 375 gambar untuk validasi (*valid set*), dan 187 gambar untuk pengujian (*test set*). Proses augmentasi gambar mencakup orientasi otomatis, perubahan ukuran, serta penerapan teknik augmentasi seperti *flip horizontal*, *grayscale*, penyesuaian kecerahan, dan penambahan *noise* pada gambar pelatihan.





Gambar pertama menunjukkan dataset *landmark* tangan yang diekstraksi menggunakan *MediaPipe* dan disimpan dalam format .CSV, di mana setiap baris berisi koordinat (x, y, z) dari 21 titik kunci tangan yang digunakan untuk klasifikasi isyarat tangan. Gambar kedua menunjukkan hasil deteksi tangan menggunakan *MediaPipe*, di mana titik-titik kunci tangan terdeteksi dan digambarkan pada gambar statis untuk memvisualisasikan koordinat yang diekstraksi dari citra tersebut.

Lampiran B

***Sour code* Arsitektur VGG-16**

```
n_feature = train_reshape.shape[2]
n_timestep = train_reshape.shape[1]
input = tf.keras.layers.Input(shape=(n_timestep, n_feature))

# Block 1
x = tf.keras.layers.Conv1D(64,3, padding="same", activation="relu")(input)
x = tf.keras.layers.Conv1D(64,3, padding="same", activation="relu")(x)
x = tf.keras.layers.MaxPool1D(pool_size=2)(x)

# Block 2
x = tf.keras.layers.Conv1D(128,3, padding="same",activation="relu")(x)
x = tf.keras.layers.Conv1D(128,3, padding="same",activation="relu")(x)
x = tf.keras.layers.MaxPool1D(2)(x)

# Block 3
x = tf.keras.layers.Conv1D(256,3, padding="same",activation="relu")(x)
x = tf.keras.layers.Conv1D(256,3, padding="same",activation="relu")(x)
x = tf.keras.layers.MaxPool1D(2)(x)

# Block 4
x = tf.keras.layers.Conv1D(512,3, padding="same",activation="relu")(x)
x = tf.keras.layers.Conv1D(512,3, padding="same",activation="relu")(x)
x = tf.keras.layers.MaxPool1D(2)(x)

# Block 5
x = tf.keras.layers.Conv1D(512,3, padding="same",activation="relu")(x)
x = tf.keras.layers.Conv1D(512,3, padding="same",activation="relu")(x)
x = tf.keras.layers.MaxPool1D(2)(x)

x = tf.keras.layers.Flatten()(x)
```

```
x = tf.keras.layers.Dropout(0.5)(x)
x = tf.keras.layers.Dense(512, activation="relu")(x)
x = tf.keras.layers.Dropout(0.3)(x)
x = tf.keras.layers.Dense(512, activation="relu")(x)
output = tf.keras.layers.Dense(2, activation="softmax")(x)

model_vgg = tf.keras.Model(input, output)
model_vgg.summary()
```

***Source Code* Arsitektir VGG-16 *with* Attention**

```
n_feature = train_reshape.shape[2]
n_timestep = train_reshape.shape[1]
input = tf.keras.layers.Input(shape=(n_timestep, n_feature))
# Block 1
x_block1 = tf.keras.layers.Conv1D(64,3, padding="same", activation="relu")(input)
x_block1 = tf.keras.layers.Conv1D(64,3, padding="same", activation="relu")(x_block1)

x_pool1 = tf.keras.layers.MaxPool1D(2, padding="same")(x_block1)
# Block 2
x_block2 = tf.keras.layers.Conv1D(128,3, padding="same",activation="relu")(x_pool1)
x_block2 = tf.keras.layers.Conv1D(128,3, padding="same",activation="relu")(x_block2)

x_pool2 = tf.keras.layers.MaxPool1D(2, padding="same")(x_block2)
# Block 3
x_block3 = tf.keras.layers.Conv1D(256,3, padding="same",activation="relu")(x_pool2)
x_block3 = tf.keras.layers.Conv1D(256,3, padding="same",activation="relu")(x_block3)
x_block3 = tf.keras.layers.Conv1D(256,3, padding="same",activation="relu")(x_block3)

x_pool3 = tf.keras.layers.MaxPool1D(2, padding="same")(x_block3)
# Block 4
x_block4 = tf.keras.layers.Conv1D(512,3, padding="same",activation="relu")(x_pool3)
x_block4 = tf.keras.layers.Conv1D(512,3, padding="same",activation="relu")(x_block4)
x_block4 = tf.keras.layers.Conv1D(512,3, padding="same",activation="relu")(x_block4)
```

```
x_pool4 = tf.keras.layers.MaxPool1D(2, padding="same")(x_block4)
# Block 5
x_block5 = tf.keras.layers.Conv1D(512,3, padding="same",activation="relu")(x_pool4)
x_block5 = tf.keras.layers.Conv1D(512,3, padding="same",activation="relu")(x_block5)
x_block5 = tf.keras.layers.Conv1D(512,3, padding="same",activation="relu")(x_block5)
x_pool5 = tf.keras.layers.MaxPool1D(2, padding="same")(x_block5)

average_pool4 = tf.keras.layers.GlobalAvgPool1D()(x_pool4)
average_pool5 = tf.keras.layers.GlobalAvgPool1D()(x_pool5)

# att1 = tf.keras.layers.Attention()([average_pool3, average_pool5])
att2 = tf.keras.layers.Attention()([average_pool4, average_pool5])
x = tf.keras.layers.Concatenate()([average_pool5, att2])

x = tf.keras.layers.Flatten()(x)
x = tf.keras.layers.Dropout(0.5)(x)
x = tf.keras.layers.Dense(512, activation="relu")(x)
x = tf.keras.layers.Dropout(0.3)(x)
x = tf.keras.layers.Dense(512, activation="relu")(x)
output = tf.keras.layers.Dense(2, activation="softmax")(x)

model_vgg_att = tf.keras.Model(input, output)
model_vgg_att.summary()
```

***Source Code* : Model MLP**

```
input = tf.keras.layers.Input(shape=(21*2,))
x = tf.keras.layers.Dropout(0.2)(input)
x = tf.keras.layers.Dense(30, activation="relu")(x)
x = tf.keras.layers.Dropout(0.2)(x)
x = tf.keras.layers.Dense(60, activation="relu")(x)
output = tf.keras.layers.Dense(2, activation="softmax")(x)

model_mlp = tf.keras.Model(input, output)
model_mlp.summary()
```