



SISTEM DETEKSI SIMBOL PADA SIBI (SISTEM ISYARAT BAHASA INDONESIA) SECARA *REALTIME* MENGGUNAKAN MOBILENET-SSD



Oleh:

AKBAR MAHARDI HIDAYAHTULLAH

18410200048

FAKULTAS TEKNOLOGI DAN INFORMATIKA

UNIVERSITAS DINAMIKA

2022

SISTEM DETEKSI SIMBOL PADA SIBI (SISTEM ISYARAT BAHASA INDONESIA) SECARA *REALTIME* MENGGUNAKAN MOBILENET-SSD

TUGAS AKHIR

**Diajukan sebagai salah satu syarat untuk menyelesaikan
Program Sarjana Teknik**



UNIVERSITAS
Dinamika

Disusun Oleh:

Nama	: Akbar Mahardi Hidayahullah
NIM	: 18410200048
Program Studi	: S1 Teknik Komputer

FAKULTAS TEKNOLOGI DAN INFORMATIKA

UNIVERSITAS DINAMIKA

2022

TUGAS AKHIR

SISTEM DETEKSI SIMBOL PADA SIBI (SISTEM ISYARAT BAHASA INDONESIA) SECARA *REALTIME* MENGGUNAKAN MOBILENET-SSD

Dipersiapkan dan disusun oleh:

Akbar Mahardi Hidayahullah

NIM : 18410200048

Telah diperiksa, dibahas dan disetujui oleh Dewan Pembahasan

Pada: 05 Juli 2022

Susunan Dewan Pembahasan

Pembimbing:

- I. Heri Pratikno, M.T., MTCNA., MTCRE.
NIDN: 0722087701
- II. Yosefine Triwidayastuti, M.T.
NIDN: 0729038504


Digitally signed
by Heri Pratikno
Date: 2022.07.15
16:54:55 +07'00'


Digitally signed by
Universitas Dinamika
Date: 2022.07.15
20:32:25 +07'00'

Pembahasan:

- I. Pauladie Susanto, S.Kom., M.T.
NIDN: 0721047201


cn=Pauladie Susanto, o=FTI
Undika, ou=Prodi STI TK,
email=pauladie@dinamika.ac.id,
c=ID
2022.07.18 12:55:24 +07'00'

Tugas Akhir ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Sarjana


Digitally signed by
Universitas Dinamika
Date: 2022.07.19
16:03:30 +07'00'

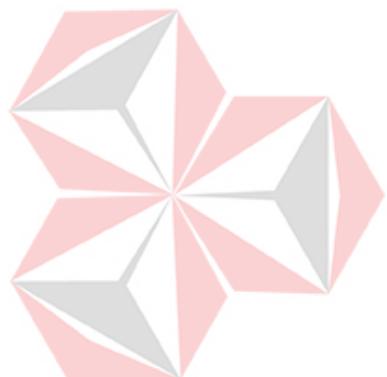
Tri Sagirani, S.Kom., M.MT.

NIDN: 0731017601

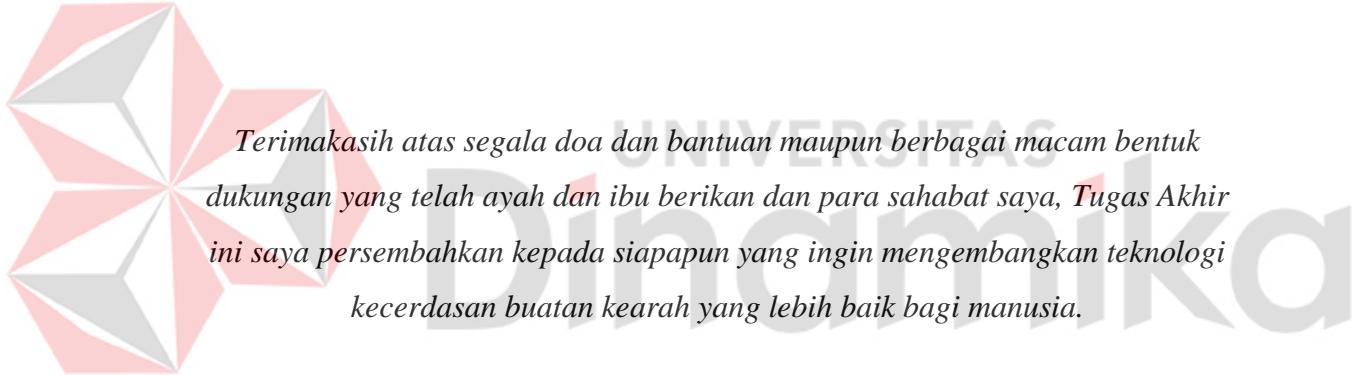
Dekan Fakultas Teknologi dan Informatika

UNIVERSITAS DINAMIKA

“We all make choices in life, but in the end our choices make us.”



UNIVERSITAS
Dinamika



Terimakasih atas segala doa dan bantuan maupun berbagai macam bentuk dukungan yang telah ayah dan ibu berikan dan para sahabat saya, Tugas Akhir ini saya persembahkan kepada siapapun yang ingin mengembangkan teknologi kecerdasan buatan kearah yang lebih baik bagi manusia.

SURAT PERNYATAAN
PERSETUJUAN PUBLIKASI DAN KEASLIAN KARYA ILMIAH

Sebagai mahasiswa Universitas Dinamika, saya :

Nama : Akbar Mahardi Hidayahullah
NIM : 18410200048
Program Studi : S1 Teknik Komputer
Fakultas : Fakultas Teknologi dan Informatika
Jenis Karya : Laporan Tugas Akhir
Judul Karya : **SISTEM DETEKSI SIMBOL PADA SIBI (SISTEM ISYARAT BAHASA INDONESIA) SECARA REALTIME MENGGUNAKAN MOBILENET-SSD**

Menyatakan dengan sesungguhnya bahwa:

1. Demi pengembangan Ilmu Pengetahuan, Teknologi dan Seni, saya menyetujui memberikan kepada Universitas Dinamika Hak Bebas Royalti Non-Eksklusif (*Non-Exclusive Royalty Free Right*) atas seluruh isi/ sebagian karya ilmiah saya tersebut di atas untuk disimpan, dialihmediakan dan dikelola dalam bentuk pangkalan data (*database*) untuk selanjutnya didistribusikan atau dipublikasikan demi kepentingan akademis dengan tetap mencantumkan nama saya sebagai penulis atau sebagai pemilik pencipta dan Hak Cipta
2. Karya tersebut di atas adalah karya asli saya, bukan plagiat baik sebagian maupun keseluruhan. Kutipan, karya atau pendapat orang lain yang ada dalam karya ilmiah ini adalah semata hanya rujukan yang dicantumkan dalam Daftar Pustaka saya
3. Apabila dikemudian hari ditemukan dan terbukti terdapat tindakan plagiat pada karya ilmiah ini, maka saya bersedia untuk menerima pencabutan terhadap gelar kesarjanaan yang telah diberikan kepada saya.

Demikian surat pernyataan ini saya buat dengan sebenarnya.

Surabaya, 15 Juni 2022

Yang menyatakan



Akbar Mahardi Hidayahullah

NIM : 18410200048

ABSTRAK

Komunikasi adalah proses ketika seseorang saling menyampaikan informasi, bagi penyandang tunarungu dan tunawicara mereka kesulitan dalam hal berkomunikasi. Bahasa isyarat adalah salah satu bentuk alternatif media komunikasi bagi mereka, untuk orang normal cukup kesulitan dalam memahami makna komunikasi menggunakan bahasa isyarat. Oleh karena itu, penggunaan teknologi *deep learning* bisa digunakan untuk menjembatani permasalahan sebagaimana tersebut diatas. Pada dua penelitian sebelumnya yang dilakukan oleh Ilham Rizaldy W.P., meneliti sistem deteksi SIBI menggunakan CNN dan metode deteksi tepi Canny, sedangkan penelitian kedua dengan objek yang penelitian yang sama dilakukan oleh Nofal Anam dengan menggunakan MediaPipe dan ResNet-50. Kedua penelitian tersebut diatas menerapkan metode *Image Classification*, adapun pada penelitian ini peneliti mengembangkannya menjadi metode *Object Detection* secara *realtime* menggunakan model *Pre-Trained* MobileNet-SSDv2. Pada Tugas Akhir ini telah dibuat *dataset* sebanyak 600 citra yang terdiri dari 6 simbol SIBI dengan pembagian 480 citra untuk *training* dan 120 citra untuk *testing*. Proses *training* model dilakukan sebanyak 3 kali dengan *step* yang berbeda, yaitu: 5.000, 10.000 dan 20.000. Hasil *training* model dengan 5.000 *step* mempunyai nilai *loss* sebesar 0,2712 dan akurasinya 83,3%, sedangkan pada proses *training* 10.000 *step* mendapatkan nilai *loss* 0,1881 dan akurasinya 83,6%, kemudian pada proses *training* 20.000 *step* memperoleh nilai *loss* 0,1387 dengan akurasinya 86,6%.

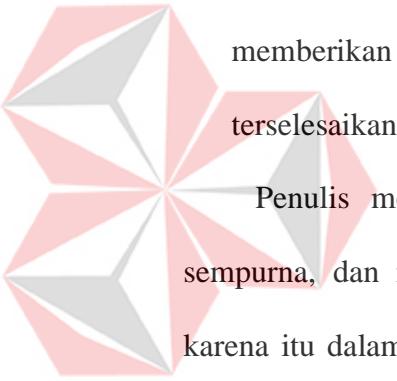
Kata kunci : Bahasa Isyarat, *Deep Learning*, MobileNet-SSD, *Pre-trained model network*.

KATA PENGANTAR

Dengan mengucapkan puji dan syukur terhadap kehadirat kepada Tuhan Yang Maha Esa yang telah memberikan limpahan berkah dan rahmat-Nya, serta segala karunia yang telah diberikan kepada penulis, sehingga penulis dapat merampungkan Laporan Tugas Akhir yang berjudul "SISTEM DETEKSI SIMBOL PADA SIBI (SISTEM ISYARAT BAHASA INDONESIA) SECARA *REALTIME* MENGGUNAKAN MOBILENET-SSD". Laporan Tugas Akhir ini disusun dalam rangka memenuhi salah satu prasyarat dalam menyelesaikan Program Sarjana Teknik Komputer di Universitas Dinamika.

Pada kesempatan yang telah diberikan ini, penulis mengucapkan rasa terima kasih terhadap individu-individu yang memberikan dukungan dan juga saran serta bimbingan dalam upaya untuk menyelesaikan laporan Tugas Akhir ini. Oleh karena itu penulis ingin mengucapkan terima kasih kepada:

1. Orang Tua penulis, yang telah memberikan kontribusi besar berupa dukungan penuh atas apa yang akan penulis lakukan dan juga atas apa yang telah penulis lakukan sehingga dapat menyelesaikan Laporan Tugas Akhir ini.
2. Ibu Tri Sagirani, S.Kom., M.MT., selaku Dekan Fakultas Teknologi dan Informatika (FTI) Universitas Dinamika.
3. Bapak Pauladie Susanto, S.Kom., M.T., selaku selaku Pembahas dalam Tugas Akhir ini dan selaku Ketua Program Studi S1 Teknik Komputer Universitas Dinamika.
4. Bapak Heri Pratikno, M.T., MTCNA., MTCRE., selaku Dosen Pembimbing I yang selalu memberikan waktu dan bimbingan serta ilmu dalam menyelesaikan Tugas Akhir beserta laporan ini.

