

**Deteksi Gestur Sistem Isyarat Bahasa Indonesia Menggunakan
Metode *Deep Learning* SSD *MobileNet V2 FPN* Lite**

Skripsi

Diajukan Untuk Memenuhi Persyaratan
Guna Meraih Gelar Sarjana Komputer



Nurul Tazkiyah Adam

2011501030

Bidang Minat

Rekayasa Perangkat Lunak

PROGRAM STUDI TEKNOLOGI INFORMASI

FAKULTAS SAINS DAN TEKNOLOGI

UNIVERSITAS 'AISYIAH YOGYAKARTA

2024

LEMBAR PENGESAHAN

Deteksi Gestur Sistem Isyarat Bahasa Indonesia Menggunakan Metode *Deep Learning* SSD *MobileNet V2* *FPN*Lite

SKRIPSI

Sebagai Persyaratan Guna Meraih Gelar Sarjana Komputer
Program Studi Teknologi Informasi
Universitas 'Aisyiyah Yogyakarta

Disusun Oleh:
Nurul Tazkiyah Adam
2011501030

Skripsi ini telah diuji dan dinyatakan lulus melalui sidang majelis
penguji pada 9 September 2024

Pembimbing 1


Zahra Arwananing
Tyas, S.Kom., M.Cs.
NIP. 8906111602343

Menyetujui

Pembimbing 2


Tikaridha Hardiani,
S.Kom., M.Eng.
NIP. 8910221611407

Penguji


Esi Putri Silmina,
S.T., M.Cs.
NIP. 8409241611408

Mengetahui

Ketua Program Studi Teknologi Informasi


Tikaridha Hardiani, S.Kom., M.Eng.
NIP. 8910221611407

LEMBAR PERNYATAAN KEASLIAN

Yang bertanda tangan dibawah ini:

NAMA : NURUL TAZKIYAH ADAM
NIM : 2011501030
FAK./PRODI. : SAINS DAN TEKNOLOGI/S1 TEKNOLOGI
INFORMASI

Dengan ini saya menyatakan bahwa skripsi dengan judul **“DETEKSI GESTUR SISTEM ISYARAT BAHASA INDONESIA MENGGUNAKAN METODE DEEP LEARNING SSD MOBILENET V2 FPNLITE”** beserta seluruh isinya adalah karya saya sendiri dan bukan merupakan karya tulis orang lain, baik sebagian maupun seluruhnya, kecuali dalam bentuk kutipan yang telah disebutkan sumbernya.

Demikian surat pernyataan ini saya buat dengan sebenar-benarnya. Apabila kemudian ditemukan adanya pelanggaran terhadap etika keilmuan dalam karya saya ini, atau ada klaim dari pihak lain terhadap keaslian karya saya ini maka saya siap menanggung segala bentuk resiko/sanksi yang berlaku.

Yogyakarta, 9 September 2024

Yang Membuat Pernyataan



Nurul Tazkiyah Adam

ABSTRAK

Penyandang disabilitas tunarungu memiliki keterbatasan pada indra pendengaran yang mengakibatkan hambatan dalam perkembangan bahasa sehingga memerlukan bimbingan dan pendidikan khusus. Penyandang disabilitas dengan usia 15 tahun ke atas sebagian besar baru mampu menyelesaikan pendidikan dengan tamatan SD berdasarkan data dari Badan Pusat Statistik. Tantangan lainnya yakni masih terdapat kesenjangan komunikasi antara tunarungu dan non-tunarungu karena tidak sedikit yang masih tidak mengerti bahasa isyarat. Seiring berkembangnya waktu, teknologi diadopsi untuk menyelesaikan permasalahan ini, salah satunya sistem *deep learning* yang mampu mendeteksi gestur SIBI (Sistem Isyarat Bahasa Indonesia). Penelitian ini menggunakan Arsitektur Model *Pre-Trained SSD MobileNet V2 FPNLite* dan *Framework TensorFlow Object Detection* untuk mendeteksi 15 gestur SIBI (Huruf Vokal A, I, U, E, O dan 10 kata yakni Baik, Bangku, Bel, Dia, Meja, Pramuka, Sakit, Saya, Teman, dan Tugas). Penelitian ini melibatkan *dataset* berjumlah 15.000 gambar yang dibagi menjadi 80% data pelatihan, 10% data validasi, dan 10% data pengujian. Kumpulan data mencakup variasi sudut, kamera, warna, latar belakang, jenis kelamin, maupun usia. Model dilatih selama 40.000 *steps* dengan *batch size* 16. Hasil evaluasi menunjukkan bahwa model berhasil mendeteksi 15 gestur SIBI tersebut dengan nilai akurasi model mAP @ 0.5:0.05:0.95 sebesar 89,93% dan nilai *loss* sebesar 0.1434.

Kata Kunci: SIBI, *Deep Learning*, *SSD MobileNet V2 FPNLite*, *TensorFlow Object Detection*, *mean Average Precision*.

Gesture Detection of Indonesian Sign Language Using Deep Learning Method SSD MobileNet V2 FPNLite

Nurul Tazkiyah Adam

Abstract

Hearing-impaired individuals face limitations in auditory perception, leading to challenges in language development and necessitating special guidance and education. According to data from the Central Statistics Agency, most people with hearing disabilities over the age of 15 have only completed elementary school education. Another challenge is the communication gap between the hearing-impaired and non-hearing-impaired communities, as many are still unfamiliar with sign language. With the advancement of technology, deep learning systems have been adopted to address this issue. This research utilized the Pre-Trained Model Architecture SSD MobileNet V2 FPNLite and the TensorFlow Object Detection Framework to detect 15 SIBI (Indonesian Sign Language System) gestures, including vowel letters (A, I, U, E, O) and 10 words (*Baik, Bangku, Bel, Dia, Meja, Pramuka, Sakit, Saya, Teman, Tugas*). The study involved a dataset of 15,000 images, divided into 80% training data, 10% validation data, and 10% testing data. The dataset includes variations in angles, cameras, colors, backgrounds, gender, and age. The model was trained for 40,000 steps with a batch size of 16. Evaluation results show that the model successfully detected the 15 SIBI gestures with a mean Average Precision (mAP) score @ 0.5:0.05:0.95 of 89.93% and a loss value of 0.1434.

Keywords : SIBI, Deep Learning, SSD MobileNet V2 FPNLite, TensorFlow Object Detection, Mean Average Precision.

LEMBAR PERSEMBAHAN

Puji syukur kepada Allah SWT atas rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi ini. Skripsi ini penulis persembahkan untuk:

1. Kedua orang tua tercinta, yang selalu memberikan doa, dukungan, cinta, dan semangat tanpa henti sepanjang perjalanan akademik penulis
2. Para dosen Program Studi Teknologi Informasi Universitas 'Aisyiyah Yogyakarta, yang telah memberikan ilmu, bimbingan, serta pengalaman berharga selama masa perkuliahan
3. Kakak-kakak, adik, dan keponakan tersayang, yang selalu menjadi sumber semangat dan motivasi
4. Rekan-rekan Teknologi Informasi, yang telah menjadi partner dalam perjalanan akademik, berbagi ilmu, tawa, dan tantangan
5. Sahabat Persicokor (Persatuan Istri Cowo Korea), yang selalu membuat hari-hari lebih berwarna dan penuh kebahagiaan
6. Park Jisung, yang selalu memberikan inspirasi, semangat, dan hiburan di saat penulis membutuhkan dorongan tambahan
7. Teman-teman Valzartafin Jogja, grup Ayo Vaksin, dan grup Bunda Valzar, yang selalu mendukung dan memberikan semangat dalam kebersamaan
8. Teman-teman yang tidak dapat penulis sebutkan satu per satu, yang kebbaikannya telah membersamai hingga detik ini.

Yogyakarta, 9 September 2024



Nurul Tazkiyah Adam

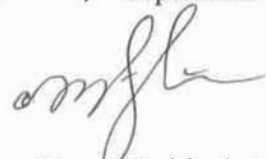
KATA PENGANTAR

Puji syukur kehadiran Allah SWT, yang telah memberikan rahmat, taufik, dan hidayah-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul "Deteksi Gestur Sistem Isyarat Bahasa Indonesia Menggunakan Metode *Deep Learning* SSD *MobileNet V2 FPNLite*" ini dengan baik. Skripsi ini disusun sebagai salah satu syarat untuk menyelesaikan Program Studi Sarjana Teknologi Informasi di Universitas 'Aisyiyah Yogyakarta. Penulis telah menerima banyak bantuan, bimbingan, serta dukungan dari berbagai pihak dalam proses penulisan skripsi ini. Oleh karena itu, sudah sepantasnya penulis dengan penuh hormat mengucapkan terimakasih dan mendoakan semoga Allah memberikan balasan terbaik kepada:

1. Ibu Dr. Warsiti, S.Kp., M.Kep., Sp.Mat., selaku Rektor Universitas 'Aisyiyah Yogyakarta
2. Ibu Tika Ainunnisa Fitria, S.T., M.T., Ph.D., selaku Dekan Fakultas Sains dan Teknologi Universitas 'Aisyiyah Yogyakarta
3. Ibu Tikaridha Hardiani, S.Kom., M.Eng., selaku Ketua Program Studi Teknologi Informasi Universitas 'Aisyiyah Yogyakarta dan pembimbing skripsi
4. Ibu Zahra Arwananing Tyas, S.Kom., M.Cs., selaku pembimbing skripsi
5. Ibu Esi Putri Silmina, S.T., M.Cs, selaku penguji skripsi
6. Bapak Sadr Lutfi Mufreni, S.Kom., M.Sc., selaku pembimbing akademik
7. Terima kasih penulis juga untuk semua pihak yang telah membantu peneliti dalam menyelesaikan skripsi ini yang tidak dapat peneliti sebutkan satu persatu.

Akhir kata, penulis menyadari bahwa hasil penelitian ini jauh dari kata sempurna.

Yogyakarta, 9 September 2024



Nurul Tazkiyah Adam

DAFTAR ISI

LEMBAR PENGESAHAN	ii
LEMBAR PERNYATAAN KEASLIAN	iii
ABSTRAK	iv
<i>ABSTRACT</i>	v
LEMBAR PERSEMBAHAN	vi
KATA PENGANTAR	vii
DAFTAR ISI.....	viii
DAFTAR GAMBAR	x
DAFTAR TABEL	xi
DAFTAR SOURCECODE	xii
DAFTAR GRAFIK	xiii
DAFTAR PERSAMAAN	xiv
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	5
1.3 Batasan Masalah.....	5
1.4 Tujuan Penelitian.....	5
BAB II TINJAUAN PUSTAKA.....	6
2.1 Kajian Pustaka	6
2.2 Tinjauan Teoritis	15
2.2.1 Sistem Isyarat Bahasa Indonesia.....	15
2.2.2 Deteksi Objek	15
2.2.3 <i>Deep Learning</i>	16
2.2.4 <i>SSD MobileNet V2 FPNLite</i>	17
2.2.5 <i>TensorFlow Object Detection</i>	19

2.2.6	<i>Google Colaboratory</i>	20
2.2.7	<i>Python</i>	20
2.2.8	<i>Mean Average Precision</i>	21
BAB III METODOLOGI PENELITIAN.....		25
3.1	Pengumpulan Data	25
3.2	Rancangan Data <i>Preprocessing</i>	28
3.2.1	Ubah Ukuran Gambar	28
3.2.2	Label Objek.....	28
3.2.3	Augmentasi	29
3.3	Rancangan Tahap Deteksi Objek	29
3.3.1	Menyiapkan Lingkungan Virtual.....	29
3.3.2	Rancangan Proses Pelatihan dan Pemodelan.....	30
3.3.3	Rancangan Uji Coba Model.....	33
3.3.4	Rencana Evaluasi Model	34
BAB IV HASIL DAN PEMBAHASAN		35
4.1	Hasil Pengumpulan Data	35
4.2	Hasil Data <i>Preprocessing</i> Awal	37
4.2.1	Hasil Ubah Ukuran Gambar	37
4.2.2	Hasil Label Objek Gestur SIBI.....	39
4.2.3	Hasil Augmentasi.....	41
4.3	Hasil Deteksi Objek.....	44
4.3.1	Hasil Menyiapkan Lingkungan Virtual	44
4.3.2	Hasil Pelatihan dan Pemodelan.....	44
4.3.3	Hasil Uji Coba Model.....	48
4.3.4	Hasil Evaluasi Model.....	53
BAB V KESIMPULAN DAN SARAN.....		57
5.1	Kesimpulan.....	57
5.2	Saran	58

DAFTAR GAMBAR

Gambar 1.1 Pendidikan Tertinggi yang Ditamatkan Penyandang Disabilitas dan Non-Disabilitas dalam Usia 15 Tahun ke Atas, 2022.	2
Gambar 2.1 Arsitektur SSD MobileNet V2 FPNLite.	17
Gambar 3.1 Diagram Alur Penelitian.	25
Gambar 3.2 Flowchart Rancangan Proses Pembuatan Model.	30
Gambar 3.3 Flowchart Rancangan Uji Coba Model.	33
Gambar 4.1 Contoh Hasil Ubah Ukuran Gambar.	39
Gambar 4.2 Memberi Label Objek Gestur SIBI.	39
Gambar 4.3 Isi Hasil Anotasi Label Objek Gestur SIBI.	40
Gambar 4.4 Hasil Anotasi Label Objek Gestur SIBI Di Dalam Folder.	41
Gambar 4.5 Contoh <i>Greyscale</i>	43
Gambar 4.6 Contoh <i>Redscale</i>	43
Gambar 4.7 Contoh <i>Greenscale</i>	43
Gambar 4.8 Contoh <i>Bluescale</i>	43
Gambar 4.9 Hasil Augmentasi Di Dalam Folder.	44
Gambar 4.10 <i>training_label.csv</i>	46
Gambar 4.11 <i>validation_labels.csv</i>	46
Gambar 4.12 <i>labelmap.txt</i>	46
Gambar 4.13 <i>labelmap.pbtxt</i>	46

DAFTAR TABEL

Tabel 2.1 Kajian Pustaka.....	11
Tabel 3.1 Panduan Gestur SIBI berdasarkan Kamus SIBI.....	27
Tabel 4.1 Contoh Hasil Variasi <i>Dataset</i> Tiap Gestur.....	35
Tabel 4.2 Detail Pembagian <i>Dataset</i>	45
Tabel 4.3 Hasil Uji Coba Model Gestur SIBI dengan Skor Kepercayaan Tertinggi	49
Tabel 4.4 Hasil Evaluasi Model	55

DAFTAR SOURCECODE

<i>Sourcecode 4.1 Source Code Ubah Ukuran Gambar.....</i>	<i>37</i>
<i>Sourcecode 4.2 Source Code Augmentasi Color Shifting.....</i>	<i>41</i>

DAFTAR GRAFIK

Grafik 4.1 <i>classification_loss</i>	47
Grafik 4.2 <i>localization_loss</i>	47
Grafik 4.3 <i>regularization_loss</i>	47
Grafik 4.4 <i>total_loss</i>	47
Grafik 4.5 <i>learning_rate</i>	47

DAFTAR PERSAMAAN

Persamaan 2.1 <i>Intersection Over Union (IoU)</i>	21
Persamaan 2.2 <i>Precision</i>	22
Persamaan 2.3 <i>Recall</i>	22
Persamaan 2.4 <i>Average Precision (AP)</i>	23
Persamaan 2.5 <i>Mean Average Precision (mAP)</i>	23
Persamaan 2.6 <i>MAP @ 0.5:0.05:0.95</i>	24

BAB I

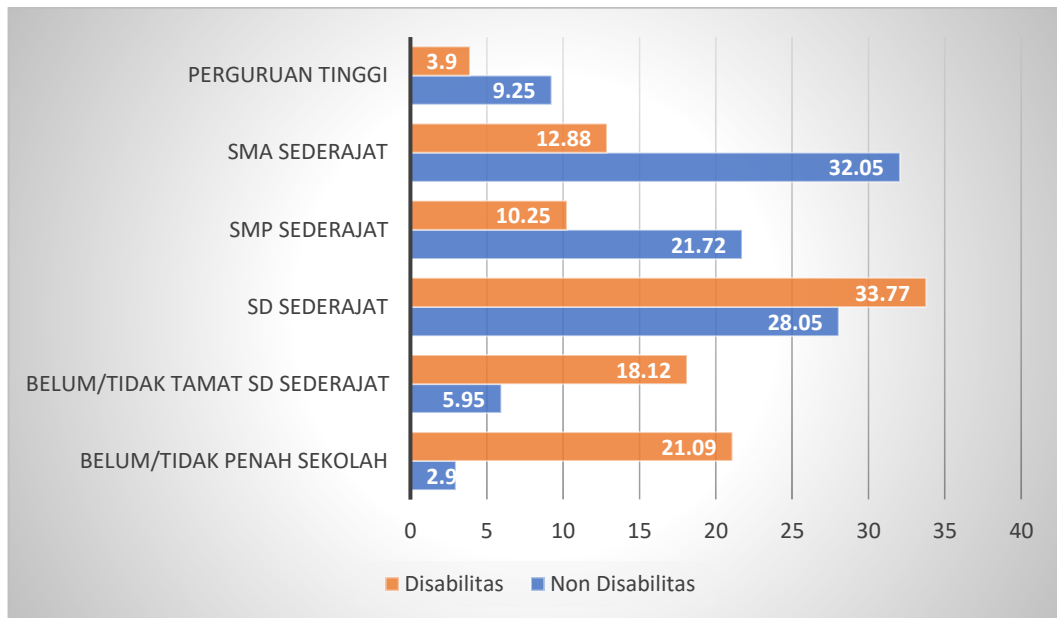
PENDAHULUAN

1.1 Latar Belakang

Pengertian penyandang disabilitas menurut Undang-Undang Republik Indonesia Pasal 1 ayat 1 Nomor 8 Tahun 2016 adalah setiap individu yang mengalami keterbatasan fisik, intelektual, mental dan/atau sensorik dalam jangka waktu yang lama, penyandang disabilitas dapat mengalami hambatan dan kesulitan untuk berpartisipasi secara penuh dan efektif dengan warga negara lainnya berdasarkan kesamaan hak (Pemerintah Republik Indonesia, 2016). Pengertian tunarungu secara medis yakni kurang atau hilangnya kemampuan dalam mendengar yang disebabkan oleh rusak atau tidak berfungsinya sebagian atau seluruh alat-alat pendengaran. Kurang atau hilangnya pendengaran dapat mengakibatkan hambatan dalam perkembangan bahasa sehingga memerlukan bimbingan dan pendidikan khusus (Suryanah, 1996).

