



**UNIVERSITAS INDONESIA**

**RANCANG BANGUN SISTEM PENGENALAN BAHASA  
ISYARAT MENGGUNAKAN ALGORITMA YOLOV5 DAN  
SSD**

**Bryan Dario Lesmana**

**1806199940**

**FAKULTAS TEKNIK  
PROGRAM STUDI TEKNIK KOMPUTER**

**DEPOK  
JUNI 2022**



**UNIVERSITAS INDONESIA**

**DEVELOPMENT OF SIGN LANGUAGE RECOGNITION  
SYSTEM USING YOLOV5 AND SSD ALGORITHMS**

**SKRIPSI**

**Diajukan sebagai salah satu syarat untuk memperoleh gelar  
Sarjana Teknik**

**Bryan Dario Lesmana**

**1806199940**

**FAKULTAS TEKNIK  
PROGRAM STUDI TEKNIK KOMPUTER**

**DEPOK  
JUNI 2022**

## HALAMAN PERNYATAAN ORISINALITAS

**Skripsi ini adalah hasil karya saya sendiri,  
dan semua sumber baik yang dikutip maupun dirujuk  
telah saya nyatakan dengan benar.**

**Nama : Bryan Dario Lesmana**

**NPM : 1806199940**

**Tanda Tangan : **

**Tanggal : 24 JUNI 2022**

## LEMBAR PENGESAHAN

Skripsi ini diajukan oleh:

Nama : Bryan Dario Lesmana

NPM : 1806199940

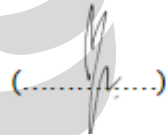
Program Studi : Teknik Komputer

Judul Skripsi : Rancang Bangun Sistem Pengenalan Bahasa Isyarat  
Menggunakan Algoritma YOLOv5 dan SSD

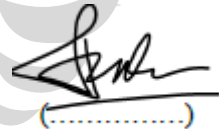
**Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagaibagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana mata kuliah Skripsi pada Program Studi Teknik Komputer Fakultas Teknik Universitas Indonesia.**

### DEWAN PENGUJI

Pembimbing : Dr. Eng. Mia Rizkinia, S.T., M.T



Penguji : Yan Maraden, S.T., M.T., M.Sc.



Penguji : I Gde Dharma Nugraha, S.T., M.T., Ph.D



Ditetapkan di : Fakultas Teknik

Tanggal : 15 Juli 2022

## KATA PENGANTAR

Puji syukur dipanjatkan kepada Tuhan Yesus Kristus karena berkat-Nya penulis dapat menyelesaikan buku skripsi yang berjudul “Rancang Bangun Sistem Pengenalan Bahasa Isyarat Menggunakan Algoritma YOLOv5 dan SSD” Buku skripsi ini disusun sebagai salah satu syarat kelulusan dari jenjang Pendidikan S1 pada Program Studi Teknik Komputer Fakultas Teknik Universitas Indonesia. Dalam mengerjakan buku skripsi, penulis mendapatkan bimbingan, bantuan, dan dukungan dari berbagai pihak. Penulis ingin menyampaikan ucapan terima kasih kepada seluruh pihak yang telah memberikan bantuan dan arahan dalam proses pelaksanaan penulisan buku :

1. Ibu Mia Rizkinia, S.T, M.T selaku dosen pembimbing yang telah membimbing dan memberikan arahan, ilmu, saran dan kritik serta menyemangati penulis sehingga penulis dapat menyelesaikan buku skripsi.
2. Kedua Orang tua dan seluruh anggota keluarga penulis yang telah memberikan dukungan dan semangat selama proses penulisan buku skripsi.
3. Nisrina Athallah, Suryanegara, Nathaniel Faustine, Leonardus Kevin, dan Fadly Ahmad Firdausy selaku teman seperbimbingan yang telah memberi bantuan kepada penulis.
4. Seluruh teman – teman Teknik Komputer 2018 yang telah menemani pembimbing selama masa perkuliahan dan memberikan hiburan, bantuan, dan ilmu kepada penulis,
5. Seluruh teman – teman Departemen Teknik Elektro yang telah membantu penulis dan memberikan bimbingan serta ilmu selama masa perkuliahan.