- 
5. Ibu Yosefine Triwidystuti, M.T., selaku Dosen Pembimbing II yang juga selalu memberi waktu dan bimbingan dalam menyelesaikan Tugas Akhir beserta laporan ini.
 6. Seluruh Dosen pengajar Program Studi S1 Teknik Komputer Universitas Dinamika yang telah memberikan ilmu, dan juga bimbingan yang berharga dari semester 1 hingga sampai saat ini.
 7. Seluruh rekan-rekan S1 Teknik Komputer angkatan 2018 yang telah memberikan dukungan dan semangatnya untuk membantu penulis untuk menyelesaikan Laporan Tugas Akhir ini.
 8. Dan seluruh pihak yang tidak dapat disebutkan satu per satu yang telah memberikan dukungan serta bantuan dalam segala bentuk yang akhirnya terselesaikannya Laporan Tugas Akhir ini.

Penulis memahami bahwa Laporan Tugas Akhir ini belum mencapai sempurna, dan masih banyak kekurangan dalam menyusun laporan ini. Oleh karena itu dalam kesempatan ini, penulis meminta maaf apabila dalam Laporan Tugas Akhir ini masih terdapat kesalahan baik dalam penulisan maupun Bahasa yang digunakan. Penulis juga memerlukan kritik dan saran dari para pembaca yang sifatnya membangun untuk kesempurnaan laporan yang telah penulis susun.

Surabaya, 10 Juni 2022

Penulis

DAFTAR ISI

	Halaman
ABSTRAK	vii
KATA PENGANTAR.....	viii
DAFTAR ISI.....	x
DAFTAR GAMBAR.....	xii
DAFTAR TABEL.....	xiii
DAFTAR LAMPIRAN	xiv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah.....	3
1.4 Tujuan.....	4
1.5 Manfaat.....	4
BAB II LANDASAN TEORI	5
2.1 Tunarungu dan Tunawicara.....	5
2.2 SIBI (Sistem Isyarat Bahasa Indonesia)	5
2.3 Python.....	6
2.4 Anaconda.....	6
2.5 <i>Deep Learning</i>	7
2.6 <i>Image Classification</i> dan <i>Object Detection</i>	7
2.7 <i>Convolution Neural Network</i> (CNN).....	8
2.8 MobileNet-SSDv2	9
BAB III METODOLOGI PENELITIAN	12
3.1 <i>Setting Virtual Environment</i>	12
3.2 Instalasi Cuda dan CUDNN	12



3.3	<i>Dataset</i>	12
3.4	<i>Labeling</i>	13
3.5	Proses <i>Training</i> model <i>Pre-Trained</i> MobileNet-SSDv2	15
3.6	Proses <i>Testing</i> model <i>Pre-trained</i> MobileNet-SSDv2	16
BAB IV HASIL DAN PEMBAHASAN		17
4.1	<i>Training dataset</i> 6 simbol SIBI	17
4.1.1	Prosedur pengujian <i>training dataset</i> model MobileNet-SSD	17
4.1.2	Hasil <i>training dataset</i> model MobileNet-SSD	17
4.1.3	Hasil kesimpulan <i>training dataset</i> model MobileNet-SSD	23
4.2	Pengujian Hasil Prediksi Model	24
4.2.1	Tujuan Pengujian Hasil Prediksi Model <i>Train</i>	24
4.2.2	Prosedur Pengujian Hasil Prediksi Model <i>Train</i>	24
4.2.3	Hasil Pengujian Deteksi Simbol dan FPS	24
4.2.4	Hasil kesimpulan pengujian deteksi simbol dan FPS	29
BAB V PENUTUP		30
5.1	Kesimpulan.....	30
5.2	Saran.....	31
DAFTAR PUSTAKA		32

DAFTAR GAMBAR

Gambar 2.1. Simbol SIBI.....	6
Gambar 2.2. Arsitektur CNN	8
Gambar 2.3. Arsitektur MobileNetV2	10
Gambar 2.4. Arsitektur MobileNet-SSD.....	11
Gambar 3.1 <i>Dataset</i>	13
Gambar 3.2. <i>Labeling</i> pada Labelmg	14
Gambar 3.3. <i>Dataset</i> dengan file XML.....	14
Gambar 3.4. <i>Flowchart</i> proses <i>training</i> model MobileNet-SSDv2	15
Gambar 3.5. <i>Flowchart</i> proses <i>testing</i> MobileNet-SSDv2 pada <i>webcam</i>	16



DAFTAR TABEL

Tabel 4.1 Hasil kesimpulan dari ketiga model <i>train</i>	23
Tabel 4.2 Hasil pengujian deteksi simbol pada model <i>train 5.000 step</i>	26
Tabel 4.3 Hasil pengujian deteksi simbol pada model <i>train 10.000 step</i>	27
Tabel 4.4 Hasil pengujian deteksi simbol pada model <i>train 20.000 step</i>	28
Tabel 4.5 Hasil kesimpulan pengujian deteksi simbol dan FPS	29



DAFTAR LAMPIRAN

Lampiran 1 <i>Source code</i>	44
Lampiran 2 Hasil Turnitin.....	47



BAB I

PENDAHULUAN

1.1 Latar Belakang

Komunikasi adalah proses ketika seseorang saling menyampaikan dan menggunakan informasi agar saling terhubung dengan lingkungan dan masyarakat sekitar sebagaimana anak-anak normal pada umumnya. Tetapi untuk penyandang tunarungu dan tunawicara mereka kesulitan dalam berinteraksi dalam kehidupan sehari-hari dikarenakan keterbatasan mereka dalam menerima dan menyampaikan informasi, maka dari itu terbentuknya bahasa isyarat sebagai salah satu alternatif media komunikasi bagi penyandang tunarungu dan tunawicara. Bahasa isyarat dilakukan dengan menggunakan gerakan-gerakan tubuh dan mimik wajah sebagai simbol pengganti bahasa isyarat.

Di Indonesia bahasa isyarat yang sering digunakan dan sudah ditetapkan adalah Sistem Isyarat Bahasa Indonesia (SIBI). Bentuk kosakata bahasa isyarat SIBI diadopsi dari *American Sign Language* (ASL), bahasa isyarat SIBI juga sering dipakai pada Sekolah Luar Biasa (SLB) untuk komunikasi antar guru dan siswa. Interaksi dan komunikasi juga tidak hanya terjadi pada antar penyandang tunarungu dan tunawicara, namun juga terhadap orang normal. Pemahaman makna yang rumit menyebabkan kesusahan pada orang normal untuk memahami bahasa isyarat, oleh karena itu penggunaan teknologi *deep learning* bisa digunakan untuk mengatasi permasalahan seperti ini.

Pada penelitian sebelumnya (Anam, 2022), membuat rancangan sistem deteksi simbol pada SIBI dengan menggunakan menggunakan MediaPipe dan Resnet-50. Dalam penelitiannya tersebut dibuat dua *dataset* yaitu dataset asli dan dataset yang menggunakan MediaPipe yang masing-masing memiliki 1200 gambar dan terbagi kedalam 6 simbol pose SIBI, yaitu: “Aku”, “Cinta”, “Dia”, “Kamu”, “Maaf” dan “Sedih”. Hasil dari penelitian untuk *training dataset* hasil ekstrasi *landmark* dari *mediapipe* didapatkan akurasi *training* sebesar 88% dan akurasi *validation* 87% dengan nilai *loss* pada *training* dan validasi sebesar 0,30 yang berarti model tersebut sudah termasuk optimal dengan kesalahan yang

minimum. Tetapi untuk hasil dengan citra asli tanpa landmark didapatkan akurasi *training* sebesar 33% dan akurasi validasi sebesar 40% yang berarti tanpa adanya *landmark* pada MediaPipe menyebabkan penurun akurasi dan validasi pada model *training*.

Menurut penelitian yang dilakukan (Putra, 2021), membuat rancangan sistem deteksi simbol pada SIBI dengan menggunakan *Convolution Neural Network* (CNN). Penelitian tersebut menggunakan *dataset* yang diambil sendiri dengan total *dataset* sebanyak 660 citra. Setelah itu melakukan *pre processing image* dan *augmentasi image* pada *dataset* seperti *resize* menjadi 100x100 dan menghilangkan *background*. Lalu *dataset* tersebut dilakukan proses *Edge Detection* menggunakan filter Canny untuk mencari garis tepi pada citra, sehingga garis tepi dari gambar *dataset* dapat terlihat jelas. Setelah proses *Edge Detection* dilakukan, maka *dataset* sudah siap untuk melakukan proses *training* dengan menggunakan metode CNN. Hasil dari penelitian untuk *traning* menggunakan 250 *epoch* diketahui akurasi *training* sebesar 91% dan akurasi *validation* sebesar 90% namun juga memiliki nilai *loss* 25 untuk *training* dan 23 untuk validasi yang berarti untuk setiap 25 *dataset* pada *training* masih ada kesalahan pada prediksi yang besar.

Pada Tugas Akhir ini, penulis mengusulkan pengembangan deteksi simbol bahasa isyarat SIBI secara *realtime* dengan menggunakan *pre-trained* model MobileNet-SSD. Dengan penambahan *realtime* maka perbandingan pada kelancaran setiap *frame per second* di butuhkan untuk menentukan kualitas model dalam hal komputasi agar tidak membebani sistem deteksi pada *webcam*. Model ini dipilih karena model tersebut mendukung proses deteksi objek secara *realtime* dengan akurasi yang tinggi dan proses komputasi yang ringan. Menurut *paper* (Liu, 2015), pada percobaan deteksi secara *realtime* menggunakan *dataset* PASCAL VOC2007 dan VOC2012 untuk *training*, terdapat model SSD satu-satunya yang mencapai rata-rata akurasi diatas 70% dengan FPS mencapai hingga 30 FPS.

Untuk MobileNet menurut *paper* (Zhu, 2017), pada percobaan *training* *dataset* yang sama dengan GoogleNet dan VGG16 sebagai perbandingan, terdapat akurasi rata-rata sebesar 70% hampir sama akurat dengan VGG16 namun untuk

penggunaan parameter terdapat perbandingan VGG16 dengan 138 juta parameter dengan MobileNet yang hanya 4,2 juta parameter dengan artian penggunaan MobileNet lebih ringan pada proses komputasinya dengan perbedaan yang sedikit pada parameter namun lebih cepat Adapun *dataset* yang digunakan pada Tugas Akhir ini adalah *dataset* yang telah dikumpulkan oleh peneliti sendiri sebanyak 600 citra, terdiri dari 5 subjek yang masing-masing melakukan pose pada 6 simbol, dimana setiap simbol terdiri dari 20 citra.

1.2 Rumusan Masalah

Berlandaskan latar belakang diatas, maka dapat dirumuskan dengan rumusan masalah pada Tugas Akhir ini sebagai berikut :

1. Bagaimana performa dari sistem deteksi simbol pada Sistem Isyarat Bahasa Indonesia dengan menggunakan *Pre-trained* MobileNet-SSDv2 ?
2. Berapa besar *Frame Per Second* (FPS) dalam proses *realtime* deteksi simbol pada Sistem Isyarat Bahasa Indonesia ?

1.3 Batasan Masalah

Dalam pembuatan Tugas Akhir ini, Pembahasan masalah memiliki sebuah batasan terhadap beberapa hal berikut :

1. *Dataset* yang digunakan hanya 6 simbol SIBI.
2. Pengujian perlu dilakukan dengan penerangan cahaya yang merata.
3. Pengujian yang dilakukan dengan menggunakan kamera *webcam*.