Penyandang disabilitas, termasuk tunarungu, memiliki hak untuk mendapatkan kesempatan pendidikan yang setara dengan non-disabilitas serta menerima proses pembelajaran yang sesuai dengan kebutuhan dan kemampuan mereka. Akan tetapi, masih terdapat ketimpangan dalam capaian pendidikan antara penyandang disabilitas dan non-disabilitas. Penyandang disabilitas dengan usia 15 tahun ke atas, sebagian besar, baru mampu menamatkan pendidikan Sekolah Dasar (SD) sedangkan 40 persen dari mereka belum mampu atau bahkan tidak pernah mengenyam pendidikan SD dan hanya 16,78 persen yang mampu menamatkan

pendidikan sampai dengan SMA ke atas (Gambar 1.1) (Badan Pusat Statistik, 2023)



Gambar 1.1 Pendidikan Tertinggi yang Ditamatkan Penyandang Disabilitas dan Non-Disabilitas dalam Usia 15 Tahun ke Atas, 2022. Sumber: Badan Pusat Statistik.

Tantangan besar yang dihadapi oleh individu tunarungu, selain tantangan capaian pendidikan, yakni komunikasi, khususnya keterbatasan dalam berkomunikasi dengan orang yang tidak mengerti bahasa isyarat. Bahasa isyarat menjadi alat komunikasi yang digunakan oleh tunarungu, tetapi tidak sedikit orang yang tidak memahami bahasa tersebut. Hal ini menyebabkan kesenjangan komunikasi antara tunarungu dan individu non-tunarungu (Sari & Altiarika, 2023).

Sekolah Luar Biasa (SLB) merupakan sekolah khusus untuk para penyandang disabilitas mulai jenjang Sekolah Dasar hingga Sekolah Menengah Atas. Di Indonesia terdapat 2 jenis bahasa isyarat yakni Bahasa Isyarat Indonesia (BISINDO) dan Sistem Bahasa Isyarat Indonesia (SIBI). Keputusan Menteri

Pendidikan dan Kebudayaan Republik Indonesia Nomor 0161/U/2994, pemerintah membakukan Kamus Sistem Isyarat Bahasa Indonesia sehingga SIBI menjadi bahasa isyarat resmi yang dipakai seluruh SLB di bawah naungan Kementerian Pendidikan dan Kebudayaan (Kemendikbud) (Bagus et al., 2019).

Seiring berkembangnya waktu, teknologi diadopsi untuk upaya dalam mengatasi kesenjangan dalam komunikasi dan pendidikan antara tunarungu dan non-tunarungu, serta teknologi dapat memperkaya kreatifitas dalam pembelajaran SIBI seperti sistem *Deep Learning* yang dapat mendeteksi gerakan atau gestur SIBI (Sri Maulida, 2023). *Deep Learning* merupakan salah satu metode dalam kecerdasan buatan yang memungkinkan komputer untuk memproses data dengan cara yang meniru fungsi otak manusia. Model-model *Deep Learning* mampu mengenali pola-pola kompleks dalam berbagai jenis data seperti gambar, teks, dan suara, sehingga dapat menghasilkan wawasan dan prediksi yang akurat (Kahlil et al., 2023).

Penelitian tentang deteksi SIBI yang menerapkan *Deep Learning* sebelumnya sudah pernah dilakukan oleh beberapa peneliti, diantaranya penelitian yang mendeteksi 6 gestur SIBI (Saya, Dia, Kamu, Cinta, Maaf, dan Sedih) menggunakan *Convolutional Neural Network* (CNN) dan *dataset* sebanyak 660 gambar. Hasilnya menunjukkan masih terdapat nilai *loss* yang signifikan, yaitu 25% untuk pelatihan dan 23% untuk validasi, dan menunjukkan kesalahan prediksi yang cukup besar pada beberapa gestur (Ilham Rizaldy Widy Putra, 2021). Tahun selanjutnya, penggunaan arsitektur Model *Pre-Trained Single Shot Detector* (SSD) *MobileNet* V2 untuk mendeteksi 6 gestur SIBI yang sama dengan jumlah *dataset* sebanyak

600 gambar. Hasil terbaik menunjukkan nilai *loss* 0,1387 dan nilai akurasi 86,6% (Akbar Mahardi Hidayatullah, 2022).

Penelitian yang membandingkan performa arsitektur Model *Pre-Trained SSD MobileNet V2* dan Model *Pre-Trained SSD MobileNet V2 Feature Pyramid Network Lite (FPN Lite)* menggunakan *Framework TensorFlow Object Detection* untuk kasus deteksi helm pengendara dengan *dataset* 604 gambar menunjukkan *SSD MobileNet V2 FPN Lite* mempunyai akurasi yang baik dengan tingkat kepercayaan 99% dan memiliki nilai *loss* yang lebih kecil dengan waktu *training* model lebih cepat dibandingkan *SSD MobileNet V2* (Dionisius et al., 2024).

Penelitian ini akan mendeteksi gestur SIBI menggunakan metode *Deep Learning* dengan memanfaatkan *Framework TensorFlow Object Detection* dan arsitektur Model *Pre-Trained SSD MobileNet V2 FPN Lite*. Penelitian ini menggunakan sejumlah *dataset* yang terbagi ke dalam 15 gestur SIBI (Huruf Vokal A, I, U, E, O dan 10 kata yakni Baik, Bangku, Bel, Dia, Meja, Pramuka, Sakit, Saya, Teman, dan Tugas) dengan berbagai variasi sudut, kamera, warna, latar belakang, jenis kelamin, maupun usia. Diharapkan penelitian ini menghasilkan model deteksi yang memiliki ketepatan dalam mendeteksi dan nilai akurasi yang lebih baik dari penelitian sebelumnya. Selain itu, penelitian ini diharapkan dapat berkontribusi dalam pengetahuan sistem deteksi serta dapat membantu melestarikan dan mempopulerkan SIBI sebagai bahasa isyarat nasional Indonesia.

1.2 Rumusan Masalah

Rumusan masalah dalam penelitian ini:

1. Bagaimana menerapkan Metode *Deep Learning* dengan *Framework TensorFlow Object Detection* dan Model *Pre-trained SSD MobileNet V2 FPNLite* untuk mendeteksi 15 gestur SIBI?
2. Bagaimana nilai akurasi deteksi SIBI menggunakan *mean Average Precision* (mAP)?

1.3 Batasan Masalah

Batasan masalah dalam penelitian ini yakni:

1. Sistem hanya mampu mendeteksi 15 gestur SIBI yang mencakup Huruf Vokal (A, I, U, E, dan O) serta 10 kata yakni (Saya, Dia, Teman, Baik, Bel, Bangku, Meja, Sakit, Tugas, dan Pramuka) menggunakan Metode *Deep Learning* dengan *Framework TensorFlow Object Detection* dan Model *Pre-Trained SSD MobileNet V2 FPNLite*.
2. Evaluasi menggunakan mAP untuk mengetahui nilai akurasi deteksi SIBI.

1.4 Tujuan Penelitian

Tujuan pada penelitian sebagai berikut:

1. Penerapan Metode *Deep Learning* dengan *Framework TensorFlow Object Detection* dan Model *Pre-Trained SSD MobileNet V2 FPNLite* untuk mendeteksi 15 gestur SIBI.
2. Mendapatkan hasil nilai akurasi deteksi SIBI menggunakan mAP.

BAB II

TINJAUAN PUSTAKA

2.1 Kajian Pustaka

Kajian pustaka dilakukan untuk mengkaji penelitian terdahulu sebagai referensi dan masih memiliki kaitan dengan topik peneliti serta untuk melihat perbandingan antara peneliti dengan peneliti sebelumnya. Referensi dalam penelitian ini juga mengacu pada penelitian-penelitian terdahulu yang dirangkum berdasarkan penerapan Metode *Deep Learning* dalam perkembangan deteksi SIBI, Model *Pre-Trained SSD MobileNet V2 FPNLite*, *Framework TensorFlow Object Detection*, dan *mean Average Precision* (mAP) untuk mendapatkan nilai akurasi model.

Penerapan metode *Deep Learning* sudah terbukti mampu mendeteksi sebuah atau lebih objek dalam suatu gambar dan mengklasifikasikan objek tersebut sesuai kategorinya, seperti halnya dalam penelitian kasus deteksi bahasa isyarat resmi di Indonesia menggunakan model *pre-trained SSD MobileNet V2* berhasil mendeteksi 6 kata dalam gestur SIBI (Aku, Kamu, Dia, Cinta, Maaf, dan Sedih). Dalam metode *Deep Learning*, mesin atau model membutuhkan *dataset* untuk belajar mengenai objek yang akan dideteksi. Penelitian tersebut mengumpulkan *dataset* secara mandiri (Akbar Mahardi Hidayatullah, 2022). Penelitian lain yang mendeteksi 6 huruf dalam gestur SIBI (Huruf A, B, C, D, E, dan F) juga mengumpulkan *dataset* secara mandiri. Kemudian *dataset* dianotasi untuk memberikan informasi letak objek beserta label kategorinya pada setiap gambar dengan bantuan *labeling tools*.

Sebelum memasuki proses pelatihan atau *training*, file anotasi dalam penelitian tersebut diubah menjadi file dengan format yang lebih sesuai dengan standar *Framework TensorFlow Object Detection*, mengubah file dengan ekstensi *.xml* menjadi *.tfrecord*. Penelitian tersebut, menggunakan Model *Pre-Trained SSD MobileNet V2* dengan 3 skema pengujian pada parameter *step* yakni 10.000, 20.000 dan 30.000 *steps*, hasilnya menunjukkan semakin besar jumlah *step* dalam pelatihan nilai *loss* berkurang dan akurasi semakin baik (Muhammad Azka Imaddudin et al., 2022).

Pada tahun 2023, penelitian deteksi 26 huruf SIBI dengan penggunaan Model *SSD MobileNet* membandingkan beberapa konfigurasi parameter yakni perbandingan 5 rasio partisi *dataset*, 60%:40%, 70%:30%, 80%:20%, 90%:10%, dengan hasil akurasi 100% ketika rasio *dataset* 90%:10%. Hal ini menunjukkan semakin banyak data *train* yang dipelajari oleh model maka hasilnya semakin baik. Kemudian, perbandingan nilai pada *learning rate*, *epoch*, *batch size*, dan *step*. Penentuan nilai parameter ini mampu mempengaruhi kinerja model. Selanjutnya perbandingan jarak, cahaya, dan sudut. Hasil optimal ketika mendeteksi gestur SIBI dari jarak 50cm, disusul 100cm, sedangkan pada jarak 150cm hasil tidak terdeteksi sama sekali. Pengujian cahaya dengan *lux*: 5, 50, 100, dan 150, menunjukkan hasil nilai akurasi terbaik didapatkan pada saat nilai *lux* 100 dan *lux* 150 dengan akurasi 100%. Pada *lux* 50 juga masih terbilang baik 97%. Namun, mengalami penurunan signifikan pada saat *lux* 5, yaitu dengan akurasi 37%. Selanjutnya pengujian sudut kamera 0°, 30° kanan, 30° kiri, 60° kanan, 60° kiri. Hasilnya menunjukkan model berhasil mendeteksi gestur SIBI secara optimal ketika sudut kamera 0° dan 30°

kanan, sisanya nilai akurasi berada dibawah 50%. Dari pengujian ini menunjukkan model belum mendeteksi dengan baik ketika berada pada jarak 150cm, cahaya redup, dan variasi sudut dikarenakan model tidak mempelajari dan tidak mendapatkan variasi tersebut dalam *dataset* (Apendi & Paryasto, 2023).

Sejauh ini, penggunaan Model *Pre-Trained SSD MobileNet V2 FPNLite* belum dijumpai untuk deteksi SIBI. Namun, pada deteksi BISINDO sudah terdapat beberapa penelitian yang menerapkan Model *Pre-Trained SSD MobileNet V2 FPNLite*, seperti penelitian yang mendeteksi BISINDO 26 huruf abjad yakni dari huruf A hingga huruf Z dan deteksi berhasil pada penggunaan *background* yang berbeda, penggunaan aksesoris, ketika jarak 140 cm, dan ketika berada dalam lingkup cahaya terang, tetapi hasil masih buruk pada kondisi gelap (Hayati et al., 2023). Penelitian yang mendeteksi 26 huruf BISINDO yang menggunakan dua versi dari model ini yakni *SSD MobileNet V2 FPNLite 320x320* dan *SSD MobileNet V2 FPNLite 640x640*. Kedua model ini menunjukkan hasil yang tidak jauh berbeda yakni pada versi 320 mencapai nilai akurasi mAP 78%, *loss* 0.167, dan *learning rate* 0.054, kemudian pada versi 640 mencapai akurasi mAP 79%, *loss* 0.162, *learning rate* 0.054 (Joan et al., 2023). Tidak hanya BISINDO, pada tahun 2023 terdapat penelitian yang menggunakan model *pre-trained SSD MobileNet V2 FPNLite* untuk mendeteksi gestur kata dalam Bahasa Isyarat Malaysia (BIM), yakni Abang, Bapa, Emak, Saya, dan Sayang (Saiful Bahri et al., 2023).

Framework TensorFlow Object Detection sangat mempercepat pengembangan deteksi objek karena menyediakan berbagai macam model *pre-trained*, alat, dan dokumentasi yang komprehensif. Pada tahun 2022 terdapat

penelitian yang mengimplementasikan *Framework TensorFlow Object Detection* untuk mendeteksi gestur BISINDO dalam nama-nama hari yakni Senin, Selasa, Rabu, Kamis, Jumat, Sabtu dan Minggu dengan Model *Pre-Trained SSD MobileNet V2 FPNLite* serta mencapai nilai akurasi 97,14% (Rondiyanto, 2022). Penelitian tentang deteksi objek masker yang memanfaatkan *TensorFlow Object Detection Application Programming Interface (API)* dan *TensorFlow Lite Model Maker* mengkonversi model menjadi format *.tflite (TensorFlow Lite)* agar model dapat berjalan dengan baik pada perangkat terbatas. Penelitian tersebut juga menggunakan 4 Model *Pre-Trained Deep Learning* dengan hasil deteksi masker dalam 5 kategori menunjukkan nilai *Average Precision (AP)* pada *EfficientDet-Lite3* mencapai 75.84%, disusul oleh *SSD MobileNet V2 FPNLite 320x320* 72.4%, *SSD ResNet50 V1 FPN 640x640 (RetinaNet50)* 71.11%, dan *EfficientDet-Lite0* 70.43% (Ferdian Nursulistio, 2022).

Evaluasi model deteksi objek penting untuk memastikan model berfungsi dengan baik dalam mendeteksi dan mengklasifikasikan objek. Metrik mAP digunakan untuk mengevaluasi kinerja model deteksi objek. Standar *Intersection over Union (IoU)* 0.5 digunakan sebagai *threshold* untuk menentukan apakah prediksi model dianggap benar. Penelitian tentang deteksi kerusakan jalan menggunakan standar *threshold IoU* 0.5 untuk mendapatkan nilai mAP Model *Pre-Trained SSD MobileNet V2 FPNLite* versi 320 dan 640 dengan *dataset* sekitar 10.000 gambar. Untuk memudahkan dan meminimalisir kesalahan perhitungan dalam mengukur tingkat akurasi, peneliti tersebut menggunakan alat kalkulator mAP yang bersifat *open source* (Emir Fakhri Muttaqiy, 2024). Penelitian tentang

deteksi ekspresi antusias wajah menggunakan standar metrik dari kompetisi global *Microsoft Common Objects in Context (COCO)* untuk mendapatkan nilai mAP dalam berbagai ambang batas IoU, mulai dari 0.5 hingga 0.95, untuk memberikan gambaran yang lebih komprehensif tentang kinerja model pada berbagai tingkat *overlap* antara prediksi dan *ground truth* (Kais Ali Syafii Fatwari, 2022).