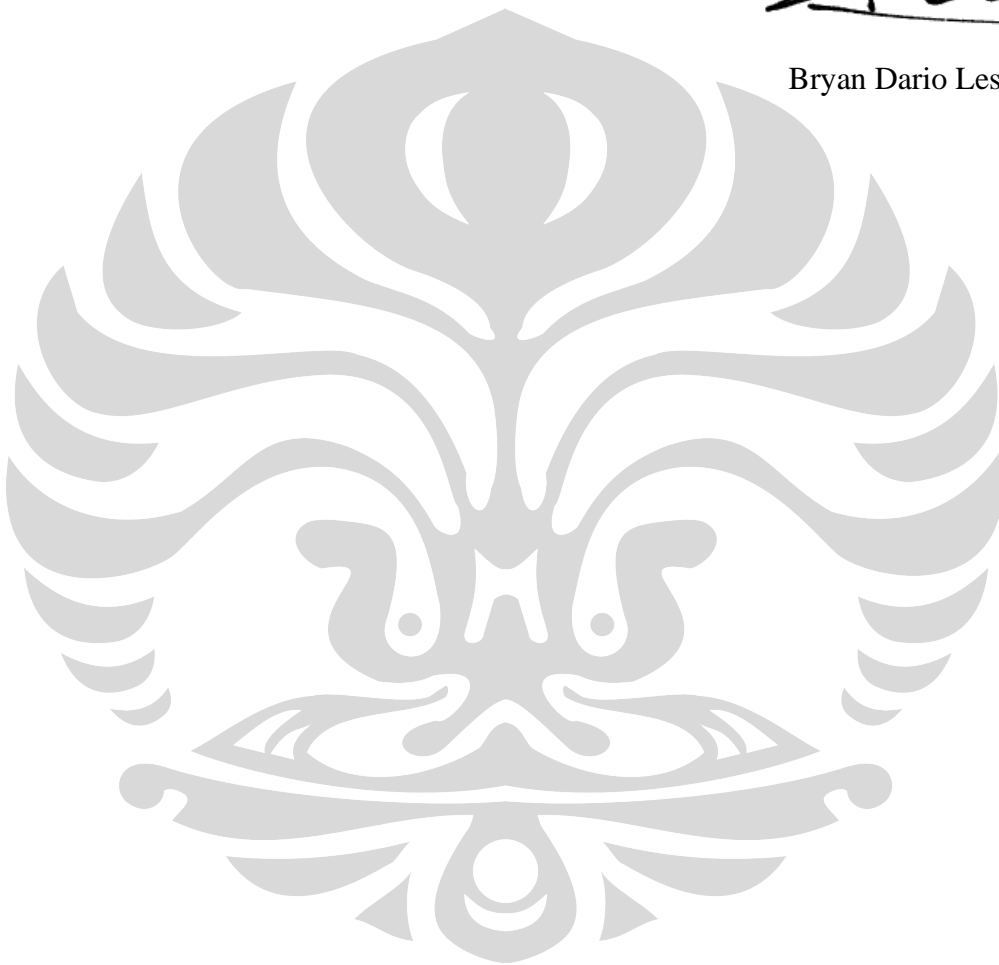
Penulis mengharapkan buku skripsi ini dapat bermanfaat bagi pembaca, dalam laporan ini masih sangat banyak kekurangan maka dari itu seluruh saran dan

kritik yang membantu sangat diharapkan sebagai pembelajaran dan evaluasi penulis untuk kedepannya.

Depok, Juni 2022



Bryan Dario Lesmana



**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI KARYA  
ILMIAH KEPENTINGAN AKADEMIS**

Sebagai sivitas akademik Universitas Indonesia, saya yang bertandatangan di bawah ini:

Nama : Bryan Dario Lesmana  
NPM : 1806199940  
Program Studi : Teknik Komputer  
Fakultas : Fakultas Teknik  
Jenis Karya : Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia **Hak Bebas Royalti Noneksklusif (Non-exclusive Royalty-Free Right)** atas karya ilmiah saya yang berjudul:

*RANCANG BANGUN SISTEM PENGENALAN BAHASA ISYARAT  
MENGUNAKAN ALGORITMA YOLOV5 DAN SSD*

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (database), merawat, dan memublikasikan tugas akhir saya tanpa meminta izin dari saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok

Pada tanggal : 24 Juni 2022

Yang menyatakan



Bryan Dario Lesmana

## ABSTRAK

Nama : Bryan Dario Lesmana  
Program Studi : Teknik Komputer  
Judul : Rancang Bangun Sistem Pengenalan Bahasa Isyarat  
Menggunakan Algoritma YOLOv5 dan SSD