1.4 Tujuan

Berlandaskan latar belakang dan rumusan masalah diatas, mendapatkan tujuan pada tugas akhir ini sebagai berikut :

1. Mengetahui performa yang diperoleh dari model *training* pada sistem deteksi simbol Sistem Isyarat Bahasa Indonesia.
2. Mengetahui kinerja *realtime* proses deteksi simbol Sistem Isyarat Bahasa Indonesia melalui besarnya *Frame Per Second* (FPS)

1.5 Manfaat

Adapun dari Tugas Akhir ini dapat memperoleh manfaat sebagai berikut :

1. Dapat menambah pengetahuan pada penerapan sistem secara *realtime* mengenai deteksi simbol pada Sistem Isyarat Bahasa Indonesia menggunakan *Pre-trained* MobileNet-SSDv2
2. Dapat menjadi referensi bagi mahasiswa jika ingin melakukan penelitian mengenai model *Pre-trained* MobileNet-SSDv2l.
3. Memudahkan orang normal untuk memahami cara berkomunikasi kepada penyandang tunarungu dan tunawicara

BAB II

LANDASAN TEORI

2.1 Tunarungu dan Tunawicara

Tunarungu merupakan anak yang mempunyai gangguan pada pendengarannya sehingga tidak dapat mendengar bunyi dengan sempurna atau bahkan tidak dapat mendengar sama sekali. Meskipun anak tunarungu sudah diberikan alat bantu dengar, tetap saja anak tunarungu masih memerlukan pelayanan pendidikan khusus. (Riadi, 2018).

Tunawicara merupakan ketidak-mampuan seseorang dalam berbicara. Hal ini disebabkan oleh kurang atau tidak berfungsi organ-organ untuk berbicara, seperti rongga mulut, lidah dan pita suara. Selain itu juga adanya kekurangan pada indra pendengaran, keterlambatan perkembangan bahasa. (Wiranda, 2019).

Sebagai pengganti komunikasi bagi penyandang tunarungu dan tunawicara adalah bahasa isyarat. Bahasa isyarat dilakukan dengan menggunakan gerakan-gerakan tubuh yang kemudian disepakati maknanya oleh para penyandang tunarungu dan tunawicara setempat, sehingga bahasa isyarat tersebut disesuaikan dengan budaya masing-masing. Di Indonesia sendiri terdapat 2 jenis bahasa isyarat yang sedang berkembang, yaitu Bahasa Isyarat Indonesia (BISINDO) dan Sistem Isyarat Bahasa Indonesia (SIBI).

2.2 SIBI (Sistem Isyarat Bahasa Indonesia)

Sistem Isyarat Bahasa Indonesia (SIBI) merupakan salah satu bahasa isyarat bagi penyandang tunarungu untuk melakukan komunikasi dengan sesama penyandang tunarungu. Dengan menggunakan tatanan jari, tangan dan bahasa tubuh. SIBI adalah bahasa isyarat yang telah diresmikan oleh pemerintah sebagai bahasa isyarat resmi dan media pembelajarannya menggunakan kamus bahasa SIBI yang sudah tertata menurut penggunaanya. (Nasir, 2021).

Sistem Isyarat Bahasa Indonesia (SIBI) merupakan bahasa isyarat yang diadopsi dari *American Sign Language* (ASL). Bahasa isyarat ini juga biasa

dipakai di Sekolah Luar Biasa (SLB) untuk berkomunikasi antara guru dan siswa maupun antar siswa.



Gambar 2.1. Simbol SIBI

(Sumber: <https://komunita.id/2018/11/30/mengenal-bahasa-isyarat/>)

2.3 Python

Python adalah bahasa pemrograman interperatif multiguna dengan filosofi perancangan yang berfokus pada tingkat keterbacaan kode. Python diklaim sebagai bahasa yang menggabungkan kapabilitas, kemampuan, dengan sintaksis kode yang sangat jelas dan dilengkapi dengan fungsionalitas pustaka yang besar serta komprehensif. Python juga didukung oleh komunitas yang besar. (Syahrudin, 2018)

2.4 Anaconda

Anaconda adalah *platform* ilmu data untuk melakukan performa pemrograman Python/R dan *machine learning* pada mesin lokal. Anaconda dibangun dari *data scientist* untuk *data scientist*. Anaconda menyediakan banyak *library* dan akses untuk model, data dan *package* untuk sebuah projek pada *data science* atau *machine learning*. Anaconda juga bersifat *open-source* yang berarti semua orang bisa mengakses atau menginstall anaconda. (Anaconda, 2022)

2.5 Deep Learning

Deep learning merupakan salah satu bidang dari *Machine Learning* yang memanfaatkan jaringan syaraf tiruan untuk implementasi permasalahan dengan dataset yang besar. Teknik *deep learning* memberikan arsitektur yang sangat kuat untuk *Supervised Learning*. Dengan menambahkan lebih banyak lapisan maka model pembelajaran tersebut bisa mewakili data citra berlabel dengan lebih baik. (Sindy, 2019).

2.6 Image Classification dan Object Detection

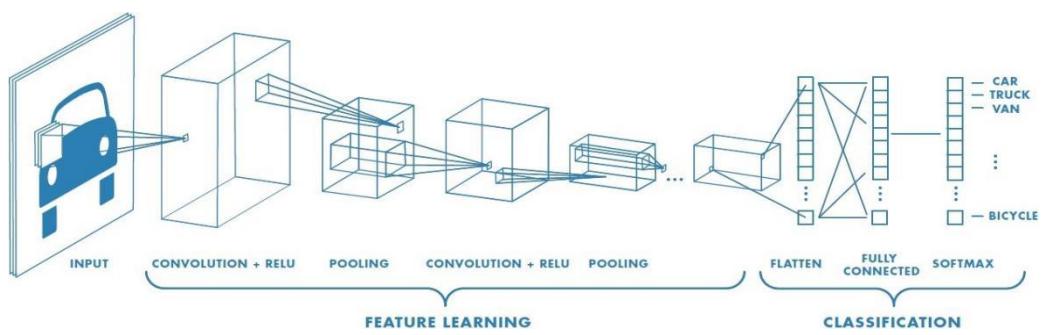
Secara singkat *Image Classification* adalah teknik yang digunakan untuk klasifikasi atau prediksi class pada suatu objek tertentu pada suatu citra, dimana *output* tersebut hanya memprediksi suatu *class* terhadap citra yang telah di *training*. Kekurangan dari *Image Classification* dalam teknik *supervised* maupun *unsupervised* adalah banyaknya waktu yang dibutuhkan selama fase pelatihan dan tidak cocok untuk menangani data besar.

Secara singkat *Object Detection* adalah salah satu teknik klasifikasi citra juga dan selain klasifikasi, teknik ini juga mengidentifikasi lokasi suatu objek dari sejumlah besar kategori yang telah ditentukan dengan memberi sebuah *bounding box* ketika berhasil menemukan kategori yang telah di prediksi. Teknik ini juga dapat digunakan dalam projek *realtime* dalam dunia nyata seperti deteksi wajah, deteksi kendaraan, pengawasan video dan lain-lain. Selama beberapa tahun terakhir, *Object Detection* mencapai sukses besar dalam menangani prediksi pada lingkungan yang terkendali, namun masalah tetap tidak terpecahkan pada tempat yang tidak terkendali, khususnya ketika objek ditempatkan dalam pose sewenang-wenang di lingkungan tertutup yang berantakan. (Choudhury, 2020)

2.7 Convolution Neural Network (CNN)

Convolution Neural Network (CNN) adalah salah satu jenis *Neural Network* yang biasa digunakan pada data *image*. CNN bisa digunakan untuk mendeteksi dan klasifikasi objek pada sebuah citra. Secara garis besar CNN tidak jauh beda dari *neural network* lainnya. CNN terdiri dari *neuron* yang memiliki *weight*, *bias* dan *activation function*. CNN memanfaatkan proses konvolusi dengan mengerakkan sebuah kernel konvolusi (filter) berukuran tertentu ke sebuah citra, komputer mendapatkan informasi representatif baru dari hasil perkalian bagian gambar tersebut dengan filter yang digunakan.

Arsitektur dari CNN dibagi menjadi 2 bagian besar. Proses pertama yang adalah melakukan “*encoding*” dari sebuah *image* menjadi *features* berupa angka-angka yang mempresentasikan *image* (*Feature Extraction Layer*). *Feature Extraction Layer* sendiri terdiri dari 2 bagian yaitu *Convolution Layer* dan *Pooling Layer*. Proses kedua adalah melakukan “*flatten*” atau *reshape feature map* yang dihasilkan dari *Feature Extraction* menjadi sebuah *vector* agar bisa digunakan sebagai *input fully connected layer*. *Fully-Connected* adalah lapisan dimana semua neuron aktifasi dari lapisan sebelumnya terhubung dengan semua *neuron* di lapisan selanjutnya. Lapisan *Fully-Connected* biasanya digunakan pada metode *Multi Layer Perceptron* yang bertujuan untuk mengolah data sehingga bisa diklasifikasikan. (LINA, 2019).



Gambar 2.2. Arsitektur CNN

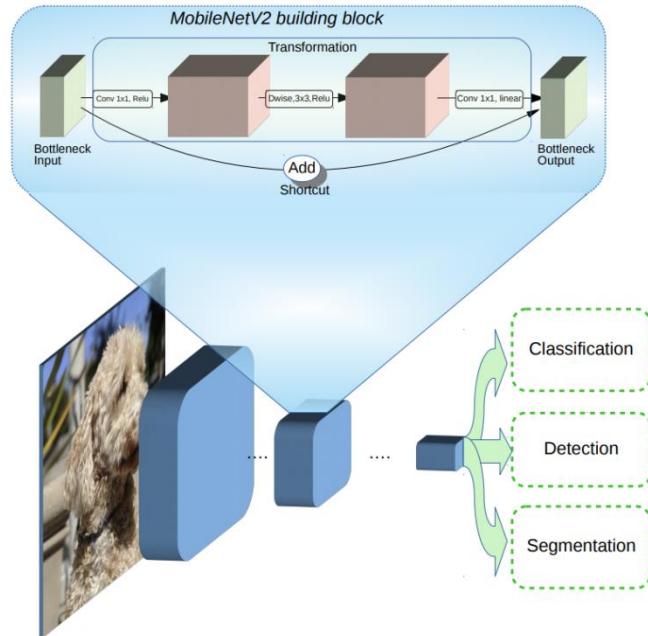
(Sumber: <https://medium.com/@samuelsena/pengenalan-deep-learning-part-7-convolutional-neural-network-cnn-b003b477dc94>)

2.8 MobileNet-SSDv2

Pada penelitian ini digunakan model arsitektur MobileNet-SSD, dimana model ini terdiri dari 2 jenis *Neural Network* yaitu *Single Shot Detector* (SSD) dan MobileNet. MobileNet adalah arsitektur *Neural Network* yang ringan dan dapat digunakan untuk komputasi rendah seperti *mobile application*. Dalam penerapan di dunia nyata seperti mobil *self-driving*, tugas untuk pendekslsian dan klasifikasi citra memiliki keterbatasan dalam komputasi seperti ukuran aplikasi yang besar, karena kebutuhan seperti itu MobileNet dikembangkan pada tahun 2017.

MobileNet menggunakan *depthwise separable convolution* yang secara signifikan mengurangi jumlah parameter jika dibandingkan dengan *reguler convolution*, dari hal tersebut MobileNet menjadi *deep neural network* yang sangat ringan (Zhu, 2017). Pada model MobileNetv2 terdapat 53 *convolution layer* dan memperkenalkan 2 fitur baru ke arsitektur yaitu *Linear Bootleneck* diantara *layer* dan *shortcut* diantara *bootleneck layer* (*Inverted Residual*). Terdapat 2 tipe *convolution layer* pada mobileNetv2 yaitu 1×1 *convolution* dan 3×3 *Depthwise Convolution*.