Setelah mengkaji penelitian-penelitian terdahulu, peneliti akan menerapkan Metode *Deep Learning* dengan arsitektur Model *Pre-Trained SSD MobileNetV2 FPNLite* untuk mendeteksi 15 gestur SIBI dengan *dataset* yang lebih bervariasi, penggunaan *Framework TensorFlow Object Detection* serta penerapan *mean Average Precision (mAP)* dengan metrik sesuai standar, *COCO Metrics*, untuk mendapatkan nilai akurasi. Dengan demikian, penelitian ini diharapkan dapat berkontribusi dalam deteksi SIBI dan mendapatkan hasil yang lebih baik dari penelitian sebelumnya. Rangkuman hasil kajian pustaka dapat dilihat pada Tabel 2.1.

Tabel 2.1 Kajian Pustaka

No	Penulis	Judul	Variabel	Metode	Hasil
1.	Akbar Mahardi Hidayatullah (2022)	Sistem Deteksi Simbol Pada SIBI (Sistem Isyarat Bahasa Indonesia) Secara Realtime Menggunakan <i>MobileNet-SSD</i>	SIBI: 6 kata (Aku, Kamu, Dia, Cinta, Maaf, dan Sedih.)	▪ Sigle Shot Multibox Detector (SSD) <i>MobileNet V2</i> ▪ <i>TensorFlow</i> API, Anaconda. ▪ <i>Dataset</i> 600 gambar. Pengujian hyperparameter yakni, 5.000 <i>step</i>, 10.000 <i>step</i>, dan 20.000 <i>step</i>.	Hasil total <i>loss</i> semakin mengecil ketika <i>step</i> training semakin besar dan <i>step</i> training paling besar memiliki nilai rata-rata akurasi paling tinggi, yakni 86,6%
2.	Muhammad Azka Imaddudin (2022)	Simulasi Penerjemah SIBI (Sistem Isyarat Bahasa Indonesia) Menggunakan <i>TensorFlow</i> dan <i>Convolutional Neural Network</i> (CNN)	SIBI: 6 huruf (A, B, C, D, E, dan F)	▪ Sigle Shot Multibox Detector (SSD) <i>MobileNet V2</i> ▪ <i>Framework: TensorFlow Object Detection</i> (TFOD) ▪ <i>Dataset</i> 1200 gambar. Pengujian hyperparameter yakni 10.000 <i>step</i>, 20.000 <i>step</i>, 30.000 <i>step</i>.	Hasil yang didapatkan yakni skema ketiga dengan 30.000 <i>step</i> menunjukkan hasil yang paling baik, memiliki nilai <i>loss</i> paling rendah dan nilai mAP mencapai 0,7594.
3.	Siroojuddin Apendi, Casi Setianingsih, Dan Marisa W. Paryasto (2023)	Deteksi Bahasa Isyarat Sistem Isyarat Bahasa Indonesia Menggunakan Metode <i>Single Shot Multibox Detector</i>	SIBI: 26 huruf, A-Z.	▪ Sigle Shot Multibox Detector (SSD) <i>MobileNet</i> ▪ <i>Framework: TensorFlow Object Detection</i> (TFOD) ▪ <i>Dataset</i> 780 gambar. Skema pengujian: 1) partisi <i>dataset</i> 5 rasio, 60:40, 70:30, 80:20, 90:10 2) <i>learning rate</i>: 0.04, 0.06, 0,08 3) <i>epoch</i>: 1, 150, 300 4) <i>batch size</i>: 4, 6, 8 5) <i>step</i>: 10k, 20k, 30k, 40k 6) jarak: 50cm, 100cm, 150cm 7) <i>lux</i>: 5, 50, 100, 150 8) sudut kamera: 0°, 30° kanan, 30° kiri, 60° kanan, 60° kiri	Hasil pengujian konfigurasi hyperparameter mendapatkan akurasi 100%, mAP.50IoU 100%. Hasil pengujian yang memiliki akurasi terbaik, rasio <i>dataset</i> 90:10, <i>learning rate</i> 0.04, <i>epoch</i> 300, <i>batch size</i> 4, <i>step</i> 40k, 100%, yakni pada posisi jarak objek 50cm dari kamera, pencahayaan <i>lux</i> 100 dan <i>lux</i> 150, dan saat sudut 0° kamera.

No	Penulis	Judul	Variabel	Metode	Hasil
4.	Lilis Nur Hayati, Anik Nur Handayani, Wahyu Sakti Gunawan Iriantoa, Rosa Andrie Asmarab, Dolly Indra, Muhammad Fahmi (2023)	<i>Classifying BISINDO Alphabet Using TensorFlow Object Detection API</i>	BISINDO: 26 huruf, A-Z	- SSD MobileNet V2 FPNLite - Framework TensorFlow Object Detection - Dataset 4.054 gambar dan 20.000 steps	hasilnya AP 71%. Pengujian berhasil menggunakan background yang berbeda, menggunakan aksesoris, jarak 140 cm, dan cahaya terang, akan tetapi hasil masih buruk pada kondisi gelap.
5.	David Joan, Vincent, Kevin Jason Daniel, Said Achmad, Rhio Sutoyo, (2023)	<i>BISINDO Hand-Sign Detection Using Transfer Learning</i>	BISINDO: 26 huruf, A-Z.	- SSD MobileNet V2 FPNLite 320x320 dan SSD MobileNet V2 FPNLite 640x640 - Framework TensorFlow Object Detection - Dataset 520 gambar dan 20.000 steps	Hasilnya SSD MobileNet V2 FPNLite 320x320 mencapai nilai mAP 78%, loss 0.167, dan learning rate 0.054, kemudian pada SSD MobileNet V2 FPNLite 640x640 mencapai mAP 79%, loss 0.162, learning rate 0.054.
6.	Iffah Zulaikha Saiful Bahri, Sharifah Saon, Abd Kadir Mahamad, Khalid Isa, Umi Fadlilah (2023)	<i>Interpretation of Bahasa Isyarat Malaysia (BIM) Using SSD-MobileNet-V2 FPNLite and COCO mAP</i>	BIM: 26 huruf A-Z dengan penambahan <i>space</i> , <i>delete</i> , dan <i>nothing</i> . 5 kata (Abang, Bapa, Emak, Saya, dan Sayang).	- MobileNet dan SSD MobileNet V2 FPNLite - Framework TensorFlow Object Detection - Dataset 87.000 untuk 29 kelas (huruf dan 3 tambahan), - Dataset 500 gambar untuk 5 kelas kata dengan percobaan 2.000 dan 2.500 steps	MobileNet digunakan untuk melatih 29 kelas dan hasil akurasi <i>training</i> mencapai 99.75%. Pada 5 gestur kata di <i>train</i> menggunakan SSD MobileNet V2 FPNLite hasilnya akurasi yang dicapai 61.60%

No	Penulis	Judul	Variabel	Metode	Hasil
7.	Rondiyanto (2022)	Implementasi <i>Framework TensorFlow Object Detection</i> Dalam Mendeteksi Gestur Bahasa Isyarat	BISINDO: 7 nama hari (Senin, Selasa, Rabu, Kamis, Jum'at, Sabtu, dan Minggu).	- SSD <i>MobileNet V2 FPNLite</i> <i>Framework TensorFlow Object Detection</i> - Dataset 1654 gambar	Mencapai akurasi 97,14% setelah <i>epoch</i> ke 4.000
8.	Ferdian Nursulistio (2022)	Deteksi Objek Masker Menggunakan <i>Object Detection API</i> Dan <i>TensorFlow Lite Model Maker</i>	5 kelas: masker medis, masker kain, masker scuba, tidak menggunakan masker dengan benar, atau tidak menggunakan masker.	<i>EfficientDet-Lite0</i>, <i>EfficientDet-Lite3</i>, SSD <i>MobileNet V2 FPNLite</i> 320x320, SSD <i>ResNet50 V1 FPN</i> 640x640 (<i>RetinaNet50</i>) <i>Framework TensorFlow Object Detection</i> - Dataset 1598 gambar. Pada <i>MobileNet</i> menggunakan 30.000 <i>steps</i>, <i>ResNet</i> menggunakan 20.000 <i>steps</i>	Hasil <i>Average Precision (AP)</i> : <i>EfficientDet-Lite3</i> 75.84%, SSD <i>MobileNet V2 FPNLite</i> 320x320 72.4%, SSD <i>ResNet50 V1 FPN</i> 640x640 (<i>RetinaNet50</i>) 71.11%, dan <i>EfficientDet-Lite0</i> 70.43%.
9.	Kais Ali Syafii Fatwari (2022)	Model Pengenalan Ekspresi Antusiasme Wajah Dengan Menggunakan Arsitektur SSD <i>MobileNet V2 FPNLite</i>	2 kelas: ekspresi wajah antusias dan tidak antusias.	- SSD <i>MobileNet V2 FPNLite</i> <i>Framework TensorFlow Object Detection</i> - Dataset 500 gambar, 25.000 <i>steps</i>, <i>batch size</i> 16, <i>epoch</i> 1.000.	Hasil mAP dengan ambang batas IoU 0.5 hingga 0.95 mencapai 77%. Hasil pengujian pada kondisi normal tegak lurus dan sedikit mendapat akurasi sebesar 97% dan 90% untuk antusias, 99% dan 100% untuk tidak antusias. Untuk objek dengan atribut kacamata dan jilbab masing-masing mendapat 98% dan 94% untuk antusias, lalu 99% dan 99% untuk tidak antusias. Sedangkan objek dengan masker tidak dapat terdeteksi pada antusias dan mendapat 82% pada tidak antusias.

No	Penulis	Judul	Variabel	Metode	Hasil
10.	Emir Fakhri Muttaqiy (2024)	Deteksi Kerusakan Jalan Menggunakan Metode <i>Deep Learning SSD MobileNet V2 FPNLite</i>	Jenis kerusakan jalan: retak kulit buaya (<i>alligator crack</i>), retak melintang (<i>transverse crack</i>), lubang (<i>potholes</i>), dan retak membujur (<i>longitudinal crack</i>).	- SSD <i>MobileNet V2 FPNLite</i> 320x320 dan SSD <i>MobileNet V2 FPNLite</i> 640x640 - Framework <i>TensorFlow Object Detection</i> - Dataset 10.000 gambar dengan 10.000 steps	Model deteksi SSD <i>MobileNet V2 FPNLite</i> 320 memperoleh skor mAP sebesar 43,21% pada ambang batas IoU 0.50, sedangkan SSD <i>MobileNet V2 FPNLite</i> 640 memperoleh skor mAP sebesar 46,48% pada ambang batas IoU 0.50.

2.2 Tinjauan Teoritis

2.2.1 Sistem Isyarat Bahasa Indonesia

Bahasa isyarat merupakan bahasa yang memanfaatkan tangan, ekspresi wajah, gerak dan gerakan tubuh untuk berkomunikasi tentang pengalaman, kebutuhan, dan perasaan. Bahasa isyarat juga memiliki tata bahasa yang terorganisir dan terstruktur seperti bahasa lisan. Salah satu bahasa isyarat di Indonesia yakni SIBI. SIBI merupakan salah satu media yang dapat membantu komunikasi sesama kaum tunarungu maupun tunarungu dan non-tunarungu di lingkungan masyarakat. SIBI memiliki tatanan sistematis (struktural) mengenai seperangkat isyarat jari tangan dan berbagai gerak yang melambangkan kosakata Bahasa Indonesia (Dewi Ratih Rapisa, 2021).

2.2.2 Deteksi Objek

Deteksi objek atau *object detection* adalah teknologi komputer yang memproses gambar untuk mendeteksi objek di dalamnya. Sering kali terdapat kebingungan antara definisi deteksi gambar dan klasifikasi gambar. Perbedaan mendasar antara keduanya terletak pada tujuan penggunaannya. Klasifikasi gambar digunakan untuk mengklasifikasikan *item* dalam gambar, sedangkan deteksi gambar digunakan untuk menemukan objek, misalnya, menentukan letak objek dalam gambar. Deteksi adalah tahap pengenalan atau pengamatan awal suatu objek. Pada interpretasi citra, tahap deteksi merupakan proses menentukan apakah keberadaan suatu objek memenuhi standar kelayakan untuk diamati. Deteksi objek adalah proses menemukan objek dengan kotak pembatas dan mengidentifikasi jenis atau kelas objek yang ada dalam gambar. *Input* dari deteksi objek adalah gambar

yang mengandung satu atau lebih objek, seperti foto. *Output* dari proses ini berupa satu atau lebih kotak pembatas (misalnya ditentukan oleh koordinat, lebar, dan tinggi), serta label kelas untuk setiap kotak pembatas (Achmar Fahrurozi et al., 2022).

2.2.3 *Deep Learning*

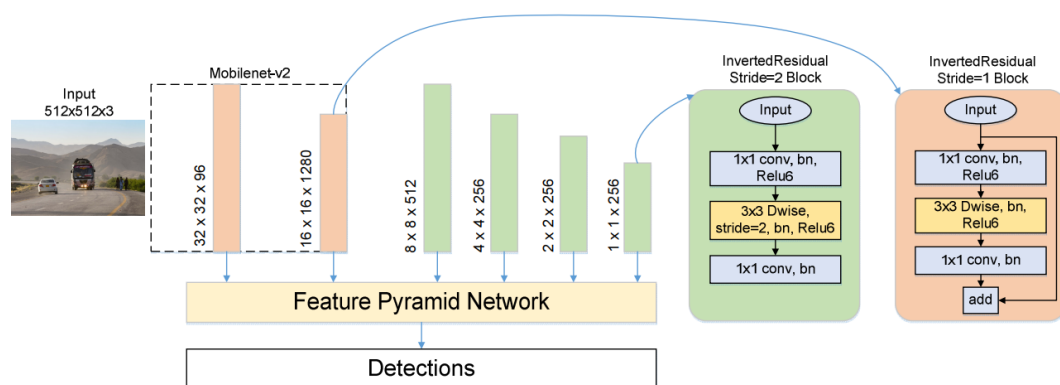
Deep Learning merupakan salah satu metode dalam kecerdasan buatan yang memungkinkan komputer untuk memproses data dengan cara yang meniru fungsi otak manusia. Model-model *Deep Learning* mampu mengenali pola-pola kompleks dalam berbagai jenis data seperti gambar, teks, dan suara, sehingga dapat menghasilkan wawasan dan prediksi yang akurat (Kahlil et al., 2023). *Deep Learning* termasuk cabang dari pembelajaran mesin yang menggunakan jaringan saraf tiruan dengan banyak lapisan, yang dikenal sebagai jaringan saraf dalam (*deep neural networks*) (Ian Goodfellow et al., 2016).

Deep Learning merupakan metode yang memanfaatkan struktur berlapis-lapis untuk mempelajari representasi data secara hierarkis. Konsep hierarki ini berarti bahwa Model *Deep Learning* dapat membangun representasi data pada berbagai tingkat abstraksi. Pada lapisan awal jaringan, model ini belajar mengenali fitur dasar seperti tepi atau pola warna dalam gambar. Fitur-fitur ini kemudian digabungkan pada lapisan berikutnya untuk membentuk representasi yang lebih kompleks, seperti bentuk atau objek. Lapisan-lapisan yang lebih dalam pada jaringan saraf ini kemudian dapat memahami dan mengenali konsep yang lebih abstrak, seperti pengenalan wajah atau objek tertentu. Pengembangan *Model Deep Learning* terdiri dari beberapa tahap utama. Proses dimulai dengan pengumpulan

data, yang diikuti oleh *preprocessing*, termasuk *resize*, *labeling*, dan augmentasi untuk memperkaya variasi data. Model dibangun menggunakan arsitektur *deep learning* dan dilatih menggunakan algoritma optimasi seperti *Stochastic Gradient Descent* (SGD) untuk meminimalkan *loss*. Hasil model *training* akan diuji dengan data baru untuk mengevaluasi performanya melalui metrik seperti akurasi *mean Average recision* (mAP). Semua tahapan ini saling berhubungan untuk menghasilkan model yang optimal. (Ian Goodfellow et al., 2016).

2.2.4 SSD MobileNet V2 FPNLite

SSD *MobileNet V2 FPNLite* adalah model deteksi objek yang dirancang untuk efisiensi tinggi dan performa optimal pada perangkat dengan keterbatasan sumber daya. Model ini mengintegrasikan SSD dengan *backbone MobileNet V2* dan *Feature Pyramid Network Lite* (FPNLite) untuk memberikan deteksi objek yang akurat dengan meminimalkan penggunaan memori dan beban komputasi. Gambar 2.1 merepresentasikan arsitektur SSD *MobileNet V2 FPNLite* oleh (Chiu et al., 2020).