Bahasa Isyarat adalah bahasa yang digunakan kebanyakan oleh kaum tuna rungu dan tuna wicara yang tidak bisa berkomunikasi secara audio, hal ini menimbulkan kesenjangan dalam berkomunikasi terlebih dalam kemampuan tuna rungu dan tuna wicara dalam melaksanakan kehidupannya sehari – hari khususnya saat ingin melamar kerja. Penelitian ini dilakukan untuk mempermudah komunikasi antara pengguna bahasa isyarat dengan mereka yang tidak memahami bahasa isyarat, dimana dengan adanya sistem ini maka translasi bahasa isyarat ke Bahasa Indonesia akan dilakukan secara otomatis dan ini akan membantu bagaimana kaum tuna wicara dan tuna rungu berkomunikasi dengan mereka yang tidak memahami bahasa isyarat sehingga ini akan berdampak dimana para pengguna bahasa isyarat bisa memiliki kesempatan yang sama dengan semua orang dalam proses pelamaran kerja dan mendapatkan pekerjaan yang layak. Sistem pengenalan bahasa isyarat ini bekerja dengan menerima bahasa isyarat yang disampaikan oleh seseorang secara *real-time* dan kemudian mengenalinya sebagai arti kata dari bahasa isyarat tersebut ke Bahasa Indonesia. Pada penelitian ini OpenCV digunakan sebagai metode pengambilan gambar dalam waktu nyata, serta algoritma YOLOv5 yang dibandingkan dengan SSD yang digunakan untuk memproses gambar tersebut serta menandakan yang mana objek yang dianggap sebagai bahasa isyarat dan mendeteksi artinya. Proses training dilakukan dengan dataset yang terdiri dari 463 citra training yang kemudian diaugmentasi sehingga berjumlah 1389 citra training. Model yang dihasilkan dari setiap algoritma yang digunakan dalam penelitian diuji menggunakan dataset testing lalu akan diuji dalam tahap *real-time* testing dan parameter yang digunakan untuk evaluasi kedua hasil model adalah akurasi atau (*confidence score*) sistem, *precision*, *recall*, dan *F1 Score* untuk masing – masing model dimana nilai perbandingan untuk nilai *confidence score* model YOLOv5 dan SSD adalah 100% : 87.66%. Sedangkan perbandingan nilai *F1 Score* untuk model YOLOv5 dan SSD adalah 1 : 0.9342. Penelitian ini juga menunjukkan bahwa *Learning Rate* dari Model SSD lebih tinggi dibanding Model YOLOv5 yaitu 0.08 : 0.009. Pada penelitian ini ditunjukkan bahwa algoritma YOLOv5 akan memiliki hasil presisi yang lebih baik dibandingkan algoritma SSD.

Kata Kunci:

Bahasa Isyarat; waktu nyata; gambar; SSD; YOLOv5



## ABSTRACT

Name : Bryan Dario Lesmana  
Study Program : Computer Engineering  
Title : Development of Sign Language Recognition System Using YOLOv5 and SSD Algorithms

Sign Language is the method used mostly by deaf and mute people which are unable to communicate by audio . This difference in the way of communicating between each other creates a gap in communicating between the deaf and mute with normal people. This research is done with the intent to further make the communication between sign language user those that do not understand sign language by automatically translating the meaning of each sign language to Bahasa Indonesia. By doing this, this will ensure to help the mute and deaf people so that they will have the same opportunity to apply for a job just like those without disability. Sign language recognition works by detecting object in real time using camera and then recognize the sign made and show the meaning of that particular sign. This is made possible using OpenCV to take images in real time and the model SSD and YOLOv5 to process those images and label them using rectangular show which object on the picture is the sign language that needs to be recognized based on the available dataset which is the sign language that have already been taken before the test. The training process of this research is done using 463 training images which then augmented and becomes 1389 training images. The models created from training using both algorithms will be tested using testing images and then further tested using real-time testing and the parameter used for evaluation of those models are the confidence score of the system accuracy, precision, recall, and F1 Score which from this research shows that the comparison of confidence score of the system accuracy between YOLOv5 model and SSD model is 100% : 87.66%. On the other hand, the comparison of the F1 Score between YOLOv5 model and SSD model is 1 : 0,9342. This research shows that YOLOv5 model have better learning rate compared to SSD which is 0.08 : 0.009. The result from this research shows that YOLOv5 algorithm will have better score of precision compared to SSD algorithm.