Pada Gambar 2.3 kotak biru menggambarkan blok pembentukan *linear convolution* atau *bootleneck*. Dengan penambahan fitur tersebut pada percobaan dengan *dataset* COCO untuk deteksi *realtime* pada *paper* (Sandler, 2019) dengan perbandingan oleh model sebelumnya yaitu MobileNetv1 terdapat parameter yang digunakan MobileNetv2 lebih sedikit 800 ribu parameter dan penggunaan CPU lebih rendah 70ms yang dapat diartikan v2 lebih ringan pada komputasinya. Maka dari itu penelitian ini menjadi menggunakan model MobileNet-SSDv2.



Gambar 2.3. Arsitektur MobileNetV2

(Sumber: <https://ai.googleblog.com/2018/04/mobilenetv2-next-generation-of-on.html>)

Single Shot Detector (SSD) adalah arsitektur *Neural Network* yang juga dikembangkan oleh tim Google Research pada tahun 2016 untuk memenuhi kebutuhan model yang dapat digunakan untuk deteksi secara *realtime* pada perangkat tanpa perbedaan akurasi yang signifikan, SSD mengambil satu bidikan tunggal untuk mendekksi banyak objek dalam gambar dengan menggunakan *bounding box* dan klasifikasi secara bersamaan. SSD dirancang untuk tidak bergantung pada jaringan dasar, sehingga dapat berjalan diatas hampir semua *neural network* lainnya, termasuk MobileNet. Arsitektur SSD dibangun diatas arsitektur VGG16, tetapi membuang bagian *fully-connected layer* nya. (Zhu, 2017).

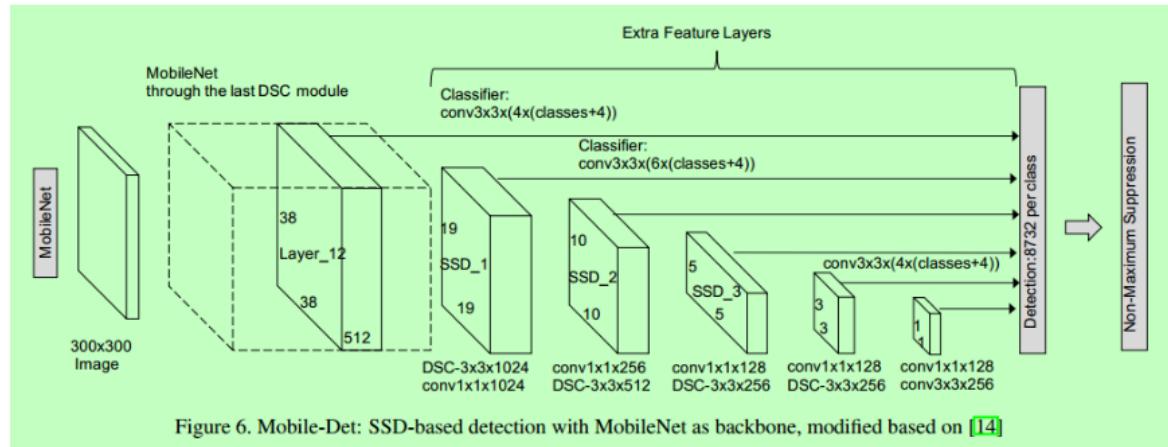


Figure 6. Mobile-Det: SSD-based detection with MobileNet as backbone, modified based on [14]

Gambar 2.4. Arsitektur MobileNet-SSD

(Sumber:<https://medium.com/@techmayank2000/object-detection-using-ssd-mobilenetv2-using-tensorflow-api-can-detect-any-single-class-from-31a31bb0691>)

Model *Single Shot Detector* (SSD) menggunakan MobileNet sebagai *base network* dan kemudian diikuti oleh beberapa *convolution layer* seperti pada Gambar 2.4 dengan *Fully-Connected*, *Maxpool* dan *softmax* dihilangkan. Jadi *output* dari *convolution layer* terakhir milik MobileNet digunakan sebagai penggantinya. Model MobileNet-SSDv2 menggabungkan *output* dari 6 tingkat resolusi dan melakukan kalkulasi hingga 3000 *bounding box*, setelah itu pada bagian akhir melakukan filter *bounding box* dengan menggunakan *Non-Maximum Supression* (NMS). (Y, 2021).

BAB III

METODOLOGI PENELITIAN

3.1 *Setting Virtual Environment*

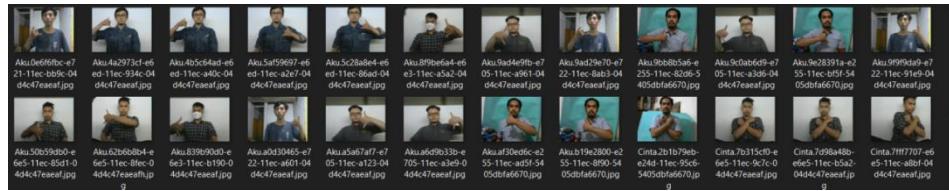
Sebelum memulai projek ini diperlukan melakukan *set up* seperti *install* anaconda dan membuat *virtual environment* pada anaconda. Dikarenakan pada projek objek deteksi ini membutuhkan *library* khusus pada tensorflow dan *library* NVIDIA *graphic card* seperti CUDA *toolkit* dan CUDNN jika *device* yang dipakai memiliki NVIDIA *graphic card*. Untuk TensorFlow diperlukan versi diatas 2.5 dan instalasi *library* untuk python.

3.2 Instalasi Cuda dan CUDNN

Khusus untuk pengguna NVIDIA *graphic card* pada *device* yang digunakan, maka membutuhkan instalasi CUDA *toolkit* dan CUDNN. Instalasi tersebut digunakan untuk melakukan *training* dengan *Graphic Processing Unit* (GPU) yang dapat mempercepat proses *training* namun pada projek ini dibutuhkan *Video Random Access Memory* (VRAM) sebesar 6GB. *Training* menggunakan GPU mempercepat proses *training* yang dimana biasanya membutuhkan lima sampai sepuluh jam *training* menjadi satu sampai dua jam *training*.

3.3 *Dataset*

Dataset yang digunakan pada penelitian ini adalah *dataset* yang telah dikumpulkan oleh penulis sendiri dengan menggunakan *webcam*. Pada *dataset* terdapat citra manusia yang memperagakan 6 simbol bahasa isyarat SIBI yang terdiri dari simbol ‘Saya’, ‘Dia’, ‘Kamu’, ‘Cinta’, ‘Maaf’, dan ‘Sedih’.



Gambar 3.1. Dataset

Dalam *dataset* tersebut sebanyak 600 citra dengan pengambilan tiap simbol sebanyak 20 citra, pengambilan citra dilakukan dengan 5 subjek orang yang berbeda sebagai model dengan total masing-masing subjek memiliki 120 citra. Pengambilan citra dilakukan dengan program pada python yang dapat mengambil citra secara otomatis sehingga untuk subjek model hanya perlu memindahkan posisi tangan dengan cepat.

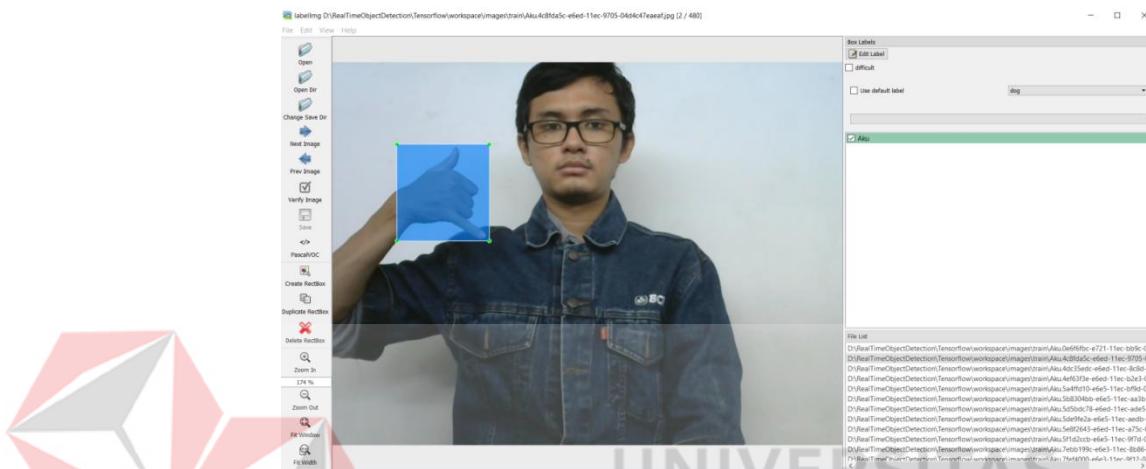
Citra yang diambil hanya pada posisi badan atas dan menggunakan latar belakang yang polos untuk memudahkan model pada saat *training* dan mendapatkan hasil akurasi yang baik. *Dataset* dibagi menjadi 2 bagian yaitu *training* dan *testing* dengan perbandingan sejumlah 80% untuk *training* dan 20% untuk *testing*. Dari total 600 citra didapatkan 480 sebagai data *training* dan 120 sebagai data *testing*.

3.4 Labeling

Proses selanjutnya setelah mengambil *dataset* adalah melakukan *labeling* pada setiap 6 simbol SIBI sebelum dilakukan proses *training*, proses ini dilakukan untuk memberi label pada setiap simbol sehingga model *training* mengetahui nama simbol yang akan di prediksi. Pada proses *labeling* ini penulis menggunakan *annotation tool* untuk memberi lokasi prediksi tiap simbol *class* sebagai titik pembelajaran seperti pada Gambar 3.2, *annotation tool* yang digunakan adalah “LabelImg”

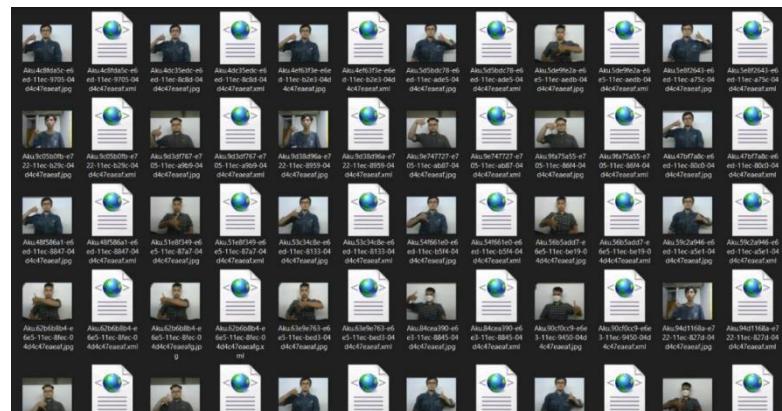
LabelImg adalah alat anotasi gambar yang programnya ditulis dengan menggunakan Python dan menggunakan Qt untuk antarmuka grafisnya. Anotasi disimpan sebagai file XML dalam format PASCAL VOC, format yang digunakan oleh ImageNet. Selain itu, LabelImg juga mendukung format YOLO dan CreateML (Tzutalin, 2015).

Dataset sebanyak 600 citra yang telah dikumpulkan oleh penulis dilakukan proses *manual* memberi kotak prediksi pada subjek yang memperagakan tiap simbol dan memberi label secara satu per satu. Supaya hasil *training* yang didapatkan nanti memiliki hasil akurasi yang bagus, maka saat *labeling* disarankan untuk memberi kotak yang sesuai dengan pose tangan, tidak terlalu lebar dan besar, seperti pada Gambar 3.2.