Gambar 2.1 Arsitektur SSD *MobileNet V2 FPNLite*. Sumber gambar: (Chiu et al., 2020)

Saat gambar dimasukkan ke dalam model, proses pertama yang dilakukan adalah ekstraksi fitur menggunakan *MobileNet V2*. Sebagai *backbone* dari model, *MobileNet V2* memproses gambar menggunakan blok *residual* terbalik dan konvolusi *depth-wise separable* untuk mengubah gambar mentah menjadi banyak peta fitur yang mengandung informasi penting. Peta fitur ini adalah serangkaian matriks yang menangkap berbagai aspek dari gambar, seperti tepi, tekstur, bentuk, dan pola lainnya, yang kemudian digunakan untuk analisis lebih lanjut oleh model. Blok *residual* terbalik membantu memastikan transfer informasi yang baik ke lapisan jaringan yang lebih dalam, sementara konvolusi *depth-wise separable* dapat mengurangi jumlah operasi konvolusi sehingga dapat meningkatkan efisiensi dalam komputasi model (Sandler et al., 2018).

Setelah peta fitur diekstraksi oleh *MobileNet V2*, *FPN Lite* mengambil alih untuk mengoptimalkan peta fitur multi-skala yang dihasilkan (Chiu et al., 2020). *FPN Lite*, sebagai varian dari *Feature Pyramid Network* (FPN), menggabungkan peta fitur dari berbagai tingkat resolusi dengan menggunakan lapisan konvolusi 1x1 untuk menyamakan jumlah saluran dan penyesuaian ukuran peta fitur sebelum digabungkan. Proses ini meningkatkan akurasi deteksi dengan memanfaatkan informasi dari berbagai tingkat resolusi fitur sambil menjaga kompleksitas model tetap rendah (Lin et al., 2016).

Setelah peta fitur diproses oleh *FPN Lite*, SSD melakukan tahap akhir deteksi objek. SSD menggunakan peta fitur yang telah digabungkan untuk mendeteksi objek dalam gambar, mengidentifikasi kelas objek, dan menentukan *bounding box* yang paling sesuai berdasarkan nilai kepercayaan tertinggi. SSD menggabungkan

klasifikasi dan regresi *bounding box* dalam satu tahap, memungkinkan deteksi objek yang cepat dan efisien (Liu et al., 2015).

Arsitektur SSD *MobileNet V2 FPNLite*, yang mengintegrasikan SSD, *MobileNet V2*, dan *FPNLite*, mencapai keseimbangan antara performa dan efisiensi, menjadikannya solusi ideal untuk deteksi objek pada perangkat dengan keterbatasan sumber daya. Model ini menyediakan deteksi objek yang akurat dan cepat sambil menjaga penggunaan memori dan beban komputasi pada tingkat optimal (Chiu et al., 2020).

2.2.5 TensorFlow Object Detection

TensorFlow adalah sebuah antarmuka yang digunakan untuk mengekspresikan algoritma pembelajaran mesin dan sebuah implementasi untuk menjalankan algoritma-algoritma tersebut. *TensorFlow* dirancang untuk dijalankan pada berbagai sistem heterogen, mulai dari perangkat seluler seperti telepon dan tablet hingga sistem terdistribusi besar yang melibatkan ratusan mesin dan ribuan perangkat komputasi seperti *Graphical Processing Unit* (GPU). Hal ini memungkinkan perhitungan yang diekspresikan menggunakan *TensorFlow* dapat dijalankan dengan sedikit atau tanpa perubahan pada berbagai *platform* (Abadi et al., 2016).

TensorFlow Object Detection adalah sebuah *framework open-source* yang dikembangkan menggunakan *TensorFlow*. *Framework* ini dirancang untuk memudahkan proses pembuatan, pelatihan, dan implementasi model deteksi objek. *TensorFlow Object Detection API* menyediakan berbagai model yang telah dilatih sebelumnya (*pre-trained models*) serta alat untuk melatih model yang dipilih.

Framework ini membantu pengguna untuk mengembangkan sistem deteksi objek sesuai dengan spesifik yang dibutuhkan (Abadi et al., 2016).

2.2.6 Google Colaboratory

Google Colaboratory, atau sering disingkat *Google Colab*, adalah *platform* berbasis *cloud* yang memungkinkan pengguna untuk menulis dan menjalankan kode *Python* pada *browser*. Beberapa *Library Python* yang sering digunakan dalam *Google Colab* untuk *Deep Learning* dan analisis data meliputi *TensorFlow*, *Keras*, *NumPy*, *Pandas*, *Matplotlib*, *Seaborn*, dan *OpenCV*. *Platform* ini juga sangat berguna untuk analisis data, pembelajaran mesin, dan pengembangan Model *Deep Learning* karena menyediakan akses gratis ke GPU dan *Tensor Processing Unit* (TPU). Pengguna dapat membuat, menyimpan, dan berbagi *notebook* (Francois Chollet, 2018).

2.2.7 Python

Python adalah salah satu bahasa pemrograman yang digunakan untuk menulis skrip dalam konteks komputasi ilmiah dan teknik. *Python scripting* mengacu pada penggunaan bahasa untuk mengotomatisasi proses komputasi, memanipulasi data, dan menjalankan simulasi numerik. *Python* memfasilitasi pemrograman ilmiah dengan sintaks yang sederhana dan efisien, memungkinkan ilmuwan dan insinyur untuk mengembangkan algoritma dan model, serta mengotomatisasi tugas-tugas komputasi yang kompleks dengan fleksibilitas yang tinggi (Hans Petter Langtangen, 2008).

2.2.8 Mean Average Precision

Proses evaluasi dalam deteksi objek menggunakan *mean Average Precision* (mAP) untuk mengukur kinerja model. mAP digunakan dalam COCO (*Common Objects in Context*) *Metrics*, yang pertama kali diperkenalkan oleh (Lin et al., 2015) dalam kompetisi global *Microsoft COCO*. Metrik COCO telah diterapkan dalam banyak penelitian untuk mengevaluasi kinerja model. Metrik COCO mencakup perhitungan mAP @ 0.5:0.05:0.95, yang menghitung *Average Precision* (AP) untuk setiap kelas dan setiap *threshold Intersection over Union* (IoU) dari 0.5 hingga 0.95 dengan kenaikan 0.05. Nilai mAP memberikan rata-rata dari nilai AP di berbagai *threshold* IoU sebagai gambaran menyeluruh tentang kinerja model deteksi objek dalam berbagai tingkat tumpang tindih. Metrik COCO telah diterapkan dalam banyak penelitian untuk mengevaluasi kinerja model. Rumus-rumus evaluasi metrik COCO untuk mendapatkan nilai mAP (Lin et al., 2015), sebagai berikut:

1. *Intersection over Union* (IoU)

IoU adalah metrik yang digunakan untuk mengevaluasi kesesuaian antara kotak prediksi dan kotak *ground truth*. IoU mengukur seberapa baik kotak prediksi menutupi kotak sebenarnya.

$$IoU = \frac{\text{Area of Intersection}}{\text{Area of Union}} \quad \textbf{Persamaan 2.1 Intersection Over Union (IoU)}$$

Keterangan:

- *Area of Intersection*: Luas area yang tumpang tindih antara kotak prediksi dan kotak *ground truth*.

- *Area of Union*: Luas gabungan dari area kotak prediksi dan kotak *ground truth*.
- IoU memberikan nilai antara 0 dan 1, di mana 1 berarti kotak prediksi sepenuhnya tumpang tindih dengan kotak *ground truth* dan 0 berarti tidak ada tumpang tindih sama sekali.

2. Average Precision (AP)

➤ Precision

Precision adalah rasio prediksi benar terhadap total prediksi.

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives} \quad \textbf{Persamaan 2.2 Precision}$$

Keterangan:

- *True Positive* (TP): jumlah kasus di mana model benar-benar memprediksi kelas positif dengan benar.
- *False Positive* (FP): jumlah kasus di mana model salah memprediksi kelas negatif sebagai positif.

➤ Recall

Recall adalah rasio prediksi benar terhadap total kasus sebenarnya.

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives} \quad \textbf{Persamaan 2.3 Recall}$$

Keterangan:

- *True Positive* (TP): jumlah kasus di mana model benar-benar memprediksi kelas positif dengan benar.
- *False Negative* (FN): jumlah kasus di mana model gagal memprediksi kelas positif yang sebenarnya ada.

➤ *Average Precision* (AP)

Area di bawah kurva *Precision-Recall* dihitung sebagai nilai AP

$$AP = \int_{r=0}^1 \text{Precision}(r) dr \quad \textbf{Persamaan 2.4 Average Precision (AP)}$$

Keterangan:

- *Precision*(r) adalah precision pada tingkat *recall* r .
- *Precision* terletak pada sumbu y.
- *Recall* terletak pada sumbu x.

3. *Mean Average Precision* (mAP)

➤ *Mean Average Precision* (mAP)

Untuk mendapatkan nilai mAP dari total nilai AP dalam *threshold* IoU tertentu ($\text{mAP}_{\text{IoU}0.5}$, $\text{mAP}_{\text{IoU}0.55}$, dst) dan nilai mAP dari total AP kelas tertentu dalam rentang IoU 0.5:0.05:0.9.5 (mAP_{baik} , mAP_{bel} , dst)

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad \textbf{Persamaan 2.5 Mean Average Precision (mAP)}$$

Keterangan:

- N adalah jumlah entitas yang dihitung

- AP_i adalah *average precision* untuk entitas ke- i
- Mean *Average Precision* (mAP) secara keseluruhan

Untuk mendapatkan nilai akurasi mAP model secara keseluruhan dalam COCO metriks yakni mAP @ 0.5:0.05:0.95.

$$mAP_{overall} = \frac{1}{N} \sum_{i=1}^N mAP_i \quad \textbf{Persamaan 2.6 MAP @ 0.5:0.05:0.95}$$

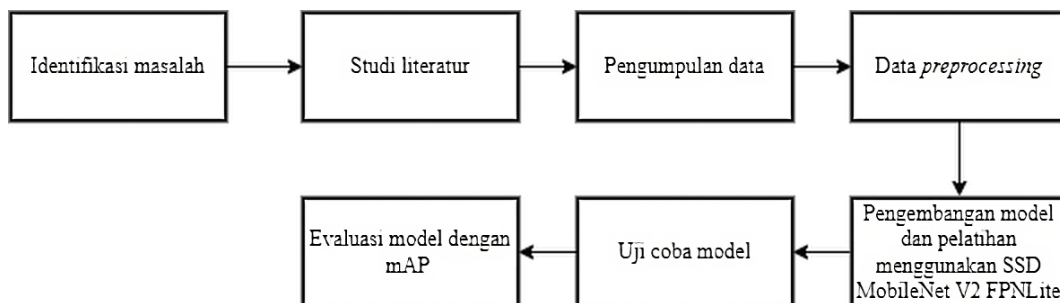
Keterangan:

- N adalah jumlah entitas
- mAP_i adalah *mean average precision* untuk entitas ke- i

BAB III

METODOLOGI PENELITIAN

Metode penelitian akan dilakukan dengan beberapa tahapan kerja untuk mencapai tujuan penelitian, yaitu model dapat mendeteksi 15 (lima belas) kelas gestur SIBI dan mengetahui berapa nilai akurasinya. Gambaran tahapan-tahapan kerja tersebut dapat dilihat pada diagram alur penelitian Gambar 3.1.



Gambar 3.1 Diagram Alur Penelitian.

3.1 Pengumpulan Data

Proses pengumpulan data pada penelitian ini dilakukan secara primer, gambar-gambar dikumpulkan langsung oleh peneliti menggunakan berbagai perangkat kamera *mobile*. *Dataset* yang digunakan terdiri dari gambar-gambar atau citra yang diambil dengan rasio aspek yang konsisten, yaitu 16:9, atau dalam mode potret, menggunakan berbagai merek kamera *device* yang berbeda-beda seperti Samsung Galaxy S22, Samsung Galaxy S23 Ultra, Vivo Y12, Xiaomi Redmi Note13, dan iPhone 12. Pemilihan berbagai merek kamera ini dilakukan untuk memastikan variasi kualitas gambar yang dihasilkan

Latar belakang atau *background* pengambilan gambar dipilih dengan beragam situasi, baik di dalam ruangan maupun di luar ruangan. Hal ini dipertimbangkan untuk memperoleh variasi pencahayaan dan latar belakang yang lebih beragam, yang dapat mempengaruhi kualitas dan karakteristik visual dari gambar-gambar yang diambil.

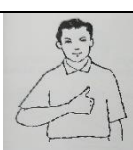
Subjek dalam gambar-gambar ini adalah manusia, yang mencakup berbagai perbedaan seperti jenis kelamin (pria dan wanita), yang memiliki variasi rambut pendek, rambut panjang, dan penggunaan jilbab pada subjek wanita. Selain itu, subjek juga dipilih dengan melibatkan variasi usia, mulai dari anak-anak, remaja, hingga dewasa. Hal ini dilakukan untuk memperluas variasi bentuk dan ukuran tangan yang mengilustrasikan gestur SIBI karena tangan antar kelompok usia berbeda.

Pengambilan gambar dilakukan dengan variasi sudut, yakni dari sisi depan, kanan, dan kiri, serta dengan fokus memperbesar bagian tangan yang membentuk gestur SIBI. Setiap subjek atau orang yang berpartisipasi dalam penelitian diminta untuk membentuk 15 jenis gestur SIBI yang berbeda. Gestur-gestur ini mencakup huruf vokal SIBI (A, I, U, E, dan O) dan sepuluh kata SIBI (Saya, Dia, Teman, Baik, Bel, Bangku, Meja, Sakit, Tugas, dan Pramuka). Pemilihan kata-kata ini didasarkan pada kosa kata yang akrab didengar dalam lingkungan sekolah. Panduan gestur mengacu pada Kamus SIBI yang dikeluarkan oleh Kementerian Pendidikan dan Kebudayaan (Kemendikbud) (Tabel 3.1).

Dataset yang dikumpulkan dalam penelitian ini tidak hanya mencakup variasi visual yang luas, tetapi juga memastikan bahwa gestur-gestur yang direkam sesuai

dengan standar dan pedoman gestur SIBI resmi yang ada di Indonesia, sehingga dapat dijadikan *dataset* yang valid untuk penelitian sistem deteksi gerakan SIBI. Total jumlah *dataset* awal yang dikumpulkan peneliti adalah 3.000 gambar dengan masing-masing gestur memiliki 200 gambar.

Tabel 3.1 Panduan Gestur SIBI berdasarkan Kamus SIBI

No.	Bahasa Lisan	SIBI
1.	A	
2.	E	
3.	I	
4.	O	
5.	U	
6.	Baik	
7.	Bangku	
8.	Bel	

No.	Bahasa Lisan	SIBI
9.	Dia	
10.	Meja	
11.	Pramuka	
12.	Sakit	
13.	Saya	
14.	Teman	
15.	Tugas	

3.2 Rancangan Data *Preprocessing*

Data *preprocessing* meliputi serangkaian teknik maupun prosedur yang digunakan untuk mempersiapkan dan mengubah data mentah menjadi format yang cocok dalam pelatihan model *Deep Learning*. Proses ini sangat penting karena data mentah biasanya tidak dalam kondisi yang optimal untuk langsung digunakan model yang disebabkan hal-hal seperti *dataset* masih mengandung *noise*, inkonsistensi, dan format yang tidak sesuai. Pada tahap *preprocessing* awal bertujuan untuk mengubah ukuran gambar, memberikan label gestur SIBI pada gambar, dan melakukan augmentasi dengan *color shifting* untuk memperbanyak variasi *dataset*.

3.2.1 Ubah Ukuran Gambar

Hal yang akan dilakukan setelah pengumpulan data yakni mengubah ukuran asli semua *dataset* gambar menjadi ukuran 320x320 piksel. Hal ini dilakukan untuk mengurangi beban komputasi selama proses pelatihan, memungkinkan proses yang lebih cepat dan penggunaan memori yang lebih efisien.

3.2.2 Label Objek

Ketika proses mengubah ukuran gambar telah selesai, setiap objek (gestur SIBI) dalam gambar akan diberi label menggunakan *labeling tools*, seperti LabelImg, dan hasil anotasi disimpan dalam format file *.xml* (*Extensible Markup Language*). Anotasi diperlukan untuk memberi tahu model lokasi dan kelas objek (gestur SIBI) dalam setiap gambar. File XML berisi informasi koordinat *bounding box* dan label kelas gestur SIBI yang sangat penting untuk proses pelatihan.

3.2.3 Augmentasi

Augmentasi dalam konteks pengolahan gambar atau data adalah teknik untuk memanipulasi data yang ada dengan cara seperti memutar, membalikkan, memotong, atau mengubah warna gambar (Shorten & Khoshgoftaar, 2019). Peneliti berencana akan melakukan *color shifting* atau mengubah warna gambar menjadi *greyscale*, *redscale*, *greenscale*, dan *bluescale*. Setiap gambar asli akan memiliki empat variasi tambahan dengan skema warna yang berbeda, menghasilkan total lima variasi gambar per gambar asli. Augmentasi gambar akan membantu meningkatkan jumlah *dataset* tanpa harus mengumpulkan gambar baru. Variasi warna dapat membantu model menjadi lebih *robust* terhadap perubahan pencahayaan dan warna dalam situasi nyata. Dengan memiliki lebih banyak variasi gambar, model juga lebih sulit untuk *overfit*, sehingga meningkatkan generalisasi model.