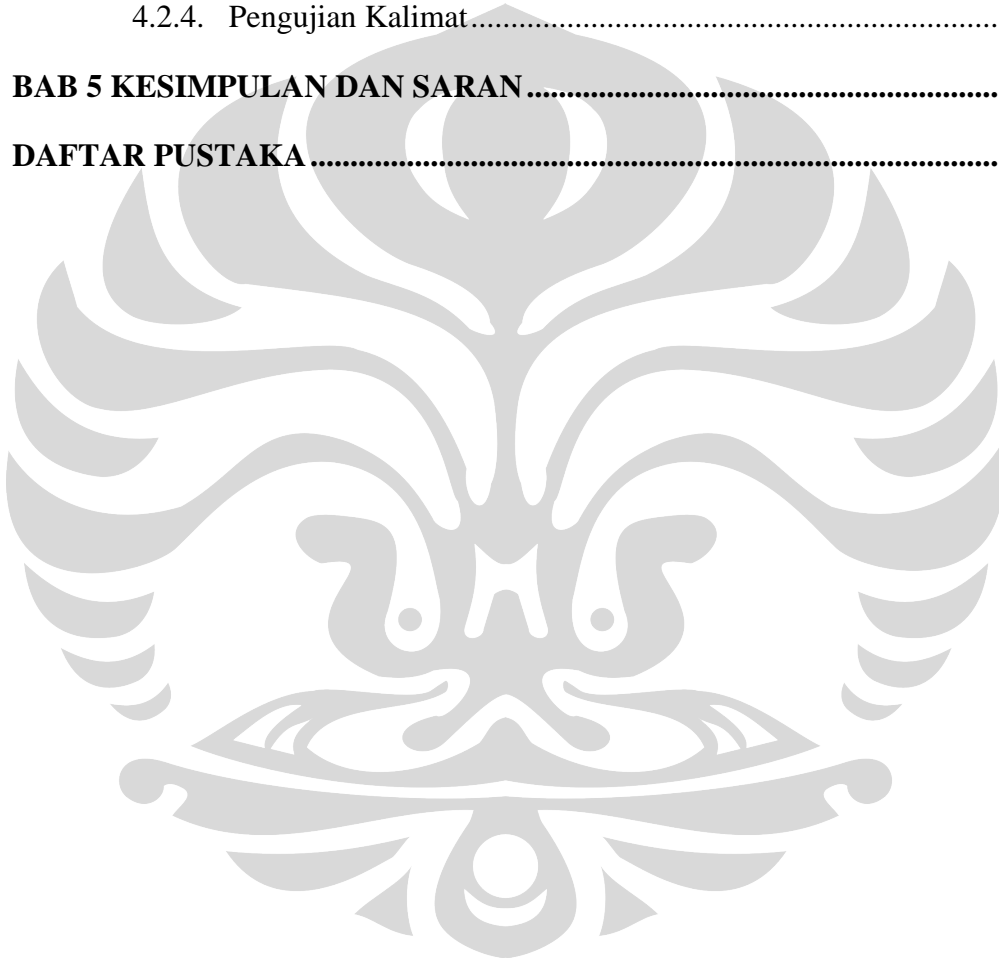
Key words:

*Sign Language; real time; images; SSD; YOLOv5*

## DAFTAR ISI

HALAMAN PERNYATAAN ORISINALITAS .....	ii
LEMBAR PENGESAHAN .....	iii
KATA PENGANTAR.....	iv
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH KEPENTINGAN AKADEMIS .....	vi
ABSTRAK .....	vii
ABSTRACT .....	viii
DAFTAR ISI.....	ix
DAFTAR GAMBAR.....	xi
DAFTAR TABEL .....	13
<b>BAB 1 PENDAHULUAN .....</b>	<b>14</b>
1.1. Latar Belakang .....	14
1.2. Tujuan Penelitian .....	16
1.3. Manfaat Penelitian .....	16
1.4. Batasan Masalah.....	16
1.5. Sistematika Penulisan .....	17
<b>BAB 2 LANDASAN TEORI .....</b>	<b>19</b>
2.1. Bahasa Isyarat .....	19
2.1.1. Bahasa Isyarat di Indonesia.....	21
2.2. Machine Learning .....	23
2.3. Deep Learning.....	23
2.4. OpenCV .....	24
2.5. Tensorflow .....	25
2.6. SSD .....	25
2.7. You Only Look Once .....	27
2.8. <i>Precision, Recall, dan F1 Score</i> .....	28
<b>BAB 3 PERANCANGAN SISTEM.....</b>	<b>30</b>
3.1. Diagram Alur .....	30
3.2. Dataset.....	32
3.2.1. Pengumpulan Dataset.....	32
3.2.2. <i>Preprocessing</i> .....	34
3.3. Perancangan Model.....	36