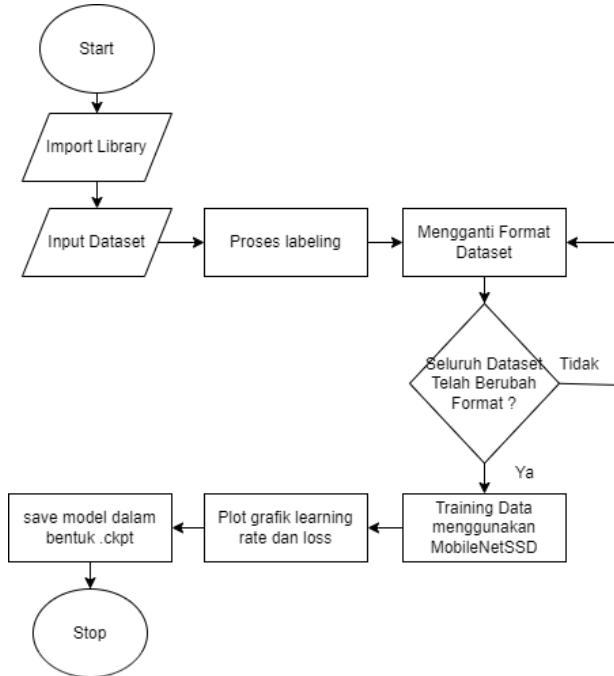
Gambar 3.2. *Labeling* pada Labelmg

Setelah memberi kotak prediksi sebanyak 600 citra maka akan didapatkan file XML, isi file XML tersebut merupakan koordinat lokasi *box* yang telah diberi oleh penulis pada citra. Total data pada setiap citra yang telah dilakukan *labeling* terdapat 1200 file dataset terdiri dari 600 file *image* dengan format JPG dan 600 file XML dengan format PASCAL VOC.



Gambar 3.3. *Dataset* dengan file XML.

3.5 Proses Training model Pre-Trained MobileNet-SSDv2



Gambar 3.4. Flowchart proses training

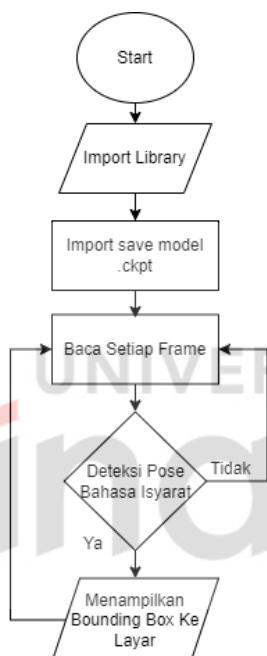
Pada Gambar 3.4 menunjukkan *flowchart* proses *training* model MobileNetSSDv2 yang diajukan pada Tugas Akhir ini. Dimulai dari *Start* lalu melakukan *import library* yang akan digunakan, setelah itu melakukan input lokasi *dataset*. Dalam proses ini dilakukan *labeling* dengan menggunakan LabelImg untuk memberi koordinat prediksi dari citra yang akan di *training* berupa format XML. Karena TensorFlow API hanya menerima format tfrecord, maka format citra dari XML dikonversi menjadi format tfrecord dengan menggunakan program yang telah disediakan oleh TensorFlow sendiri. Jika masih ada format xml yang belum di konversi maka harus diubah menjadi format tfrecord terlebih dahulu, setelah itu *dataset* siap untuk dilakukan proses *training* dengan model MobileNet-SSDv2. Pada saat proses *training dataset*, citra dibagi kedalam folder *train* dan *test* untuk dilakukan proses *training* secara terpisah.

Training dataset dilakukan pada Anaconda Command dan menggunakan GPU untuk *training*. Karena hal tersebut proses *training* hanya memerlukan waktu sebanyak satu hingga dua jam saja. Pada saat *training* model berakhir *training command* memunculkan hasil nilai *loss*, *learning_rate* dan *total step*. Dengan selesainya proses *training*, hasil model tersebut otomatis tersimpan

berupa .ckpt file. Penggunaan model *training* pada objek deteksi *realtime* sering menggunakan file .ckpt dikarenakan dapat melakukan *tuning* lagi atau ingin melakukan eksperimen pada model jika model terlalu *overtrain* atau hasil akurasi masih kurang optimal.

3.6 Proses *Testing* model *Pre-trained* MobileNet-SSDv2

Setelah model *training* didapatkan, proses untuk mendeteksi citra secara *realtime* dapat dilakukan dengan menggunakan *save* file model .ckpt terakhir.



Gambar 3.5. Flowchart proses *testing*

Pada Gambar 3.5 merupakan *flowchart* pada sistem deteksi simbol SIBI. Diawali dengan *import library* OpenCV untuk mendapatkan akses *webcam* yang digunakan dan dilanjutkan dengan melakukan *load save* model .ckpt. Pembacaan frame mulai berjalan ketika *webcam* menyala dan program deteksi tampil pada layar. Jika terdapat gerakan pose simbol bahasa isyarat yang telah di *training* muncul, maka *bounding box* akan terbentuk pada lokasi perkiraan gerakan pose yang terbaca sekaligus melakukan klasifikasi *class* bentuk simbol dan juga memunculkan akurasi prediksi dalam bentuk persen dan akan terus melakukan pembacaan setiap frame meskipun tidak ada pose yang terdeteksi.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Training dataset 6 simbol SIBI

Sebelum melakukan data *testing* pada sistem deteksi, diperlukan *training dataset* untuk mendapatkan model *training* sehingga model dapat mengetahui objek yang ingin di deteksi.

4.1.1 Prosedur pengujian training dataset model MobileNet-SSD

1. Memberikan *input* dari *dataset* yang telah dikumpulkan.
2. Memberikan konfigurasi berupa *training* sebesar 5.000, 10.000 dan 20.000 *step*.
3. Memberikan konfigurasi untuk nilai *start* pada *learning_rate* sebesar 0,08.
4. Hasil *training* data disimpan berupa file .ckpt.

4.1.2 Hasil training dataset model MobileNet-SSD

Pada *metric* TensorFlow API untuk *object detection* tersebut hanya bisa menampilkan *loss*, *learning_rate* dan *step*. Untuk nilai *recall* dan *precision* masih belum dapat dimunculkan dikarenakan terdapat *bug* pada versi TensorFlow yang digunakan yaitu versi 2.5.0. *Bug* tersebut menyebabkan *error* pada API saat melakukan program untuk menampilkan nilai tersebut. Data *training* menggunakan metode *optimizer* SGD (*Stochastic Gradient Descent*) yang merupakan metode iteratif untuk mengoptimalkan fungsi dengan tingkat *smoothness* yang sesuai.

Untuk mengurangi beban komputasi yang sangat tinggi dan mencapai *step* hingga ribuan dengan cepat digunakanlah proses komputasi yaitu ‘*learning decay*’. *Learning decay* adalah ketika semakin lama *training* maka *learning rate* menjadi rendah sebagai ganti untuk menurunkan nilai *error/loss*. *Learning decay* akan aktif ketika grafik pada *learning rate* mengalami lonjakan. Lonjakan pada grafik

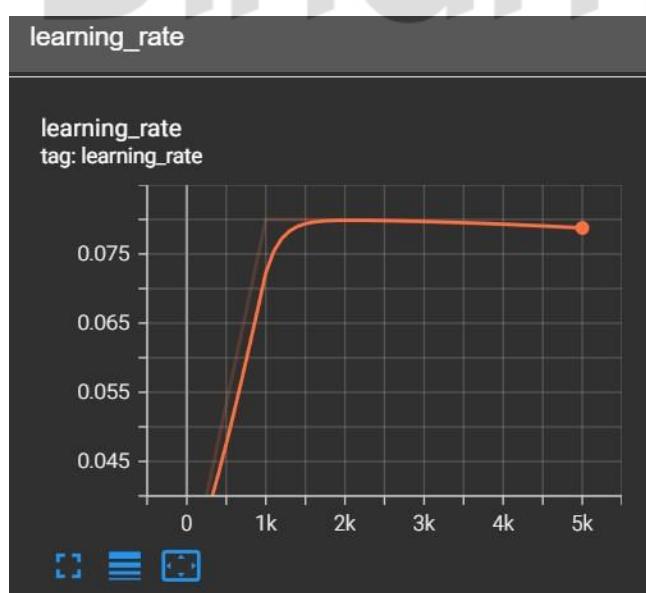
ini terjadi ketika pembelajaran pada model memasuki tahap yang tidak efektif yaitu mengalami kenaikan pada nilai loss pada pembelajaran, lonjakan ini juga dapat disebut dengan ‘*overshoot*’.

A. *Training Dataset* untuk 5.000 Step

Pada Gambar 4.1 merupakan hasil pengujian *loss training dataset* sebanyak 5.000 step dengan nilai akhir *loss* sebesar 0,2712 pada *loss/error*.

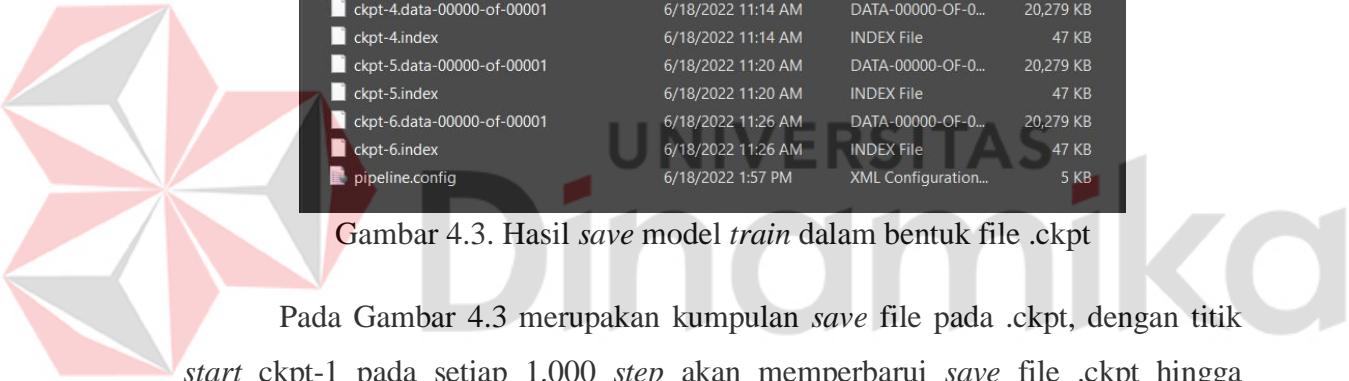


Gambar 4.1. Hasil uji *training total_loss* pada Tensorboard



Gambar 4.2. Hasil uji *training learning_rate* pada Tensorboard

Pada Gambar 4.2 untuk nilai *learning_rate* mendapatkan hasil akhir pada *step* ke 5.000 sebesar 0,0786. grafik *learning rate* pada model 5.000 terlihat konstan pada nilai 0,08 hingga memasuki *step* 4.000 mulai mengalami penurunan. Dapat terlihat pada Gambar 4.1, ketika memasuki *step* ke 4.500 grafik *loss* mengalami *overshoot* atau lonjakan pada nilai *loss*. Maka dari itu *learning decay* mulai berjalan untuk menghindari hal tersebut.