3.3 Rancangan Tahap Deteksi Objek

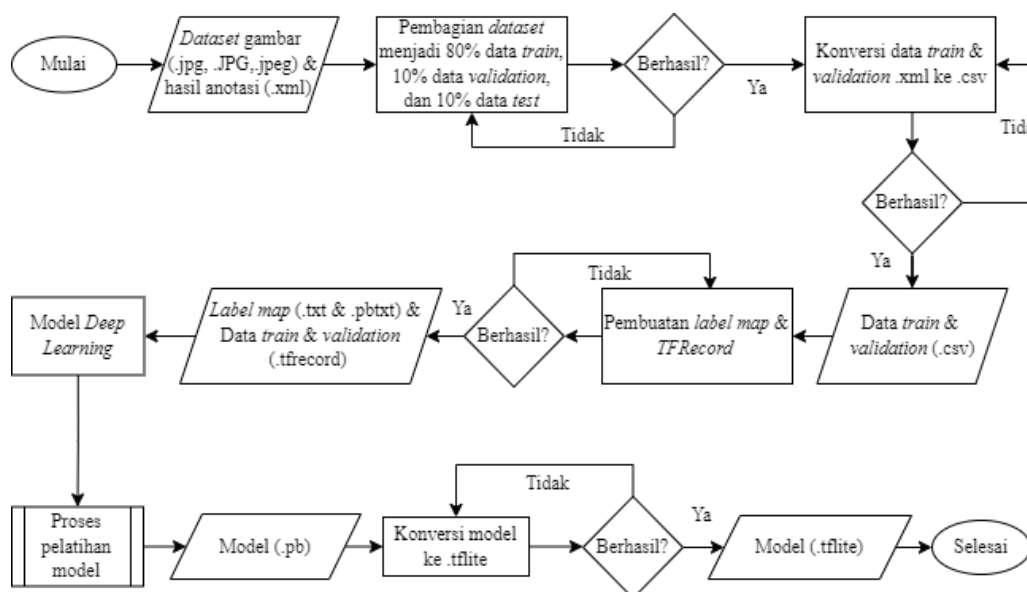
3.3.1 Menyiapkan Lingkungan Virtual

Deteksi objek menggunakan model arsitektur SSD *MobileNet V2 FPNLite* memerlukan persiapan lingkungan virtual seperti *Google Colaboratory* lalu menginstal *TensorFlow Object Detection API* beserta dependensinya. Proses ini melibatkan *cloning repository TensorFlow Models* dari *GitHub*, yang berisi implementasi *TensorFlow Object Detection API* serta berbagai model dan konfigurasi. Setelah *cloning repository*, perlu menginstal *Protoc, compiler* untuk *protocol buffers (protobuf)*, karena *TensorFlow Object Detection API*

menggunakan *protobuf* untuk mengkonfigurasi model dan melakukan pelatihan. Dengan persiapan ini, lingkungan virtual sudah siap digunakan untuk mengkonfigurasi dan melatih model deteksi objek menggunakan *SSD MobileNet V2 FPNLite* di *Google Colaboratory*.

3.3.2 Rancangan Proses Pelatihan dan Pemodelan

Penelitian ini akan menggunakan arsitektur Model *Pre-Trained SSD MobileNet V2 FPNLite* yang termasuk arsitektur model dalam Metode *Deep Learning* dan menggunakan *Framework TensorFlow Object Detection API* untuk mendeteksi gestur SIBI. Diagram *Flowchart* Rancangan Proses Pembuatan Model ditunjukkan pada Gambar 3.2 yang memberikan gambaran proses dari data setelah *data preprocessing* hingga menjadi model akhir yang disiapkan untuk mendeteksi gestur SIBI.



Gambar 3.2 *Flowchart* Rancangan Proses Pembuatan Model.

Lingkungan virtual dan *data preprocessing* awal telah dilakukan maka selanjutnya melakukan beberapa tahap *data preprocessing* sebelum data diproses dalam pemodelan dan pelatihan, yakni *dataset* akan dibagi menjadi tiga bagian: 80% *train*, 10% *validation*, dan 10% *test*. Data pada *train* digunakan untuk melatih model. Pada setiap langkah pelatihan, sekumpulan gambar pada data *train* dimasukkan ke dalam jaringan saraf. Jaringan ini memprediksi kelas dan lokasi objek dalam gambar. Data pada *validation* digunakan oleh algoritma pelatihan untuk memeriksa kemajuan pelatihan. Berbeda dengan sekumpulan gambar pada data *train*, gambar pada data *validation* hanya digunakan secara periodik selama pelatihan. Sedangkan, data pada *test* adalah sekumpulan gambar yang tidak pernah dilihat oleh jaringan saraf selama pelatihan sehingga dapat digunakan untuk pengujian model dan memeriksa seberapa akurat model tersebut. Jika proses pembagian *dataset* tidak berhasil, perlu dilakukan pengecekan kembali pada *script Python* yang digunakan untuk membagi *dataset*.

File .xml pada folder data *train* dan folder data *validation* akan dikonversi ke format .csv (*Comma-Separated Values*) karena format ini lebih mudah diproses untuk tahap selanjutnya (*TFRecord*). Jika konversi tidak berhasil, *script Python* yang digunakan untuk konversi .xml ke .csv perlu diperiksa kembali.

Label map dengan format .txt (*Text File*) dibuat dengan memasukkan nama-nama kelas gestur SIBI, dalam kasus ini ada 15 kelas. Kemudian, membuat format .tfrecord (*TensorFlow Record*) dari data *train* dan *validation* menjadi *train.tfrecord* dan *val.tfrecord*, selanjutnya membuat *label map* dengan format .pbtxt (*Protocol Buffer Text*). *TFRecord* berupa format file biner oleh

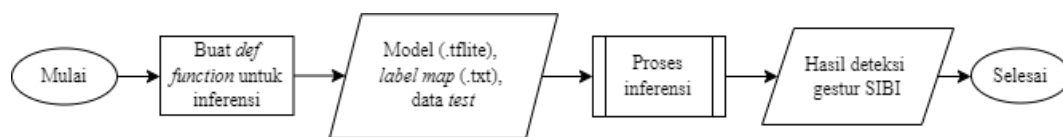
TensorFlow untuk mengoptimalkan pembacaan dan penulisan data secara efisien dalam *Framework TensorFlow*. Dalam konteks *TensorFlow*, file *PBTEXT* digunakan untuk menyimpan *label map* yang memetakan label numerik ke nama kelas. *Label map* dan *TFRecord* saling berkaitan karena *TFRecord* menyimpan data yang sudah dikonversi, sementara *label map* menyediakan informasi tentang kelas yang ada dalam *dataset*.

Tahap selanjutnya yakni, menyiapkan model yang akan digunakan dan konfigurasi pelatihan (*training*). Model *pre-trained SSD MobileNet V2 FPNLite*, model arsitektur dengan metode *Deep Learning*, digunakan sebagai dasar untuk deteksi objek pada penelitian ini. Model ini sebelumnya telah dilatih oleh *TensorFlow* menggunakan *COCO 2017 dataset*. Proses pemodelan ini melibatkan penyetelan ulang (*fine-tuning*). Kemudian, melakukan beberapa modifikasi pada parameter penting seperti jumlah langkah pelatihan (*steps*), jumlah gambar yang digunakan per langkah pelatihan (*batch size*), dan lain-lain. Proses pelatihan ditampilkan melalui *TensorBoard*, yang merupakan alat visualisasi untuk memantau proses pelatihan secara *real-time* dengan menampilkan grafik seperti grafik *loss*. *TensorBoard* membantu dalam analisis performa model dan penyesuaian parameter. Setelah pelatihan selesai, model disimpan dalam format *.pb* (*Protocol Buffer*). File *.pb* ini berisi arsitektur model dan bobot yang telah dilatih. Bobot yang telah dilatih ini adalah parameter dalam jaringan neural yang telah belajar selama pelatihan untuk meminimalkan *loss* dan meningkatkan akurasi prediksi.

Model yang disimpan dalam format .pb akan dikonversi ke format *TensorFlow Lite (.tflite)* agar dapat digunakan pada perangkat dengan sumber daya terbatas, selain itu penggunaan model *.tflite* memungkinkan peneliti untuk melakukan *testing* pada *interface Google Colab*. Jika proses konversi tidak berhasil, perlu diperiksa kembali langkah-langkah konversinya. Model *.tflite* yang dihasilkan kemudian akan diuji dan dievaluasi menggunakan data *test* pada tahap selanjutnya.

3.3.3 Rancangan Uji Coba Model

Tahap ini akan menguji apakah model yang telah dibuat sebelumnya berhasil mendeteksi dan mengklasifikasi objek SIBI dan bagaimana gambaran hasil deteksi tersebut dengan menampilkan beberapa gambar dari data *test*. Diagram *Flowchart* Rancangan Uji Coba Model ditunjukkan pada Gambar 3.3.



Gambar 3.3 *Flowchart* Rancangan Uji Coba Model.

Pertama membuat fungsi *def* untuk melakukan inferensi dengan model *TFLite* dan menampilkan hasilnya. Selanjutnya, menginput model *.tflite*, *label map* dengan file .txt, dan data *test* yang berisikan file gambar dan file .xml. Ketiga ini dibutuhkan untuk proses inferensi. Proses inferensi adalah tahap di mana model yang telah dilatih digunakan untuk membuat prediksi berdasarkan data baru yang tidak pernah dilihat selama pelatihan (data *test*). Dalam konteks ini, proses inferensi akan menampilkan hasil deteksi gestur SIBI dari beberapa gambar pada data *test* secara

acak. Hasil deteksi ini akan menunjukkan *bounding box* yang berisi label gestur beserta nilai kepercayaan dari prediksi tersebut.

3.3.4 Rencana Evaluasi Model

Evaluasi model dilakukan untuk mengetahui nilai rata-rata akurasi menggunakan mAP (*mean Average Precision*), yang merupakan metrik umum dalam penilaian kinerja model deteksi objek seperti *Faster R-CNN*, SSD, dan lain-lain. mAP mengukur seberapa baik model dalam mendeteksi dan mengklasifikasikan objek dalam gambar. Dalam penelitian ini, peneliti menggunakan standar metrik COCO yakni menghitung mAP @ 0.5:0.05:0.95. Peneliti akan menggunakan alat kalkulator mAP yang bersifat *open source* dari (Cartucho et al., 2018). Untuk tahap evaluasi, gambar dan data anotasi dari data *test* akan diduplikat atau disalin ke tempat baru. Data ini akan digunakan sebagai "*ground truth data*", yaitu data referensi yang akan dibandingkan dengan hasil deteksi model. Hasil deteksi model akan melibatkan pemanggilan fungsi *def* yang telah dibuat pada tahap uji coba sebelumnya dan digunakan untuk mendeteksi seluruh gambar yang ada pada data *test*. Hasil dari evaluasi ini berupa nilai akurasi mAP pada tiap *threshold* IoU dalam rentang 0.5 hingga 0.95 dengan kenaikan 0.05, kemudian nilai akurasi mAP masing-masing gestur SIBI (15 gestur), dan nilai akurasi mAP @ 0.5:0.05:0.95 sebagai nilai akurasi model secara keseluruhan.

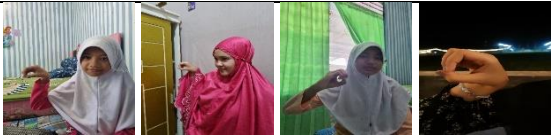
BAB IV

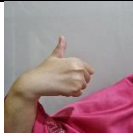
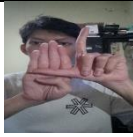
HASIL DAN PEMBAHASAN

4.1 Hasil Pengumpulan Data

Pengumpulan data menggunakan berbagai variasi sudut, kamera, latar belakang, jenis kelamin, maupun usia untuk *dataset* tiap 15 gestur SIBI (Huruf Vokal A, I, U, E, O dan 10 kata yakni Baik, Bangku, Bel, Dia, Meja, Pramuka, Sakit, Saya, Teman, dan Tugas). Total jumlah *dataset* awal yang digunakan peneliti adalah 3.000 gambar dengan masing-masing gestur memiliki 200 gambar. *Dataset* ini berupa *dataset* awal yang nantinya akan bertambah setelah proses augmentasi. Contoh hasil variasi *dataset* tiap gestur ditunjukkan pada Tabel 4.1.

Tabel 4.1 Contoh Hasil Variasi *Dataset* Tiap Gestur

No.	Bahasa Lisan	Contoh Hasil Variasi <i>Dataset</i> Gestur SIBI
1.	A	
2.	E	
3.	I	
4.	O	

No.	Bahasa Lisan	Contoh Hasil Variasi <i>Dataset</i> Gestur SIBI
5.	U	   
6.	Baik	   
7.	Bangku	   
8.	Bel	   
9.	Dia	   
10.	Meja	   
11.	Pramuka	   
12.	Sakit	   
13.	Saya	   

No.	Bahasa Lisan	Contoh Hasil Variasi <i>Dataset</i> Gestur SIBI
14.	Teman	
15.	Tugas	

4.2 Hasil Data *Preprocessing* Awal

4.2.1 Hasil Ubah Ukuran Gambar

Semua *dataset* gambar yang berukuran potret dengan rasio 19:6 dari berbagai kamera diubah menjadi ukuran 320x320 piksel. Proses ubah ukuran gambar menggunakan *script* untuk memudahkan *resize* dalam jumlah banyak dapat dilihat pada *Source Code* ubah ukuran gambar pada *Soucecode* 4.1. Pada tahap ini penulis juga mengubah penamaan file gambar agar menjadi lebih teratur. Hasil ubah ukuran gambar dapat dilihat pada Gambar 4.1.

***Sourcecode* 4.1** *Source Code* Ubah Ukuran Gambar

```
import os
from PIL import Image

# Yang perlu diubah
dataset_name = "Image_A"
width = 320
height = 320
# Folder input dengan gambar
input_folder = r'C:\Users\dell\Documents\SEMESTER
7\Penelitian_Skripsi\dataset\Vokal_U'
# Folder output untuk gambar yang sudah diresize dan direname
output_folder = r'C:\Users\dell\Documents\SEMESTER
7\Penelitian_Skripsi\dataset\Resize_Vokal_U'

# Fungsi untuk memeriksa dan memperbaiki rotasi gambar
```

```

def fix_image_rotation(image):
    if hasattr(image, '_getexif'):
        exif = image._getexif()
        if exif is not None:
            orientation = exif.get(0x0112)
            if orientation == 3:
                image = image.rotate(180, expand=True)
            elif orientation == 6:
                image = image.rotate(270, expand=True)
            elif orientation == 8:
                image = image.rotate(90, expand=True)
    return image

# Fungsi untuk melakukan resize gambar dan mengubah nama file
def resize_and_rename_images(input_folder, output_folder):
    file_counter = 1
    for filename in os.listdir(input_folder):
        if filename.endswith('.jpg'):
            input_image_path = os.path.join(input_folder,
            filename)
            new_filename = f"{dataset_name}_{file_counter}.jpg"
            output_image_path = os.path.join(output_folder,
            new_filename)

            # Buka gambar
            image = Image.open(input_image_path)

            # Perbaiki rotasi gambar jika diperlukan
            image = fix_image_rotation(image)

            # Resize gambar
            resized_image = image.resize((width, height))

            # Simpan gambar dengan nama dan ukuran baru
            resized_image.save(output_image_path)

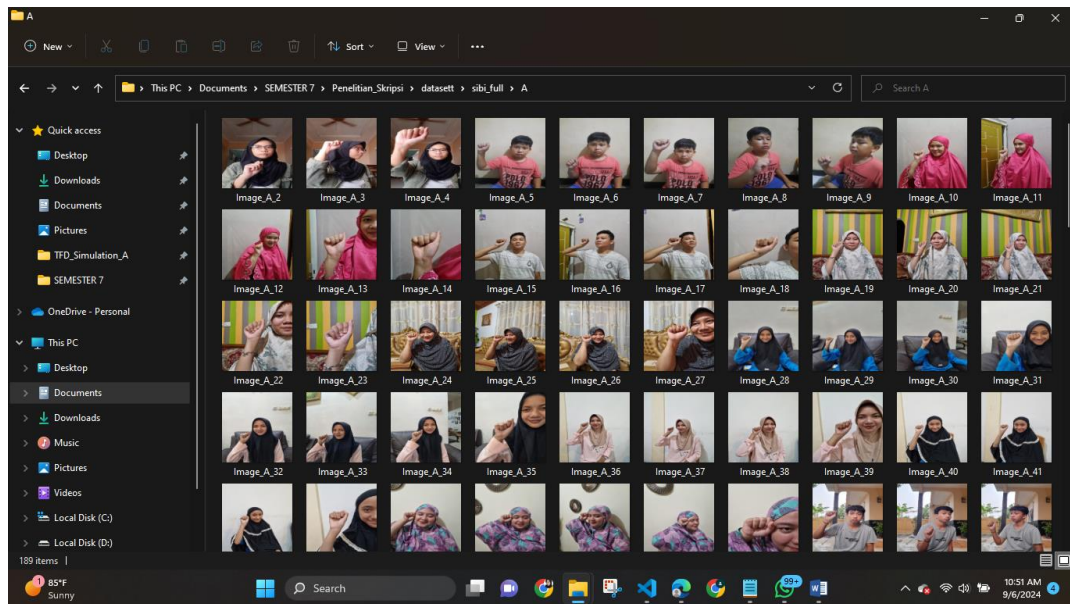
            print(f"File {filename} berhasil diubah menjadi
            {new_filename}")

            file_counter += 1

# Membuat folder output jika belum ada
os.makedirs(output_folder, exist_ok=True)

# Panggil fungsi untuk melakukan resize dan rename gambar
resize_and_rename_images(input_folder, output_folder)

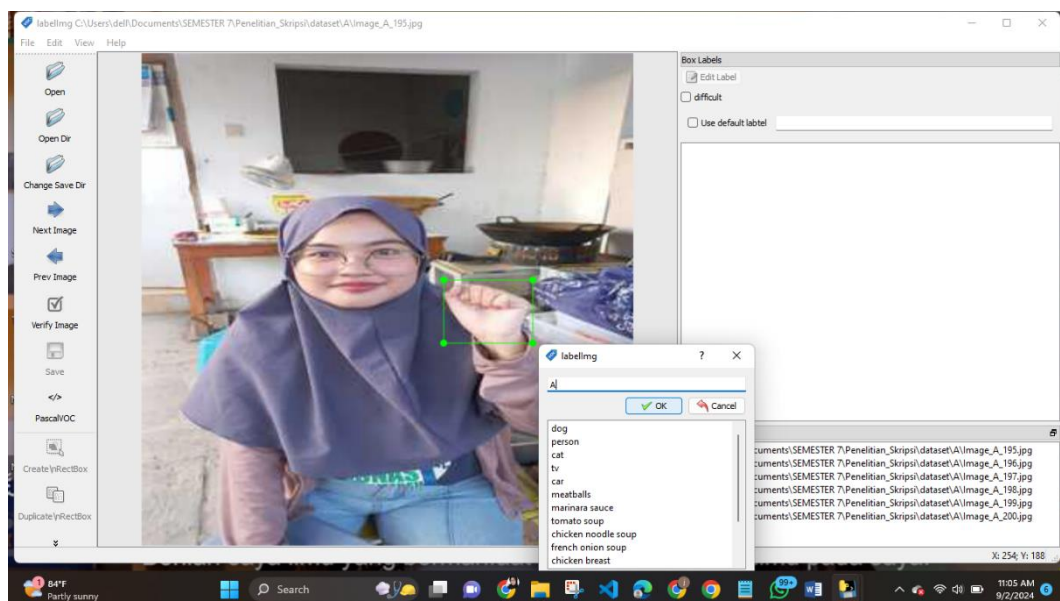
```