3.4.	Skenario Pengujian.....	37
3.5.	Parameter Evaluasi.....	38
3.5.1.	Evaluasi Akurasi, <i>Precision</i> , <i>Recall</i> , dan <i>F1 Score</i> Model.....	38
<b>BAB 4 EVALUASI DAN ANALISIS HASIL PENELITIAN.....</b>		<b>40</b>
4.1.	Implementasi Sistem .....	40
4.2.	Hasil Penelitian .....	41
4.2.1.	Analisis <i>Loss</i> dan <i>Learning Rate</i> Model .....	41
4.2.2.	Analisis Grafik Total Loss .....	44
4.2.3.	Hasil Percobaan <i>Testing</i> .....	47
4.2.4.	Pengujian Kalimat.....	54
<b>BAB 5 KESIMPULAN DAN SARAN .....</b>		<b>59</b>
<b>DAFTAR PUSTAKA .....</b>		<b>61</b>



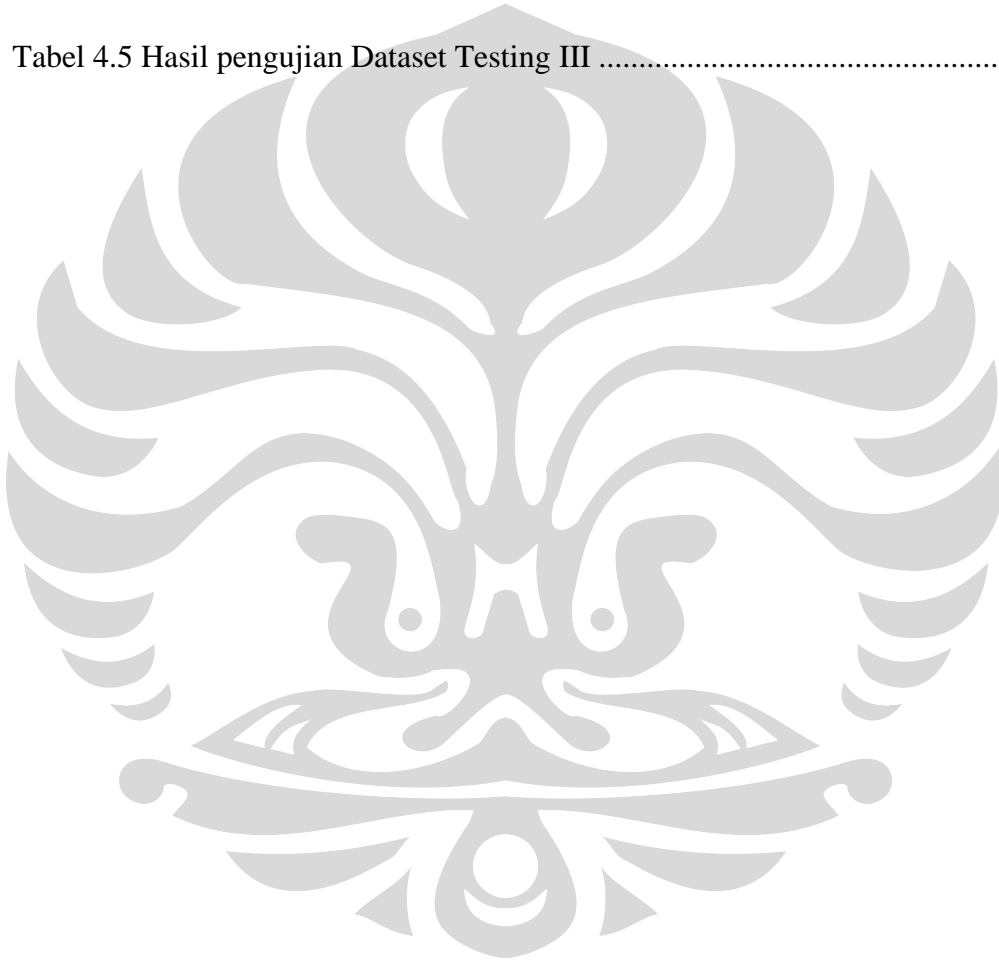
## DAFTAR GAMBAR

Gambar 2.1 Bahasa Isyarat Dari Seluruh Dunia [3] .....	21
Gambar 2.2 Abjad BISINDO (Kiri) dan SIBI (Kanan) [5].....	22
Gambar 2.3 Kata “Saya” (kiri) dan “Anda” (kanan) dalam SIBI [7].....	22
Gambar 2.4 MultiBox Detector [12] .....	25
Gambar 2.5 Perbandingan model antara SSD dan YOLO [12] .....	26
Gambar 2.6 YOLO <i>Detection System</i> [13] .....	27
Gambar 2.7 Seleksi Model YOLOv5 [15] .....	28
Gambar 3.1 Diagram Alur Penelitian.....	31
Gambar 3.2 Dataset Topik Pekerjaan "Ahli", “Apa”, “Daftar” (kiri ke kanan)[7]	33
Gambar 3.3 Dataset Topik Pekerjaan “Tidak”, “Ya”, “Saya”, “Rumah”, “Kerja” (kiri ke kanan)[7].....	33
Gambar 3.4 Dataset Topik Pekerjaan "Ahli", “Apa”, “Daftar” (kiri ke kanan .....	34
Gambar 3.5 Proses <i>Labelling</i> gambar menggunakan Labelimg .....	35
Gambar 4.1 Grafik Total Loss Training Model SSD MobileNet v2 FPNLite 320 x 320.....	44
Gambar 4.2 Grafik Total Loss Training dan Validasi Model YOLOv5s .....	45
Gambar 4.3 Perbandingan grafik Total Loss Training Model SSD MobileNet v2 FPNLite 320 x 320 vs YOLOv5s.....	46
Gambar 4.4 Percobaan deteksi objek “Ahli” dengan menggunakan Model SSD MobileNet v2 FPNLite 320 x 320 (atas) dan YOLOv5s (bawah) .....	47