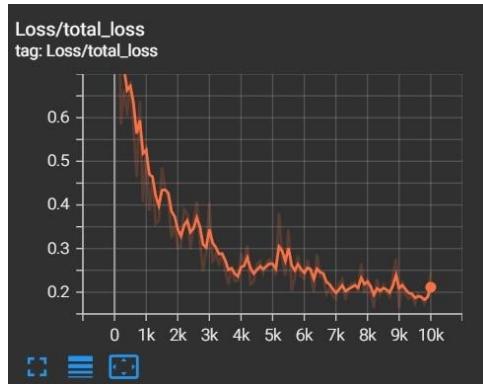
eval	6/18/2022 2:02 PM	File folder
train	6/18/2022 2:02 PM	File folder
checkpoint	6/18/2022 11:26 AM	File 1 KB
ckpt-1.data-00000-of-00001	6/18/2022 10:55 AM	DATA-00000-OF-0... 10,230 KB
ckpt-1.index	6/18/2022 10:55 AM	INDEX File 26 KB
ckpt-2.data-00000-of-00001	6/18/2022 11:02 AM	DATA-00000-OF-0... 20,279 KB
ckpt-2.index	6/18/2022 11:02 AM	INDEX File 47 KB
ckpt-3.data-00000-of-00001	6/18/2022 11:08 AM	DATA-00000-OF-0... 20,279 KB
ckpt-3.index	6/18/2022 11:08 AM	INDEX File 47 KB
ckpt-4.data-00000-of-00001	6/18/2022 11:14 AM	DATA-00000-OF-0... 20,279 KB
ckpt-4.index	6/18/2022 11:14 AM	INDEX File 47 KB
ckpt-5.data-00000-of-00001	6/18/2022 11:20 AM	DATA-00000-OF-0... 20,279 KB
ckpt-5.index	6/18/2022 11:20 AM	INDEX File 47 KB
ckpt-6.data-00000-of-00001	6/18/2022 11:26 AM	DATA-00000-OF-0... 20,279 KB
ckpt-6.index	6/18/2022 11:26 AM	INDEX File 47 KB
pipeline.config	6/18/2022 1:57 PM	XML Configuration... 5 KB

Gambar 4.3. Hasil *save* model *train* dalam bentuk file .ckpt

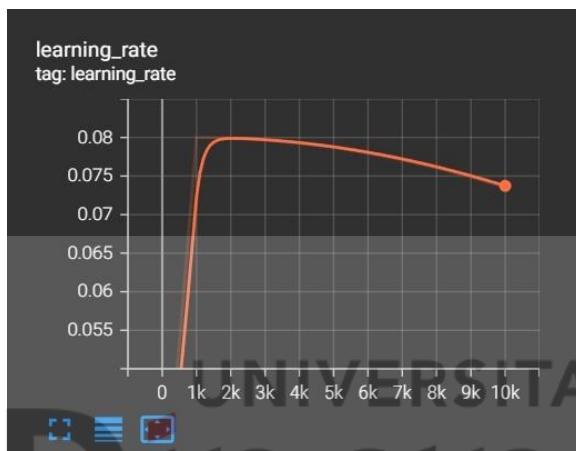
Pada Gambar 4.3 merupakan kumpulan *save* file pada .ckpt, dengan titik *start ckpt-1* pada setiap 1.000 *step* akan memperbarui *save* file .ckpt hingga berakhir pada file .ckpt-6. Perlu diketahui juga bahwa file .ckpt hanya menyimpan 7 file .ckpt terakhir. Yang akan digunakan sebagai model untuk memprediksi pada *webcam* adalah *save* file terakhir yaitu file .ckpt-6.

B. Training Dataset untuk 10.000 Step

Pada Gambar 4.4 merupakan hasil pengujian *loss training dataset* sebanyak 10.000 *step* dengan nilai akhir *loss* sebesar 0,1238 pada *loss/error*.



Gambar 4.4. Hasil uji *training total_loss* pada Tensorboard



Gambar 4.5. Hasil uji *training learning_rate* pada Tensorboard

Pada Gambar 4.5 untuk nilai *learning_rate* mendapatkan hasil akhir pada *step* ke 10.000 sebesar 0,0732. Ketika memasuki *step* ke 5.000, *learning rate* memasuki tahap *learning decay* dikarenakan pada grafik Gambar 4.4 yang terlihat pada saat *step* ke 5.000 mengalami *overshoot* atau lonjakan pada nilai *loss*.

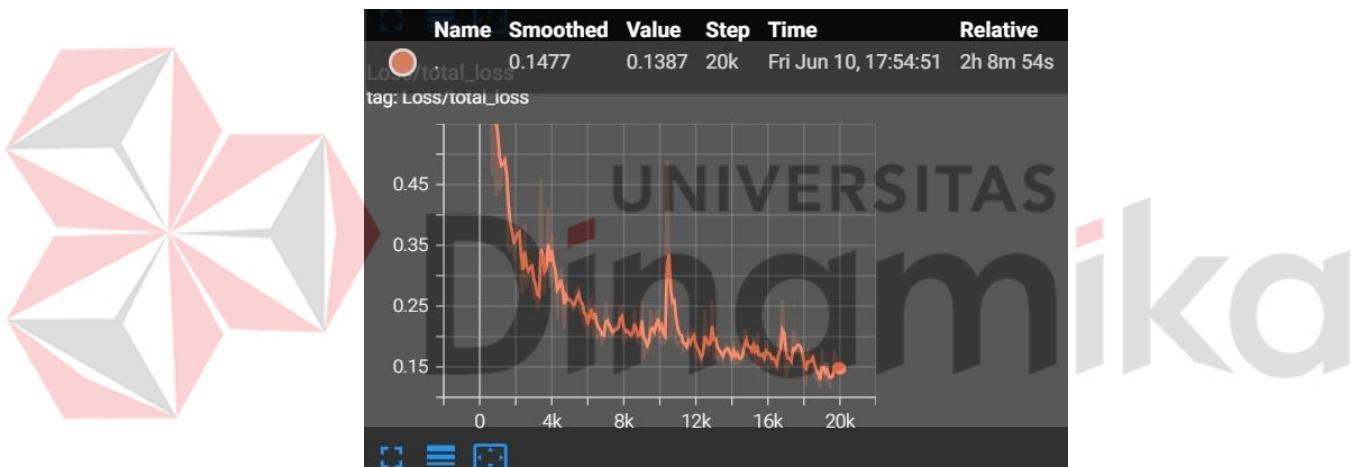
Name	Date modified	Type	Size
train	6/4/2022 4:12 PM	File folder	
checkpoint	6/4/2022 5:20 PM	File	1 KB
ckpt-5.data-00000-of-00001	6/4/2022 4:40 PM	DATA-00000-OF-0...	20,279 KB
ckpt-5.index	6/4/2022 4:40 PM	INDEX File	47 KB
ckpt-6.data-00000-of-00001	6/4/2022 4:47 PM	DATA-00000-OF-0...	20,279 KB
ckpt-6.index	6/4/2022 4:47 PM	INDEX File	47 KB
ckpt-7.data-00000-of-00001	6/4/2022 4:53 PM	DATA-00000-OF-0...	20,279 KB
ckpt-7.index	6/4/2022 4:53 PM	INDEX File	47 KB
ckpt-8.data-00000-of-00001	6/4/2022 5:00 PM	DATA-00000-OF-0...	20,279 KB
ckpt-8.index	6/4/2022 5:00 PM	INDEX File	47 KB
ckpt-9.data-00000-of-00001	6/4/2022 5:07 PM	DATA-00000-OF-0...	20,279 KB
ckpt-9.index	6/4/2022 5:07 PM	INDEX File	47 KB
ckpt-10.data-00000-of-00001	6/4/2022 5:13 PM	DATA-00000-OF-0...	20,279 KB
ckpt-10.index	6/4/2022 5:13 PM	INDEX File	47 KB
ckpt-11.data-00000-of-00001	6/4/2022 5:20 PM	DATA-00000-OF-0...	20,279 KB
ckpt-11.index	6/4/2022 5:20 PM	INDEX File	47 KB
pipeline.config	5/30/2022 10:56 AM	XML Configuration...	5 KB

Gambar 4.6. Hasil save model *train* dalam bentuk file .ckpt

Pada Gambar 4.6 merupakan kumpulan *save* file pada .ckpt, dengan titik *start ckpt-1* pada setiap 1.000 *step* akan memperbarui *save* file .ckpt hingga berakhir pada file .ckpt-11. Perlu diketahui juga bahwa file .ckpt hanya menyimpan 7 file .ckpt terakhir. Yang akan digunakan sebagai model untuk memprediksi pada *webcam* adalah *save* file terakhir yaitu file .ckpt-11.

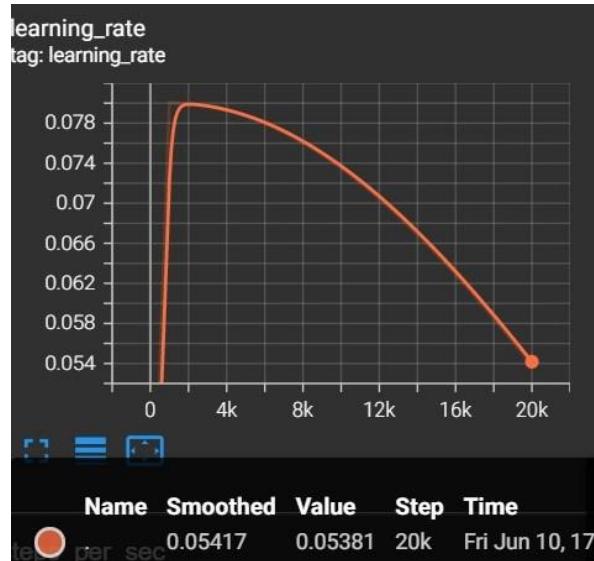
C. Training Dataset untuk 20.000 Step

Pada Gambar 4.7 merupakan hasil pengujian *loss/total_loss* *training dataset* sebanyak 20.000 *step* dengan nilai *total_loss* sebesar 0,1387 pada *loss/error*.



Gambar 4.7. Hasil uji *training total_loss* pada Tensorboard

Pada Gambar 4.8 nilai *learning_rate* mendapatkan hasil akhir pada *step* ke 20.000 sebesar 0,0538. Ketika memasuki *step* ke 3.000 dan terus menurun dengan drastis, *learning rate* memasuki tahap *learning decay* yang menyebabkan *learning rate* mengalami penurunan dan memperlambat training pada *step* sebagai ganti melakukan *smoothing* pada grafik Gambar 4.7 yang terlihat pada saat *step* ke 3.000 dan 10.000 mengalami lonjakan atau *overshoot* pada nilai *loss*.



Gambar 4.8. Hasil uji *training dataset learning_rate* pada Tensorboard

A screenshot of a file explorer window. The left pane shows a tree view with a yellow folder icon labeled 'train' and a grey folder icon labeled 'checkpoint'. The right pane lists files and folders:

train	6/10/2022 3:44 PM	File folder	
checkpoint	6/10/2022 5:54 PM	File	1 KB
ckpt-15.data-00000-of-00001	6/10/2022 5:16 PM	DATA-00000-OF-0...	20,279 KB
ckpt-15.index	6/10/2022 5:16 PM	INDEX File	47 KB
ckpt-16.data-00000-of-00001	6/10/2022 5:22 PM	DATA-00000-OF-0...	20,279 KB
ckpt-16.index	6/10/2022 5:22 PM	INDEX File	47 KB
ckpt-17.data-00000-of-00001	6/10/2022 5:28 PM	DATA-00000-OF-0...	20,279 KB
ckpt-17.index	6/10/2022 5:28 PM	INDEX File	47 KB
ckpt-18.data-00000-of-00001	6/10/2022 5:35 PM	DATA-00000-OF-0...	20,279 KB
ckpt-18.index	6/10/2022 5:35 PM	INDEX File	47 KB
ckpt-19.data-00000-of-00001	6/10/2022 5:41 PM	DATA-00000-OF-0...	20,279 KB
ckpt-19.index	6/10/2022 5:41 PM	INDEX File	47 KB
ckpt-20.data-00000-of-00001	6/10/2022 5:48 PM	DATA-00000-OF-0...	20,279 KB
ckpt-20.index	6/10/2022 5:48 PM	INDEX File	47 KB
ckpt-21.data-00000-of-00001	6/10/2022 5:54 PM	DATA-00000-OF-0...	20,279 KB
ckpt-21.index	6/10/2022 5:54 PM	INDEX File	47 KB

Gambar 4.9. Hasil *save model train* dalam bentuk file .ckpt

Pada Gambar 4.9 merupakan kumpulan *save* file pada .ckpt, dengan titik *start ckpt-1* pada setiap 1.000 *step* akan memperbarui *save* file .ckpt hingga berakhir pada file .ckpt-21. Diketahui juga bahwa file .ckpt hanya menyimpan 7 file .ckpt terakhir. Yang digunakan sebagai model untuk memprediksi pada *webcam* adalah *save* file terakhir yaitu file .ckpt-21.