Gambar 4.1 Contoh Hasil Ubah Ukuran Gambar

4.2.2 Hasil Label Objek Gestur SIBI

Setiap objek (gestur SIBI) dalam gambar dianotasi dengan memberikan *bounding box* dan label kelas secara manual dengan menggunakan program *open source* Labelling (Label Studio community, 2015).



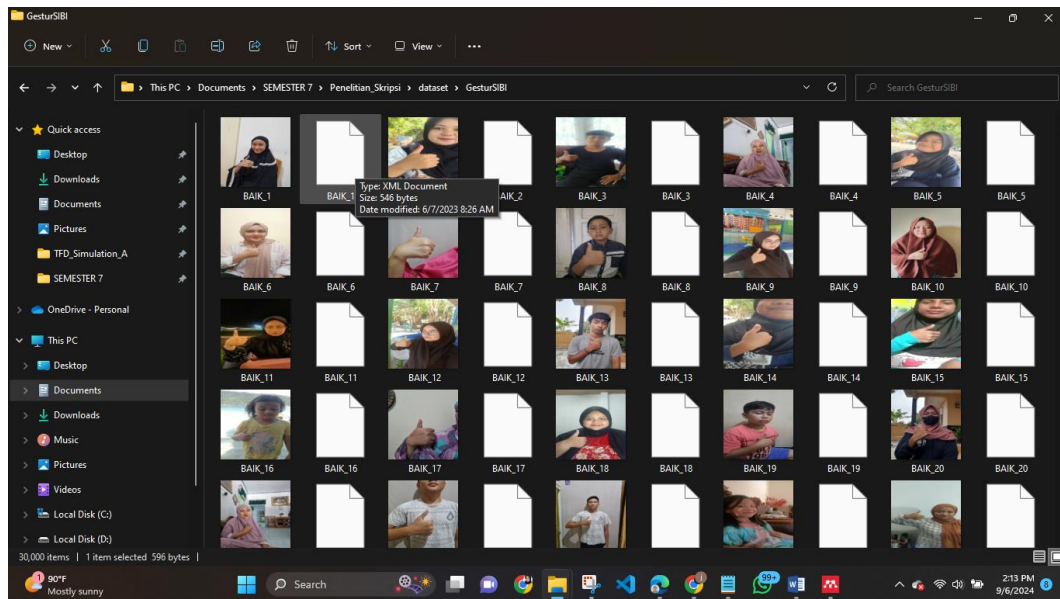
Gambar 4.2 Memberi Label Objek Gestur SIBI

Hasil anotasi tersimpan dalam format file .xml. Anotasi diperlukan untuk memberi tahu model lokasi dan kelas objek (gestur SIBI) dalam setiap gambar. File XML berisi informasi-informasi penting yang akan digunakan untuk proses pelatihan berupa label *class* gestur, *filename* gambar, *width* 320, *height* 320, dan titik koordinat *bounding box* label berupa *xmin*, *ymin*, *xmax*, dan *ymax*. Isi hasil anotasi label objek gestur SIBI dapat dilihat pada Gambar 4.3. Adapun gambaran hasil anotasi label objek gestur SIBI di dalam folder dapat dilihat pada Gambar 4.4.

```
<annotation>
  <folder>A</folder>
  <filename>Image_A_195.jpg</filename>
  <path>C:\Users\dell\Documents\SEMESTER 7\Penelitian_Skripsi\dataset
  \A\Image_A_195.jpg</path>
  <source>
    <database>Unknown</database>
  </source>
  <size>
    <width>320</width>
    <height>320</height>
    <depth>3</depth>
  </size>
  <segmented>0</segmented>
  <object>
    <name>A</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <difficult>0</difficult>
    <bndbox>
      <xmin>196</xmin>
      <ymin>147</ymin>
      <xmax>254</xmax>
      <ymax>188</ymax>
    </bndbox>
  </object>
</annotation>
```

Ln 1, Col 1 | 559 characters | 100% | Unix (LF) | UTF-8

Gambar 4.3 Isi Hasil Anotasi Label Objek Gestur SIBI



Gambar 4.4 Hasil Anotasi Label Objek Gestur SIBI Di Dalam Folder

4.2.3 Hasil Augmentasi

Proses ini menggunakan bantuan *script* untuk memudahkan proses augmentasi *color shifting* atau mengubah warna gambar tanpa harus melakukan proses *labeling* lagi. *Source Code* Augmentasi *Color Shifting* dapat dilihat pada *Sourcecode* 4.2. Setiap gambar asli memiliki empat variasi tambahan dengan skema warna *greyscale* (Gambar 4.5), *redscale* (Gambar 4.6), *greenscale* (Gambar 4.7), dan *bluescale* (Gambar 4.8). Hasil augmentasi di dalam folder dapat dilihat pada Gambar 4.9.

Sourcecode 4.2 Source Code Augmentasi Color Shifting

```
import glob
import os
import xml.etree.ElementTree as ET
import cv2
import shutil

current_directory = os.path.abspath(os.getcwd())
input_paths = glob.glob(r'C:\Users\dell\Documents\SEMESTER
7\Penelitian_Skripsi\dataset\Vokal_U\*')
output_folder = r'C:\Users\dell\Documents\SEMESTER
7\Penelitian_Skripsi\dataset\GesturSIBI'
```

```

def modifyXML(xml_path, file_name, file_path):
    tree = ET.parse(xml_path)
    root = tree.getroot()
    filename_element = root.find('filename')
    path_element = root.find('path')

    filename_element.text = file_name
    path_element.text = file_path
    tree.write(xml_path)

def copyXML(src, dest):
    return shutil.copy(src, dest)

def toGrey(file_path, output_folder, output_name):
    img = cv2.imread(file_path)
    gray_img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    cv2.imshow('Gray Images', gray_img)
    output_path = os.path.join(output_folder, f'{output_name}')
    cv2.imwrite(output_path, gray_img)
    cv2.waitKey(100)
    cv2.destroyAllWindows()

def toRed(file_path, output_folder, output_name):
    img = cv2.imread(file_path)
    red_img = img.copy()
    red_img[:, :, 0] = 0 #Mengatur saluran biru (blue) menjadi
0
    red_img[:, :, 1] = 0 # Mengatur saluran hijau (green) menjadi
0
    cv2.imshow('Red Scale Images', red_img)
    output_path = os.path.join(output_folder, f'{output_name}')
    cv2.imwrite(output_path, red_img)
    cv2.waitKey(100)
    cv2.destroyAllWindows()

def toGreen(file_path, output_folder, output_name):
    img = cv2.imread(file_path)
    red_img = img.copy()
    red_img[:, :, 0] = 0 # Mengatur saluran biru (blue) menjadi
0
    red_img[:, :, 2] = 0 # Mengatur saluran hijau (green) menjadi
0
    cv2.imshow('Red Scale Images', red_img)
    output_path = os.path.join(output_folder, f'{output_name}')
    cv2.imwrite(output_path, red_img)
    cv2.waitKey(100)
    cv2.destroyAllWindows()

def toBlue(file_path, output_folder, output_name):
    img = cv2.imread(file_path)
    red_img = img.copy()
    red_img[:, :, 1] = 0 # Mengatur saluran biru (blue) menjadi
0
    red_img[:, :, 2] = 0 # Mengatur saluran hijau (green) menjadi
0

```

```

cv2.imshow('Red Scale Images', red_img)
output_path = os.path.join(output_folder,f'{output_name}')
cv2.imwrite(output_path, red_img)
cv2.waitKey(100)
cv2.destroyAllWindows()

i = 0
for file_path in input_paths:
    isXMLModified = False
    output_names =
    [f'Image_U_Grey{i}',f'Image_U_Red{i}',f'Image_U_Green{i}',f'Image_U_Blue{i}']
    file = os.path.splitext(file_path)
    current_ext = file[1]
    print(current_ext)
    if current_ext == ".xml":
        # Start Parse XML
        for name in output_names:
            output_xml = os.path.join(output_folder,
f'{name}.xml')
            copyXML(file_path,output_xml)

modifyXML(output_xml,f'{name}.jpg',f'{output_folder}\\{name}.jpg'
)
        isXMLModified = True
    else:
        toGrey(file_path,
output_folder,f'{output_names[0]}{current_ext}')
        toRed(file_path,
output_folder,f'{output_names[1]}{current_ext}')
        toGreen(file_path,
output_folder,f'{output_names[2]}{current_ext}')
        toBlue(file_path,
output_folder,f'{output_names[3]}{current_ext}')

    if isXMLModified:
        i += 1

```



Gambar 4.5
Contoh *Greyscale*



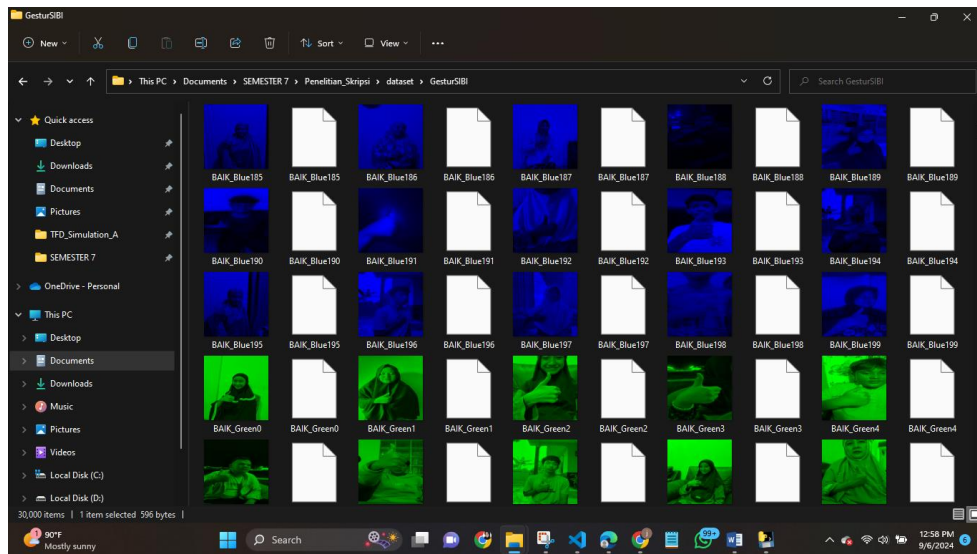
Gambar 4.6
Contoh *Redscale*



Gambar 4.7
Contoh *Greenscale*



Gambar 4.8
Contoh *Bluescale*



Gambar 4.9 Hasil Augmentasi Di Dalam Folder

4.3 Hasil Deteksi Objek

4.3.1 Hasil Menyiapkan Lingkungan Virtual

Lingkungan virtual yang disiapkan diantaranya *Google Colaboratory* untuk proses penelitian deteksi objek termasuk menginstal *TensorFlow Object Detection* API beserta dependensinya. Peneliti juga menyiapkan dan menghubungkan *Google Colaboratory* dengan *Google Drive* untuk menyimpan *input* dan *output* selama proses pembuatan deteksi objek gestur SIBI, karena jika hanya mengandalkan penyimpanan lokal di *Google Colab*, data akan hilang ketika sesi *running* dihentikan. Peneliti menggunakan referensi kode yang berasal dari *GitHub* yang bersifat *open-source repository* (Evan Juras, 2023), sebagai acuan kode untuk menyelesaikan penelitian ini.

4.3.2 Hasil Pelatihan dan Pemodelan

Penelitian menggunakan arsitektur Model *Pre-Trained SSD MobileNet V2 FPNLite* dan menggunakan *Framework TensorFlow Object Detection* API untuk

mendeteksi gestur SIBI dengan jumlah dataset 15.000 gambar dan 15.000 file xmlnya. *Dataset* dibagi menjadi 80% data *train* (12.000), 10% data *validation* (1.500), dan 10% data *test* (1.500). Detail pembagian *dataset* dapat dilihat pada Tabel 4.2.

Tabel 4.2 Detail Pembagian *Dataset*

No.	Gestur SIBI	Data <i>Train</i>	Data <i>Validation</i>	Data <i>Test</i>
1.	A	792	103	105
2.	E	805	93	102
3.	I	807	92	101
4.	O	801	94	105
5.	U	810	102	88
6.	Baik	785	104	111
7.	Bangku	809	104	87
8.	Bel	780	109	111
9.	Dia	833	90	77
10.	Meja	804	95	101
11.	Pramuka	793	104	103
12.	Sakit	781	108	111
13.	Saya	815	95	90
14.	Teman	793	101	106
15.	Tugas	792	106	102
Total		12.000	1.500	1.500

File .xml pada folder data *train* dan folder data *validation* dikonversi ke format .csv (*Comma-Separated Values*), hasilnya tersimpan dalam *google drive* dapat dilihat pada Gambar 4.10 *training_label.csv* dan Gambar 4.11

validation_labels.csv. File .csv menyimpan informasi-informasi penting dari xml berupa *filename*, *width*, *height*, *class*, *xmin*, *ymin*, *xmax*, dan *ymax*.

	A	B	C	D	E	F	G	H
	filename	width	height	class	xmin	ymin	xmax	ymax
1	BAH_1.jpg	320	320	baik	86	174	183	231
2	BAH_10.jpg	320	320	baik	93	183	188	236
3	BAH_103.jpg	320	320	baik	102	84	240	236
4	BAH_104.jpg	320	320	baik	103	86	240	240
5	BAH_105.jpg	320	320	baik	102	144	243	252
6	BAH_106.jpg	320	320	baik	82	148	135	211
7	BAH_107.jpg	320	320	baik	119	19	244	148
8	BAH_108.jpg	320	320	baik	125	198	180	280
9	BAH_109.jpg	320	320	baik	116	180	189	234
10	BAH_110.jpg	320	320	baik	144	130	226	236
11	BAH_111.jpg	320	320	baik	136	181	178	223
12	BAH_112.jpg	320	320	baik	85	188	139	213
13	BAH_113.jpg	320	320	baik	38	142	124	213
14	BAH_114.jpg	320	320	baik	104	180	158	223
15	BAH_115.jpg	320	320	baik	131	228	181	278
16	BAH_116.jpg	320	320	baik	81	187	130	220
17	BAH_117.jpg	320	320	baik	160	170	204	215
18	BAH_118.jpg	320	320	baik	80	149	140	203
19	BAH_119.jpg	320	320	baik	70	180	131	240
20	BAH_120.jpg	320	320	baik	120	180	187	209
21	BAH_121.jpg	320	320	baik	120	151	189	229
22	BAH_122.jpg	320	320	baik	175	203	227	249

Gambar 4.10 *training_label.csv*

	A	B	C	D	E	F	G	H
	filename	width	height	class	xmin	ymin	xmax	ymax
1	BAH_123.jpg	320	320	baik	112	158	178	233
2	BAH_124.jpg	320	320	baik	86	188	152	238
3	BAH_125.jpg	320	320	baik	80	187	148	238
4	BAH_126.jpg	320	320	baik	51	136	150	216
5	BAH_127.jpg	320	320	baik	103	218	150	273
6	BAH_128.jpg	320	320	baik	105	75	239	247
7	BAH_129.jpg	320	320	baik	156	143	255	224
8	BAH_130.jpg	320	320	baik	135	173	204	240
9	BAH_131.jpg	320	320	baik	122	124	230	215
10	BAH_132.jpg	320	320	baik	157	162	184	242
11	BAH_133.jpg	320	320	baik	120	194	186	248
12	BAH_134.jpg	320	320	baik	88	122	174	205
13	BAH_135.jpg	320	320	baik	118	192	189	277
14	BAH_136.jpg	320	320	baik	96	143	162	246
15	BAH_137.jpg	320	320	baik	135	180	173	200
16	BAH_138.jpg	320	320	baik	78	87	218	258
17	BAH_139.jpg	320	320	baik	100	121	231	257
18	BAH_140.jpg	320	320	baik	132	201	191	266
19	BAH_141.jpg	320	320	baik	73	108	130	217
20	BAH_142.jpg	320	320	baik	86	174	153	231
21	BAH_143.jpg	320	320	baik	93	183	168	238
22	BAH_144.jpg	320	320	baik	102	152	177	219

Gambar 4.11 *validation_labels.csv*

Hasil pembuatan *Label map* dengan format .txt dapat dilihat pada Gambar 4.12 dan format .pbtxt dapat dilihat pada Gambar 4.13. *Label map* berisikan nama-nama kelas 15 gestur SIBI, A, I, U, E, O, Baik, Bangku, Bel, Dia, Meja, Pramuka, Sakit, Saya, Teman, dan Tugas. Data *train* .csv dan data *validation* .csv juga telah berhasil diubah menjadi format .tfrecord, file data *train.tfrecord* dan *val.tfrecord* hasilnya tersimpan di dalam *Google Drive*. Proses persiapan data telah selesai maka penelitian siap dilanjutkan ke tahap pemodelan dan pelatihan.

```

File Edit View
A
E
I
U
O
baik
bangku
bel
dia
meja
pramuka
sakit
saya
teman
tugas

```

Gambar 4.12 *labelmap.txt*

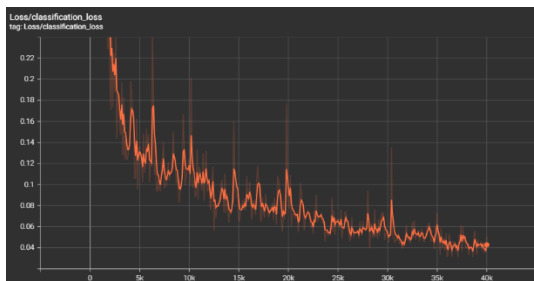
```

File Edit View
item {
  id: 1
  name: 'A'
}
item {
  id: 2
  name: 'E'
}
item {
  id: 3
  name: 'I'
}
item {
  id: 4
  name: 'U'
}
item {
  id: 5
  name: 'O'
}
item {
  id: 6
  name: 'A'
}

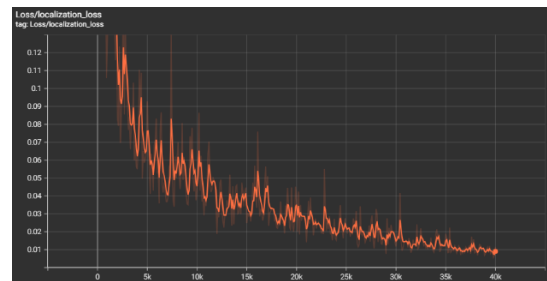
```

Gambar 4.13 *labelmap.pbtxt*

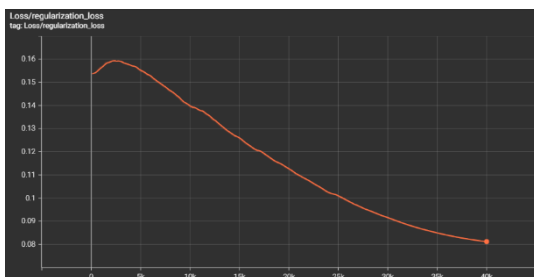
Proses pelatihan menggunakan Model *Pre-Trained SSD MobileNet V2 FPNLite* versi 320x320 dengan jumlah langkah pelatihan 40.000 *steps* dan jumlah langkah gambar yang digunakan per langkah pelatihan (*batch size*) adalah 16. Proses pelatihan tervisualisasi secara *realtime* melalui *TensorBoard*, hasilnya berupa grafik *classification_loss* (Grafik 4.1), *localization_loss* (Grafik 4.2), *regularization_loss* (Gambar 4.3), *total_loss* (Gambar 4.4), dan *learning_rate* (Gambar 4.5).



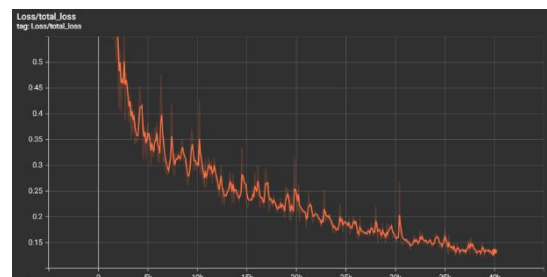
Grafik 4.1 *classification_loss*



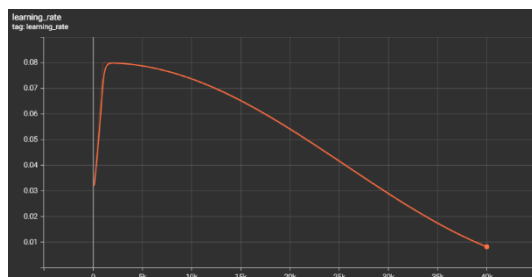
Grafik 4.2 *localization_loss*



Grafik 4.3 *regularization_loss*



Grafik 4.4 *total_loss*



Grafik 4.5 *learning_rate*

Pada *step* ke-40.000, *classification_loss* mengukur kesalahan model dalam mengklasifikasikan objek pada gambar, dengan nilai sebesar 0.05082879. *Localization_loss* menunjukkan kesalahan dalam penempatan *bounding box* atau posisi objek dalam gambar, tercatat sebesar 0.011469895. *Regularization_loss*, menunjukkan kesalahan model dalam menggeneralisasi pola untuk setiap objek, memiliki nilai sebesar 0.081098825. *Total_loss*, gabungan dari semua jenis *loss*, mencerminkan performa keseluruhan model, dengan nilai 0.14339751. Sementara itu, *learning_rate*, yang mengukur kecepatan pembaruan parameter selama pelatihan, berada pada angka 0.007943453.

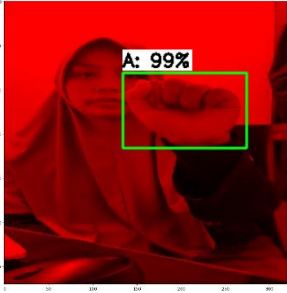
Proses pelatihan model berlangsung sekitar 2 jam 38 menit dan hasil model tersimpan dalam format .pb (*saved_model.pb*) selanjutnya dikonversi ke format *TensorFlow Lite*. File hasil konversi, yaitu *detect.tflite*, kini tersimpan di *Google Drive* dan akan digunakan untuk proses uji coba model.

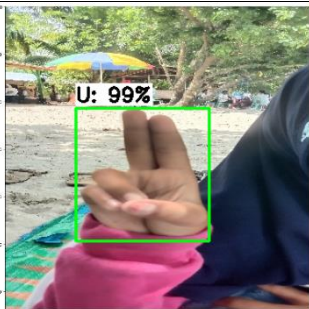
4.3.3 Hasil Uji Coba Model

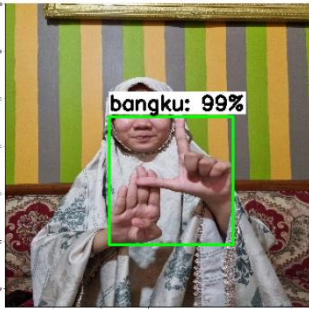
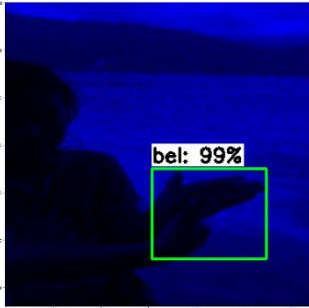
Fungsi *def tflite_detect_images* dibuat untuk proses inferensi menggunakan model *TFLite* dengan menginput model .*tflite*, *label map* dengan file .txt, dan data *test*. Proses inferensi pada tahap uji coba dilakukan dengan menampilkan hasil deteksi gestur SIBI dari beberapa gambar pada data *test* secara acak, dalam kasus ini 100 gambar. Hasil dari 100 gambar secara acak ini sudah memuat deteksi untuk setiap 15 gestur SIBI. Hasil uji coba model gestur SIBI dengan skor kepercayaan tertinggi dapat dilihat pada Tabel 4.3. Hasil uji coba menunjukkan model yang dibuat oleh peneliti sudah mampu mendeteksi 15 Gestur SIBI dengan berbagai kondisi tertentu yakni, model mampu mendeteksi pada pria maupun wanita, model

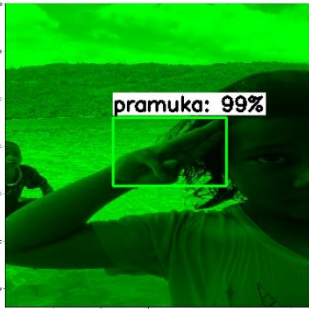
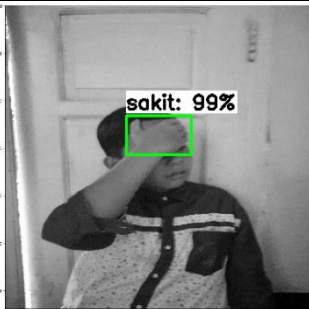
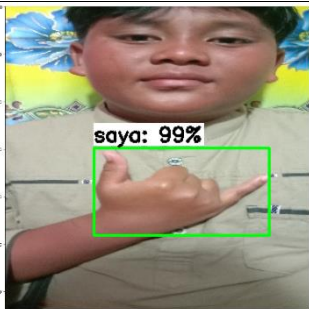
mampu mendeteksi pada anak-anak, remaja, dan dewasa, model mampu mendeteksi pada sudut depan, kiri, dan kanan, model mampu mendeteksi dalam kondisi *indoor* maupun *outdoor*, model mampu mendeteksi warna dengan intensitas gelap maupun terang ditandai dengan model mampu mendeteksi pada kondisi *bluescale*, *redscale*, *greenscale*, *greyscale*, dan RGB, model mampu mendeteksi gestur pada berbagai latar belakang yang memiliki banyak atribut atau bervariasi maupun polos, dan model mampu mendeteksi gambar yang diambil dari kamera *device* yang beragam.

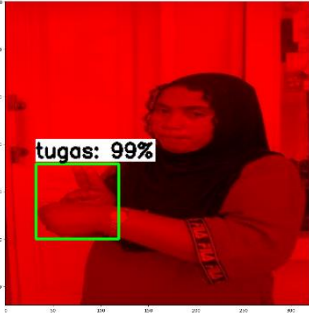
Tabel 4.3 Hasil Uji Coba Model Gestur SIBI dengan Skor Kepercayaan Tertinggi

No.	Hasil Uji Coba Deteksi	Ket. Gambar	Ket. Deteksi
1.		❖ Gestur A ❖ Wanita remaja ❖ Sudut depan ❖ <i>Indoor</i> ❖ <i>Redscale</i>	Gestur A terdeteksi dengan benar dengan skor kepercayaan 99%
2.		❖ Gestur E ❖ Wanita remaja ❖ Sudut ke kiri ❖ <i>Indoor</i> ❖ <i>Greenscale</i>	Gestur E terdeteksi dengan benar dengan skor kepercayaan 99%

No.	Hasil Uji Coba Deteksi	Ket. Gambar	Ket. Deteksi
3.		❖ Gestur I ❖ Wanita dewasa ❖ Sudut depan ❖ <i>Indoor</i> ❖ RGB	Gestur I terdeteksi dengan benar dengan skor kepercayaan 99%
4.		❖ Gestur O ❖ Pria remaja ❖ Sudut ke kanan ❖ <i>Indoor</i> ❖ <i>Redscale</i>	Gestur O terdeteksi dengan benar dengan skor kepercayaan 99%
5.		❖ Gestur U ❖ Anak perempuan ❖ Sudut depan ❖ <i>Outdoor</i> ❖ RGB	Gestur U terdeteksi dengan benar dengan skor kepercayaan 99%
6.		❖ Gestur Baik ❖ Wanita dewasa ❖ Sudut depan ❖ <i>Indoor</i> ❖ <i>Greenscale</i>	Gestur Baik terdeteksi dengan benar dengan skor kepercayaan 99%

No.	Hasil Uji Coba Deteksi	Ket. Gambar	Ket. Deteksi
7.		❖ Gestur Bangku ❖ Wanita remaja ❖ Sudut depan ❖ <i>Indoor</i> ❖ <i>RGB</i>	Gestur Bangku terdeteksi dengan benar dengan skor kepercayaan 99%
8.		❖ Gestur Bel ❖ Anak perempuan ❖ Sudut depan ❖ <i>Outdoor</i> ❖ <i>Bluescale</i>	Gestur Bel terdeteksi dengan benar dengan skor kepercayaan 99%
9.		❖ Gestur Dia ❖ Wanita Remaja ❖ Sudut ke kanan ❖ <i>Indoor</i> ❖ <i>Bluescale</i>	Gestur Dia terdeteksi dengan benar dengan skor kepercayaan 99%
10.		❖ Gestur Meja ❖ Pria remaja ❖ Sudut depan ❖ <i>Indoor</i> ❖ <i>Greyscale</i>	Gestur Meja terdeteksi dengan benar dengan skor kepercayaan 97%

No.	Hasil Uji Coba Deteksi	Ket. Gambar	Ket. Deteksi
11.		❖ Gestur Pramuka ❖ Anak perempuan ❖ Sudut depan ❖ <i>Outdoor</i> ❖ <i>Greyscale</i>	Gestur Pramuka terdeteksi dengan benar dengan skor kepercayaan 99%
12.		❖ Gestur Sakit ❖ Anak laki-laki ❖ Sudut ke kiri ❖ <i>Indoor</i> ❖ <i>Greyscale</i>	Gestur Sakit terdeteksi dengan benar dengan skor kepercayaan 99%
13.		❖ Gestur Saya ❖ Pria remaja ❖ Sudut depan ❖ <i>Indoor</i> ❖ <i>RGB</i>	Gestur Saya terdeteksi dengan benar dengan skor kepercayaan 99%
14.		❖ Gestur Teman ❖ Anak perempuan ❖ Sudut ke kiri ❖ <i>Indoor</i> ❖ <i>RGB</i>	Gestur Teman terdeteksi dengan benar dengan skor kepercayaan 99%

No.	Hasil Uji Coba Deteksi	Ket. Gambar	Ket. Deteksi
15.		❖ Gestur Tugas ❖ Wanita dewasa ❖ Sudut ke kanan ❖ <i>Outdoor</i> ❖ <i>Redscale</i>	Gestur Tugas terdeteksi dengan benar dengan skor kepercayaan 99%

4.3.4 Hasil Evaluasi Model

Tahap evaluasi model melibatkan deteksi seluruh gambar yang ada pada data *test*, 1.500 gambar, dengan memanggil fungsi *def tf_lite_detect_images* dengan tidak menampilkan gambar inferensi pada *Google Colaboratory*. Hal ini disebabkan *buffer* memiliki batasan dalam menampilkan *output* gambar pada *interface Google Colab*.

Hasil evaluasi menunjukkan nilai $mAP_{IoU0.50}$ 99.93%, $mAP_{IoU0.55}$ 99.93%, $mAP_{IoU0.60}$ 99.93%, $mAP_{IoU0.65}$ 99.93%, $mAP_{IoU0.70}$ 99.93%, $mAP_{IoU0.75}$ 99.66%, $mAP_{IoU0.80}$ 99.35%, $mAP_{IoU0.85}$ 97.60%, $mAP_{IoU0.90}$ 81.07%, $mAP_{IoU0.95}$ 21.95% dan nilai mAP kelas dari total AP kelas tertentu dalam rentang IoU 0.5:0.05:0.9.5 yakni, mAP_A 88.76%, mAP_E 89.90%, mAP_I 91.23%, mAP_O 90.22%, mAP_U 87.97%, mAP_{Baik} 90.65%, mAP_{Bangku} 95.42%, mAP_{Bel} 93.31%, mAP_{Dia} 89.53%, mAP_{Meja} 83.72%, $mAP_{Pramuka}$ 88.59%, mAP_{Sakit} 90.10%, mAP_{Saya} 88.90%, mAP_{Teman} 91.00%, mAP_{Tugas} 89.63%, dimana nilai mAP_{Bangku} menunjukkan nilai tertinggi yakni 95.42% dan mAP_{Meja} menunjukkan nilai terendah yakni 83.72%.