4.1.3 Hasil kesimpulan *training dataset* model MobileNet-SSD

Pada Tabel 4.1 merupakan rangkuman yang berisi tabel tersebut berupa nilai *loss* setiap model, nilai akhir *learning_rate* dari awalan input sebesar 0,08 dan juga save file .ckpt terakhir yang digunakan sebagai *testing* model. Dapat disimpulkan bahwa batas untuk mendapatkan model dengan nilai loss paling baik yaitu 0,1238 adalah *training* hingga 10.000 *step*. Dikarenakan pada saat melebihi 10.000 *step*, *learning rate* mengalami penurunan hingga lebih dari 0,0100, yang berarti pembelajaran pada model mulai tidak efektif untuk menurunkan nilai loss. Proses *learning decay* perlu menguatkan lagi dengan menurunkan lebih banyak *learning rate* dan membuat *training* menjadi lebih lama satu setengah jam dari model 5.000 dan 10.000 *step*.

Tabel 4.1. Hasil kesimpulan dari ketiga model *train*

Model <i>Train</i>	Nilai <i>loss</i>	<i>Learning_rate</i>	Save file .ckpt	Keterangan
5.000	0,2712	0,0786	.ckpt-6	Pada saat memasuki <i>step</i> ke 4.000 nilai <i>loss</i> mengalami kenaikan pada grafik dari 0,25 hingga 0,45 dan mulai menjalankan proses <i>learning decay</i> dengan menurunkan nilai <i>learning rate</i> sebesar 0,0014 yang digunakan untuk mengembalikan nilai <i>loss</i> hingga mencapai nilai akhir <i>loss</i> sebesar 0,27.
10.000	0,1238	0,0732	.ckpt-11	Pada saat memasuki <i>step</i> ke 4.000, 5.000 dan 9.000 nilai <i>loss</i> mengalami kenaikan pada grafik dari 0,25 hingga 0,30. Proses <i>learning decay</i> mulai berjalan pada saat <i>step</i> ke 4.000, dengan menurunkan nilai <i>learning rate</i> hingga 0,0068 dan berhasil menurunkan nilai <i>loss</i> hingga 0,12.
20.000	0,1387	0,0538	.ckpt-21	Model ini terdapat 2 lonjakan tinggi, yaitu saat memasuki <i>step</i> 3.000 dengan nilai 0,25 naik hingga 0,35. Proses <i>learning decay</i> aktif dan berhasil menurunkan nilai <i>loss</i> hingga 0,20 pada <i>step</i> 10.000. namun saat melebihi <i>step</i> 10.000 mengalami lonjakan lagi dari 0,20 hingga 0,35, nilai <i>learning rate</i> akan turun menurun sebesar 0,0262 untuk mengembalikan nilai <i>loss</i> kembali kedalam nilai 0,20

4.2 Pengujian Hasil Prediksi Model

Pengujian program deteksi simbol SIBI meliputi dari akurasi tiap simbol pada subjek yang di tes dalam bentuk nilai persentase hasil testing dan besar FPS yang didapatkan pada *webcam*

4.2.1 Tujuan Pengujian Hasil Prediksi Model Train

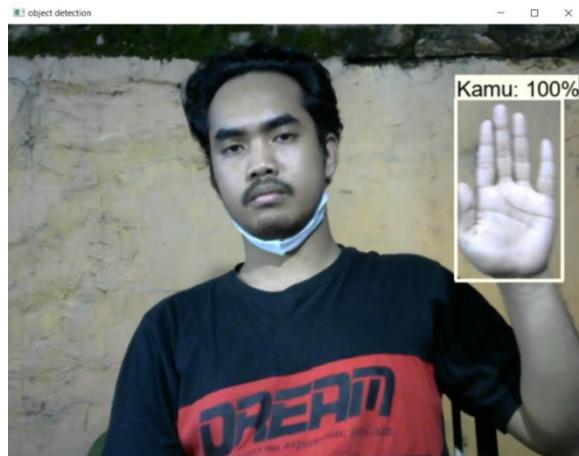
Tujuan pengujian ini adalah untuk mengetahui akurasi rata-rata dari prediksi yang dihasilkan pada model yang sudah di training dari 5.000, 10.000 dan 20.000 *step* terhadap 6 simbol SIBI

4.2.2 Prosedur Pengujian Hasil Prediksi Model Train

1. Setiap subjek model memperagakan 6 pose simbol SIBI.
2. Uji coba dilakukan dengan penerangan yang bagus dan menggunakan latar belakang yang sama.
3. Setiap subjek dilakukan uji percobaan video ke 1 untuk durasi 5 detik, 2 untuk durasi 10 detik dan 3 untuk durasi 15 detik.

4.2.3 Hasil Pengujian Deteksi Simbol dan FPS

Hasil pengujian didapatkan dari analisa video yang di uji pada setiap subjek lalu diambil akurasi yang paling tinggi pada saat durasi video tersebut.



Gambar 4.10 *Bounding box* dan akurasi tertinggi muncul pada uji video

A. Hasil pengujian deteksi simbol untuk model *train 5.000 step*

Pada Tabel 4.1. Dapat disimpulkan bahwa model *train* sebanyak 5.000 step mendapatkan nilai lebih dari 70% pada ke 6 pose simbol. Pada pengujian video juga terlihat *bounding box* dengan akurat dapat menemukan lokasi prediksi dari pose yang diperagakan oleh setiap subjek, terutama pada pose yang menggunakan 2 tangan yaitu “Cinta” dan “Sedih” dengan akurasi rata-rata sebesar 70,8 % dan 80,1 %. Namun dari ketiga model, model ini hanya memiliki 1 pose yang mencapai akurasi diatas 90% yaitu kamu dengan akurasi rata-rata sebesar 96%.

Frame Per Second yang didapatkan untuk setiap video pada *webcam* juga mendapatkan hasil rata-rata sebesar 17,8 FPS atau 18 FPS, yang dapat diartikan model tersebut dapat berjalan dengan lancar pada *webcam* dan sesuai yang diharapkan pada saat menggunakan model *pre-trained* MobileNet-SSDv2 untuk membuat proses komputasinya menjadi ringan.

Tabel 4.2 Hasil pengujian deteksi simbol pada model *train 5.000 step*

Subjek uji ke-	Uji video percobaan ke-	DETEKSI SIMBOL BAHASA ISYARAT						FPS
		Aku	Kamu	Dia	Cinta	Maaf	Sedih	
Marshel	1	66	93	57	72	55	73	17
	2	74	99	41	60	53	72	18
	3	80	95	43	65	86	69	17
Rifki	1	94	92	70	37	95	91	19
	2	85	82	81	40	94	95	18
	3	73	92	92	51	96	95	17
Akbar	1	81	97	94	90	96	93	18
	2	93	98	91	82	95	94	17
	3	97	98	91	93	98	93	19
Frans	1	98	100	97	84	100	64	19
	2	98	100	100	78	99	82	18
	3	98	100	99	71	100	85	17
Angie	1	95	98	91	73	96	92	17
	2	90	98	85	77	96	94	17
	3	98	98	97	89	97	95	19
Rata-rata		88	96	81,2	70,8	83,7	80,1	17,8

B. Hasil pengujian deteksi simbol untuk model *train 10.000 step*

Pada Tabel 4.2. Dapat disimpulkan juga bahwa dari 6 pose simbol SIBI terdapat 2 pose simbol yang memiliki rata-rata akurasi yang buruk yaitu simbol ‘Cinta’ dan ‘Sedih’. Jika dibandingkan dengan 4 simbol lainnya, 2 pose tersebut menggunakan 2 tangan sebagai bentuk pose yang digunakan, pada pengambilan citra dataset juga pose tersebut selalu berada pada posisi yang dapat bertabrakan dengan anggota tubuh lainnya seperti tubuh dan wajah. Maka dari itu, bahwa 2 pose tersebut hasil akurasi nya menurun bisa disebabkan dari posisi pose gerakan yang bersama anggota tubuh lainnya atau bisa juga dikarenakan pada saat melakukan *labeling box* kepada 2 simbol tersebut terlalu besar, jika dibandingkan dengan 4 simbol lainnya yang hanya sampai pergelangan tangan.

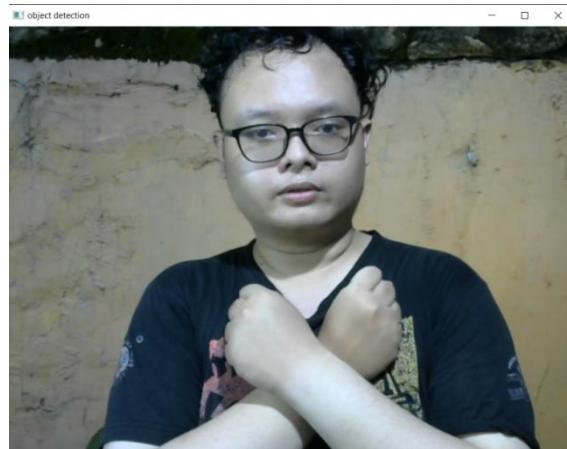
Tabel 4. 3 Hasil pengujian deteksi simbol pada model *train 10.000 step*

Subjek uji ke-	Uji video percobaan ke-	DETEKSI SIMBOL BAHASA ISYARAT						FPS
		Aku	Kamu	Dia	Cinta	Maaf	Sedih	
Marshel	1	82	97	98	58	99	51	18
	2	96	96	98	41	99	78	18
	3	98	99	99	41	99	91	19
Rifki	1	97	100	95	69	99	94	17
	2	96	99	97	78	99	92	19
	3	99	100	99	65	99	98	17
Akbar	1	98	98	94	56	99	48	17
	2	97	99	97	48	99	44	18
	3	95	100	98	54	99	52	18
Frans	1	97	99	94	91	98	45	19
	2	98	98	97	50	97	58	17
	3	97	98	96	58	100	55	19
Angie	1	84	99	92	31	97	52	18
	2	90	99	91	37	97	41	18
	3	78	98	81	23	97	66	19
Rata-rata		93,4	98,6	95	53,3	98,4	63,3	18,2

Frame Per Second yang didapatkan untuk setiap video pada *webcam* juga mendapatkan hasil rata-rata sebesar 18.2 FPS atau 18 FPS, yang dapat diartikan model tersebut dapat berjalan dengan lancar pada *webcam* dan sesuai yang diharapkan pada saat menggunakan model *Pre-trained* MobileNet-SSDv2 untuk membuat proses komputasinya menjadi ringan.

C. Hasil pengujian deteksi simbol untuk model *train 20.000 step*

Pada Tabel 4.3 Dapat disimpulkan bahwa pada model *training* dengan 20.000 step mendapatkan kenaikan pada statistik rata-rata akurasi. Namun pada pose cinta masih mendapatkan nilai akurasi rata-rata sebesar 59.4%, lalu pada saat uji video pada Rifki didapatkan 0% pada akurasi yang dikarenakan *bounding box* tidak muncul sama sekali pada ketiga uji video seperti pada Gambar 4.10. Hal tersebut dikarenakan adanya kemungkinan *overfitting* pada model *training*. *Overfitting* adalah keadaan dimana data *training* mendapat akurasi tinggi namun memiliki akurasi rendah pada prediksi. Hal tersebut didukung juga pada setiap pengujian video, *bounding box* selalu menghilang ketika subjek sedikit berbeda dari pose saat memperagakannya



Gambar 4.11 *Bounding box* tidak muncul pada uji video

Rata-rata akurasi yang didapatkan pada model ini membaik hingga mencapai tingkat akurasi tertinggi dari 2 model lainnya, namun keakuratan *bounding box* untuk memprediksi gerakan pada uji video melemah.

Tabel 4.4 Hasil pengujian deteksi simbol pada model *train 20.000 step*

Subjek uji ke-	Uji video percobaan ke-	DETEKSI SIMBOL BAHASA ISYARAT						FPS
		Aku	Kamu	Dia	Cinta	Maaf	Sedih	
Marshel	1	76	76	70	66	65	66	19
	2	89	93	73	85	90	82	19
	3	74	89	90	91	98	76	18
Rifki	1	84	99	99	0	99	90	18
	2	92	100	99	0	100	96	17
	3	98	100	90	0	100	95	17
Akbar	1	61	100	99	33	100	77	18
	2	71	100	97	90	100	85	18
	3	99	100	99	98	100	76	17
Frans	1	98	100	66	92	100	91	18
	2	99	100	99	90	100	96	18
	3	99	100	99	86	100	96	18
Angie	1	94	100	99	28	100	70	19
	2	98	100	98	89	100	78	19
	3	98	100	91	88	100	97	18
Rata-rata		88,6	97,1	93,2	56,4	96,7	84,7	18,1

Frame Per Second didapatkan untuk setiap percobaan pada *webcam* mendapatkan hasil rata-rata sebesar 18.1 FPS atau 18 FPS, yang dapat diartikan model tersebut dapat berjalan dengan lancar pada *webcam* dan sesuai yang diharapkan pada saat menggunakan model *pre-trained* MobileNet-SSDv2 untuk membuat proses komputasi nya menjadi ringan.

4.2.4 Hasil kesimpulan pengujian deteksi simbol dan FPS

Berdasarkan Tabel 4.5 menunjukan hasil *testing* video pada model 5.000 *step* memiliki akurasi *testing* yang paling rendah yaitu 83,3% namun proses *bounding box* mampu mengikuti gerakan pose dengan baik dibandingkan dengan model 20.000 *step* yang memiliki akurasi testing paling tinggi yaitu 86,6% namun pose juga harus menyerupai bentuk pose model, ketika pose sudah tidak mirip *bounding box* sudah tidak dapat memprediksi pose tersebut.

Tabel 4.5 Hasil kesimpulan pengujian deteksi simbol dan FPS

Model Train	Rata-rata akurasi testing	Rata-rata nilai FPS
5.000	83,3	18
10.000	83,6	18
20.000	86,6	18

Nilai FPS untuk ketiga model memiliki rata-rata nilai sebesar 18 FPS. dimana nilai tersebut dalam *realtime detection* sudah termasuk besar dimana jika dibawah nilai 10 FPS untuk *realtime detection* merupakan nilai yang rendah. Pada saat pemilihan *pre-trained* model, MobileNet-SSD dipilih dikarenakan mampu memiliki nilai FPS rata-rata dari 10 FPS hingga 30 FPS.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil analisa dan pembahasan pada bab 4, dapat disimpulkan sebagai berikut:

1. Hasil *training* dari model *Pre-Trained* MobileNet-SSDv2 menghasilkan nilai *total loss* sebesar 0,2712 pada *training 5.000 step*, 0,1881 pada *training 10.000 step* dan 0,1387 pada *training 20.000 step*.
2. Hasil *training* dari ketiga model *Pre-Trained* MobileNet-SSDv2 memiliki nilai rata-rata akurasi untuk keseluruhan simbol pose dengan model *train 20.000 step* yang mempunyai rata-rata akurasi *testing* paling tinggi yaitu 86,6% di ikuti dengan model *train 10.000 step* yaitu 83,6% dan paling rendah dengan model *train 5.000 step* yaitu 83,3%.
3. Hasil dari pengujian FPS untuk ketiga model mendapatkan nilai 18 FPS, dimana nilai tersebut sudah dapat dianggap cukup lancar karena model pre-trained MobileNet-SSD diharapkan mendapat nilai 10 hingga 30 FPS.
4. Hasil dari pengujian akurasi rata-rata deteksi setiap 6 simbol untuk 5.000 *step* yaitu ‘Aku’ sebesar 88%, ‘Kamu’ sebesar 96%, ‘Dia’ sebesar 81,2%, ‘Cinta’ sebesar 70,8%, ‘Maaf’ sebesar 83,7% dan ‘Sedih’ sebesar 80,1%.
5. Hasil dari pengujian akurasi rata-rata deteksi setiap 6 simbol untuk 10.000 *step* yaitu ‘Aku’ sebesar 93,4%, ‘Kamu’ sebesar 98,6%, ‘Dia’ sebesar 95% , ‘Cinta’ sebesar 53,3%, ‘Maaf’ sebesar 98,4% dan ‘Sedih’ sebesar 63,3%.
6. Hasil dari pengujian akurasi rata-rata deteksi setiap 6 simbol untuk 20.000 *step* yaitu ‘Aku’ sebesar 88,6%, ‘Kamu’ sebesar 97,1%, ‘Dia’ sebesar 93,2% , ‘Cinta’ sebesar 56,4%, ‘Maaf’ sebesar 96,7% dan ‘Sedih’ sebesar 84,7%.
7. Hasil dari pengujian deteksi simbol mendapatkan bahwa simbol untuk ‘Cinta’ memiliki akurasi rata-rata paling tidak baik untuk 5.000, 10.000 dan 20.000 yaitu: 70,8%, 53,3% dan 56,4% hal ini disebabkan dari pose tersebut berdekatan dengan baju dan mendekati leher dan mulut sehingga model dapat

- kebingungan bila baju yang digunakan terdapat garis-garis atau huruf atau terlalu dekat dengan wajah.
8. Hasil dari pengujian deteksi simbol mendapatkan bahwa simbol untuk ‘Kamu’ memiliki akurasi rata-rata paling baik untuk 5.000, 10.000 dan 20.000 yaitu: 96%, 98,6% dan 97,1% hal ini disebabkan dari pose tersebut menampilkan telapak tangan yang tidak miring dan datar kepada webcam. Bahkan untuk testing akurasi pada tangan kiri dan kanan masih dapat terdeteksi.
 9. Pada pengujian untuk deteksi simbol dan FPS, terdapat beberapa permasalahan yang di akibatkan oleh cahaya yang minim dapat mempengaruhi proses deteksi dan klasifikasi pose simbol SIBI terutama pada simbol yang membutuhkan 2 tangan seperti ‘Sedih’ dan ‘Cinta’.

5.2 Saran

Pada saat penelitian selanjutnya diharapkan dapat mengembangkan dan menyempurnakan kekurangan dari Tugas Akhir ini dengan menerapkan beberapa petunjuk yang diberikan penulis yaitu:

1. Mencoba untuk mengimplementasikan *keypoint landmark* dari MediaPipe yang digunakan untuk lebih memudahkan model agar klasifikasi dan deteksi saat *realtime* berjalan dengan mulus dan bagus.
2. Pemilihan *dataset* yang menggunakan *green screen* dan memiliki varian pencahayaan dari redup hingga cerah, supaya pada kondisi apapun model tetap dapat mendeksi dengan bagus dan baik.
3. Penggunaan *device* laptop atau PC jika menggunakan *graphic card* NVIDIA akan membutuhkan CUDA dan CUDNN untuk melakukan *training* pada *realtime object detection*, maka dari itu disarankan mempunyai spesifikasi *graphic card* yang memiliki VGA RAM 6 GB atau menggunakan Google Colab Pro+.

DAFTAR PUSTAKA

- Anaconda. (2022). *anaconda.com*. Retrieved Juni 15, 2022, from anaconda.com:
<https://www.anaconda.com/>
- Anam, N. (2022). *Sistem Deteksi Simbol pada SIBI (Sistem Isyarat Bahasa Indonesia) menggunakan MediaPipe dan ResNet-50*. Surabaya: Universitas Dinamika.
- Choudhury, A. (2020, Mei 20). <https://analyticsindiamag.com/>. Retrieved Juni 15, 2022 , from What Is The Difference Between Image Classification & Object Detection Techniques?: <https://analyticsindiamag.com/what-is-the-difference-between-image-classification-object-detection-techniques/>
- LINA, Q. (2019, Januari 2). *Apa itu Convolutional Neural Network?* Retrieved Maret 25, 2022, from Medium: <https://medium.com/@16611110/apa-itu-convolutional-neural-network-836f70b193a4>
- Liu, W. (2015, Desember 8). *arxiv.org*. Retrieved Maret 25, 2022, from SSD: Single Shot MultiBox Detector: <https://arxiv.org/abs/1512.02325>
- Nasir, M. C. (2021). PENGGUNAAN SISTEM ISYARAT BAHASA INDONESIA (SIBI). *Prodi Ilmu Komunikasi, FISIP, Untag Surabaya*, 1-7.
- Putra, I. R. (2021). *Sistem Deteksi Simbol Pada SIBI (Sistem Isyarat Bahasa Indonesia) Menggunakan Convolutional Neural Network*. Surabaya: Universitas Dinamika.
- Riadi, M. (2018). Perancangan Aplikasi Communication Board Berbasis Android Tablet Sebagai Media Pembelajaran dan Komunikasi Bagi Anak Tuna Rungu. *Pengembangan Teknologi Informasi dan Ilmu Komputer*, 2933-2943.
- Sandler, M. (2019, Maret 21). *arxiv.org*. Retrieved Maret 25, 2022, from MobileNetV2: Inverted Residuals and Linear Bottlenecks: <https://arxiv.org/abs/1801.04381v4>
- Sindy, F. (2019). PENDETEKSIAN OBJEK MANUSIA SECARA REAL TIME DENGAN. In F. Sindy, *PENDETEKSIAN OBJEK MANUSIA SECARA REAL TIME DENGAN* (pp. 14-15). Medan: UNIVERSITAS SUMATERA UTARA.

- Syahrudin, A. N. (2018). INPUT DAN OUTPUT PADA BAHASA. *Jurnal Dasar Pemograman Python STMIK*, 1-7.
- Tzutalin. (2015). *github.com*. Retrieved Juni 10, 2022, from LabelImg: <https://github.com/tzutalin/labelImg>
- Wiranda, N. (2019). Model Identifikasi Kata Ucapan Tuna Wicara. *Indonesian Journal of Electronics and Instrumentation Systems* , 131-140.
- Y, S. (2021). Sign Language Recognition System using Transfer. *Artificial & Computational Intelligence*.
- Zhu, M. (2017, April 17). *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications*. Retrieved Maret 25, 2022, from arxiv: <https://arxiv.org/abs/1704.04861>