Hasil keseluruhan model $mAP @ 0.5:0.05:0.95$ menunjukkan nilai akurasi model deteksi gestur SIBI yakni 89,93%. Hasil evaluasi model dapat dilihat pada

Tabel 4.4. Hasil nilai akurasi dalam penelitian ini mendapatkan nilai akurasi model yang lebih baik jika dibandingkan dengan nilai akurasi model deteksi SIBI yang dikembangkan oleh (Akbar Mahardi Hidayatullah, 2022), yang sebelumnya telah dipaparkan pada bagian latar belakang. Penelitian tersebut mendeteksi 6 Gestur SIBI (Saya, Dia, Kamu, Cinta, Maaf, dan Sedih) menggunakan Arsitektur Model *Pre-Trained SSD MobileNet V2* dengan jumlah *dataset* sebanyak 600 gambar dan dilatih menggunakan 20.000 *steps* yang menunjukkan hasil nilai akurasi 86,6%, sedangkan dalam penelitian penulis mendeteksi 15 Gestur SIBI (Huruf Vokal A, I, U, E, O dan 10 kata yakni Baik, Bangku, Bel, Dia, Meja, Pramuka, Sakit, Saya, Teman, dan Tugas) menggunakan Arsitektur Model *Pre-Trained SSD MobileNet V2 FPNLite* dengan jumlah *dataset* sebanyak 3.000 gambar dan dilatih menggunakan 40.000 *steps* yang menunjukkan hasil lebih baik yakni 89,93%. Hasil kontribusi penulis dalam penelitian diharapkan menambah pengetahuan sistem deteksi serta dapat membantu melestarikan dan mempopulerkan SIBI sebagai bahasa isyarat nasional Indonesia

Tabel 4.4 Hasil Evaluasi Model

No.	Gestur SIBI	<i>Average Precision (AP) Pada Threshold Intersection over Union (IoU)</i>										mAP Gestur @ 0.5:0.05:0.95
		0.50	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95	
1.	A	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	97.88%	95.49%	73.59%	21.81%	88.76%
2.	E	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	97.58%	86.84%	14.62%	89.90%
3.	I	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	98.90%	92.59%	20.80%	91.23%
4.	O	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	97.50%	71.02%	11.17%	90.22%
5.	U	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	84.24%	18.00%	87.97%
6.	Baik	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	81.25%	25.28%	90.65%
7.	Bangku	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	97.98%	56.26%	95.42%
8.	Bel	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	88.39%	44.73%	93.31%
9.	Dia	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	98.48%	98.48%	85.26%	13.06%	89.53%
10.	Meja	99.01%	99.01%	99.01%	99.01%	99.01%	99.01%	96.74%	88.81%	51.29%	6.26%	83.72%
11.	Pramuka	100.00%	100.00%	100.00%	100.00%	100.00%	97.09%	97.09%	97.09%	75.91%	18.75%	88.59%
12.	Sakit	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	98.01%	83.94%	19.07%	90.10%
13.	Saya	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	98.16%	76.37%	14.45%	88.90%

No.	Gestur SIBI	<i>Average Precision (AP) Pada Threshold Intersection over Union (IoU)</i>										mAP Gestur @ 0.5:0.05:0.95
		0.50	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95	
14.	Teman	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	98.77%	85.82%	25.40%	91.00%
15.	Tugas	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	95.20%	81.49%	19.65%	89.63%
mAP IoU		99.93%	99.93	99.93%	99.93%	99.93%	99.66%	99.35%	97.60%	81.07%	21.95%	mAP @ 0.50:0.05:0.95 (Overall) 89.93%

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Kesimpulan dari penelitian ini yakni:

1. Deteksi gestur SIBI menggunakan Metode *Deep Learning* dengan *Framework TensorFlow Object Detection* dengan Model *Pre-Trained SSD MobileNet V2 FPNLite* dengan total jumlah 15.000 gambar *dataset*, yang dibagi menjadi 80% data *train*, 10% data *validation*, dan 10% data *test* yang setiap gambar data memiliki variasi sudut, kamera, warna, latar belakang, jenis kelamin, maupun usia, yang dilatih menggunakan 40.000 *steps* dan 16 *batch size* menunjukkan hasil nilai *loss* 0.1434. Hasil uji coba menunjukkan model dalam penelitian ini sudah mampu mendeteksi 15 Gestur SIBI (Huruf Vokal A, I, U, E, O dan 10 kata yakni Baik, Bangku, Bel, Dia, Meja, Pramuka, Sakit, Saya, Teman, dan Tugas) dengan berbagai variasi kondisi tertentu.
2. Hasil evaluasi $mAP @ 0.5:0.05:0.95$ menunjukkan nilai akurasi model deteksi gestur SIBI sebesar 89,93%, dengan nilai mAP kelas objek tertinggi hingga terendah yakni mAP_{Bangku} 95.42%, mAP_{Bel} 93.31%, mAP_I 91.23%, mAP_{Teman} 91.00%, mAP_{Baik} 90.65%, mAP_O 90.22%, mAP_{Sakit} 90.10%, mAP_E 89.90%, mAP_{Tugas} 89.63%, mAP_{Dia} 89.53%, mAP_{Saya} 88.90%, $mAP_{Pramuka}$ 88.59%, mAP_U 87.97%, dan mAP_{Meja} 83.72%.

5.2 Saran

Saran-saran yang dapat diberikan untuk peneliti selanjutnya:

1. Distribusi *dataset* yang lebih proporsional: Mengoptimalkan pembagian *dataset* antara data *train*, *validation*, dan *test* dengan mempertimbangkan keseimbangan jumlah di setiap kelas gestur dapat membantu meningkatkan kemampuan model dalam menggeneralisasi pola yang ada.
2. Implementasi *real-time*: Mengembangkan model deteksi gestur untuk diimplementasikan pada *platform mobile* dan diuji secara *real-time* akan memberikan wawasan tentang kinerja model dalam situasi dunia nyata. Pengujian di lingkungan yang dinamis akan membantu melihat bagaimana model beradaptasi terhadap variasi seperti cahaya, latar belakang, dan sudut pengambilan deteksi.
3. Penambahan *dataset*: Mengembangkan *dataset* dengan lebih banyak variasi gestur dan kondisi lingkungan dapat meningkatkan performa model, termasuk menambahkan gestur non-vokal atau variasi gerakan lain yang relevan untuk memperluas cakupan deteksi model.

DAFTAR REFERENSI

- Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., ... Zheng, X. (2016). *Tensorflow: Large-Scale Machine Learning On Heterogeneous Distributed Systems*. [Http://Arxiv.Org/Abs/1603.04467](http://Arxiv.Org/Abs/1603.04467)
- Achmar Fahrurrozi, Desti Rimirasih, & Nurma Nugraha. (2022). *Sistem Klasifikasi Kayu Berbasis Citra Tekstur Menggunakan Machine Learning*. Uwais Inspirasi Indonesia. [Https://Www.Google.Co.Id/Books/Edition/Sistem_Klasifikasi_Kayu_Berbas is_Citra_T/5pykeaaaqbaj?Hl=Id&Gbpv=1&Dq=Objek+Deteksi&Pg=Pa80&Printsec=Frontcover](https://Www.Google.Co.Id/Books/Edition/Sistem_Klasifikasi_Kayu_Berbas_is_Citra_T/5pykeaaaqbaj?Hl=Id&Gbpv=1&Dq=Objek+Deteksi&Pg=Pa80&Printsec=Frontcover)
- Akbar Mahardi Hidayatullah. (2022). *Sistem Deteksi Simbol Pada Sipi (Sistem Isyarat Bahasa Indonesia) Secara Realtime Menggunakan Mobilenet-Ssd*. Universitas Dinamika.
- Apendi, S., & Paryasto, M. W. (2023). *Deteksi Bahasa Isyarat Sistem Isyarat Bahasa Indonesia Menggunakan Metode Single Shot Multibox Detector* (Vol. 10, Issue 1).
- Badan Pusat Statistik. (2023). *Buku-I-Analisis-Tematik-Kependudukan-Indonesia--Fertilitas-Remaja--Kematian-Maternal--Kematian-Bayi--Dan-Penyandang-Disabilitas-*. Badan Pusat Statistik.
- Bagus, M., Bakti, S., & Pranoto, Y. M. (2019). *Pengenalan Angka Sistem Isyarat Bahasa Indonesia Dengan Menggunakan Metode Convolutional Neural Network*.
- Cartucho, J., Ventura, R., & Veloso, M. (2018). *Robust Object Recognition Through Symbiotic Deep Learning In Mobile Robots**. [Http://Cocodataset.Org](http://Cocodataset.Org)
- Chiu, Y.-C., Tsai, C.-Y., Ruan, D., Shen, G.-Y., & Lee, T.-T. (2020). Mobilenet-Ssdv2: An Improved Object Detection Model For Embedded Systems. In *2020 International Conference On System Science And Engineering (Icsse)*.
- Dewi Ratih Rapisa. (2021). *Sistem Komunikasi Anak Dengan Hambatan Pendengaran*. Cv Budi Utama. [Https://Www.Google.Co.Id/Books/Edition/Sistem_Komunikasi_Anak_Dengan_Hambatan_P/Unbbeaaaqbaj?Hl=Id&Gbpv=1&Dq=Tuna+Rungu+Merupakan&Pg=Pa69&Printsec=Frontcoverhttps://Www.Google.Co.Id/Books/Edition/Sistem_Komunikasi_Anak_Dengan_Hambatan_P/Unbbeaaaqbaj?Hl=Id&Gbpv=1&Dq=Tuna+Rungu+Merupakan&Pg=Pa69&Printsec=Frontcover](https://Www.Google.Co.Id/Books/Edition/Sistem_Komunikasi_Anak_Dengan_Hambatan_P/Unbbeaaaqbaj?Hl=Id&Gbpv=1&Dq=Tuna+Rungu+Merupakan&Pg=Pa69&Printsec=Frontcoverhttps://Www.Google.Co.Id/Books/Edition/Sistem_Komunikasi_Anak_Dengan_Hambatan_P/Unbbeaaaqbaj?Hl=Id&Gbpv=1&Dq=Tuna+Rungu+Merupakan&Pg=Pa69&Printsec=Frontcover)
- Dionisius, R. A. S., Finsensia Riti, Y., & Christian, N. (2024). Perbandingan Performa Model Ssd Mobilenet V2 Dan Fpnlite Dalam Deteksi Helm Pengendara Sepeda Motor Performance Comparison Of Ssd Mobilenet V2 And Fpnlite In Motorcycle Rider Helmet Detection. *Jurnal Sistem Komputer*, 13(1), 2020. [Https://Doi.Org/10.34010/Komputika.V13i1.10333](https://Doi.Org/10.34010/Komputika.V13i1.10333)
- Emir Fakhri Muttaqiy. (2024). *Deteksi Kerusakan Jalan Menggunakan Metode Deep Learning Ssd Mobilenet V2 Fpnlite*. Institut Teknologi Sepuluh Nopember.

- Evan Juras, E. T. C. (2023). *Tensorflow Lite Object Detection On Android And Raspberry Pi*. <https://Github.Com/Edjeelectronics/Tensorflow-Lite-Object-Detection-On-Android-And-Raspberry-Pi>
- Ferdian Nursulistio. (2022). *Deteksi Objek Masker Menggunakan Object Detection Api Dan Tensorflow Lite Model Maker*. Universitas Islam Indonesia.
- Francois Chollet. (2018). *Deep Learning With Python* (1st Ed.). Manning Publication Co.
- Hans Petter Langtangen. (2008). Python Scripting For Computational Science. In *Python Scripting For Computational Science*. Springer Berlin Heidelberg. <https://doi.org/10.1007/978-3-540-73916-6>
- Hayati, L. N., Handayani, A. N., Irianto, W. S. G., Asmara, R. A., Indra, D., & Fahmi, M. (2023). Classifying Bisindo Alphabet Using Tensorflow Object Detection Api. *Ilkom Jurnal Ilmiah*, 15(2), 358–364. <https://doi.org/10.33096/Ilkom.V15i2.1692.358-364>
- Ian Goodfellow, Yoshua Bengio, & Aaron Courville. (2016). *Deep Learning*. Mit Press. <http://www.deeplearningbook.org>
- Ilham Rizaldy Widy Putra. (2021). *Sistem Deteksi Simbol Pada Sibi (Sistem Isyarat Bahasa Indonesia) Menggunakan Convolutional Neural Network*.
- Joan, D., Vincent, V., Daniel, K. J., Achmad, S., & Sutoyo, R. (2023). Bisindo Hand-Sign Detection Using Transfer Learning. *8th International Conference On Recent Advances And Innovations In Engineering: Empowering Computing, Analytics, And Engineering Through Digital Innovation, Icaie 2023*. <https://doi.org/10.1109/Icaie59459.2023.10468194>
- Kahlil, Muhammad Rizky Munggaran, Laksono Kurnianguro, Adhiguna Mahendra, Nona Zarima, Fina Noviantika, & Alifya Febriana. (2023). *Computer Vision Berbasis Deep Learning Untuk Aplikasi Pertanian: Teori Dan Praktik* (Nana Diana, Ed.). Syiah Kuala University Press. https://www.google.co.id/books/edition/Computer_Vision_Berbasis_Deep_Learning_Un/L6bkeaaaqbaj?hl=id&gbpv=1&dq=MobileNet+adalah&pg=Pa100&printsec=frontcover
- Kais Ali Syafii Fatwari. (2022). *Model Pengenalan Ekspresi Antusiasme Wajah Dengan Menggunakan Arsitektur Ssd Mobilenet V2 FpnLite*. Universitas Amikom Yogyakarta.
- Label Studio Community. (2015). *Label Studio Is A Modern, Multi-Modal Data Annotation Tool*. <https://github.com/Humansignal/labelimg?tab=MIT-1-OV-file>
- Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2016). *Feature Pyramid Networks For Object Detection*. <http://arxiv.org/abs/1612.03144>
- Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C. L., & Dollár, P. (2015). *Microsoft Coco: Common Objects In Context*.
- Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2015). *Ssd: Single Shot Multibox Detector*. https://doi.org/10.1007/978-3-319-46448-0_2

- Muhammad Azka Imaddudin, Ida Wahidah Hamzah, & Sri Astuti. (2022). Simulasi Penerjemah Sibi (Sistem Isyarat Bahasa Indonesia) Menggunakan Tensorflow Dan Convolutional Neural Network (Cnn) Simulation Of Sibi Translator (Sistem Isyarat Bahasa Indonesia) Using Tensorflow And Convolutional Neural Network (Cnn). *E-Proceeding Of Engineering*, 8.
- Pasal 1 Ayat 1 Uu Nomor 8 Tahun 2016 (2016).
- Rondiyanto. (2022). *Implementasi Framework Tensorflow Object Detection Dalam Mendeteksi Gestur Bahasa Isyarat*.
- Saiful Bahri, I. Z., Saon, S., Mahamad, A. K., Isa, K., Fadlilah, U., Ahmadon, M. A. Bin, & Yamaguchi, S. (2023). Interpretation Of Bahasa Isyarat Malaysia (Bim) Using Ssd-Mobilenet-V2 Fpnlite And Coco Map. *Information (Switzerland)*, 14(6). <https://doi.org/10.3390/info14060319>
- Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). *Mobilenetv2: Inverted Residuals And Linear Bottlenecks*. <http://arxiv.org/abs/1801.04381>
- Sari, I., & Altiarika, E. (2023). Sistem Pengembangan Bahasa Isyarat Untuk Berkomunikasi Dengan Penyandang Disabilitas (Tunarungu). In *Journal Of Information Technology And Society (Jits)* (Vol. 1, Issue 1). <https://jits.unmuhbabel.ac.id/>
- Shorten, C., & Khoshgoftaar, T. M. (2019). A Survey On Image Data Augmentation For Deep Learning. *Journal Of Big Data*, 6(1). <https://doi.org/10.1186/s40537-019-0197-0>
- Sri Maulida. (2023). *Analisis Akurasi Pada Simbol Abjad Sistem Isyarat Bahasa Indonesia (Sibi) Dengan Menggunakan Metode Cnn Dan Yolo (You Only Look Once)*.
- Suryanah. (1996). *Keperawatan Anak Untuk Siswa Spk* (Ester Diana A.). Penerbit Buku Kedokteran Egc. https://www.google.co.id/books/edition/Keperawatan_Anak_Untuk_Siswa_Spk/I7if6jaezwgc?hl=id&gbpv=1

LAMPIRAN

Google Colaboratory :

https://colab.research.google.com/drive/1d-JFYJhVWqV2zX-dqVtXBY_qCm2GigQc?usp=sharing

Google Drive :

<https://drive.google.com/drive/folders/1TZu5G8HflNC8zrJFzy76XtzFtUpscJ1e?usp=sharing>

Output perhitungan mAP dapat dilihat pada folder mAP bagian output :

https://drive.google.com/drive/folders/1x8Xj3v9fcecB2bsDmKai1Uee1vQl9hs?usp=drive_link