Gambar 4.5 Dataset Testing I “Bulan”, “Lupa”, “Sini” pada model SSD MobileNetv2 FPNLite 320 x 320.....	49

Gambar 4.6 Dataset Testing II “Ingin” dan “Hari” (atas), “Lupa” dan “Sini”(bawah) pada model SSD MobileNetv2 FPNLite 320 x 320.....	51
Gambar 4. 7 Dataset Testing III “Anda”, “Hari”, “Lupa” pada model SSD MobileNetv2 FPNLite 320 x 320.....	53
Gambar 4.8 "Kerja" "Saya" "Cepat" (Model SSD MobileNet v2 FPNLite 320 x 320) .....	55
Gambar 4.9 "Kerja" "Saya" "Cepat" (Model YOLOv5s) .....	55
Gambar 4.10 "Saya" "Cepat" "Tanggap" (Model SSD MobileNet v2 FPNLite 320 x 320) .....	55
Gambar 4.11 "Saya" "Cepat" "Tanggap" (Model YOLOv5s) .....	56
Gambar 4.12 "Saya" "Orang" "-Nya" "Pe-" "Lupa" (Model SSD MobileNet v2 FPNLite 320 x 320).....	56
Gambar 4.13 "Saya" "Orang" "-Nya" "Pe-" "Lupa" (Model YOLOv5s) .....	57
Gambar 4.14 "Rumah" "Saya" "Jauh" (Model SSD MobileNet v2 FPNLite 320 x 320) .....	57
Gambar 4.15 "Rumah" "Saya" "Jauh" (Model YOLOv5s).....	58

## DAFTAR TABEL

Tabel 4.1 Evaluasi Hasil <i>Training</i> Model SSD FPNLite 320 x 320.....	41
Tabel 4.2 Evaluasi Hasil <i>Training</i> Model YOLOv5s .....	43
Tabel 4.3 Hasil Pengujian Dataset Testing I.....	48
Tabel 4.4. Hasil Pengujian Dataset Testing II.....	50
Tabel 4.5 Hasil pengujian Dataset Testing III .....	52



## BAB 1 PENDAHULUAN

### 1.1. Latar Belakang

Bahasa adalah suatu sarana utama yang digunakan oleh anggota masyarakat dalam berkomunikasi antara satu dengan yang lainnya. Namun, tidak semua manusia dilahirkan sempurna, di mana ada beberapa manusia yang tidak beruntung dan kehilangan kemampuannya untuk berkomunikasi secara normal dengan berbicara. Menurut World Federation of the Deaf (WFD), terdapat 70 juta orang tuna rungu yang ada di dunia [1]. Orang – orang pada komunitas tuna rungu ini menggunakan bahasa isyarat atau *sign language* sebagai cara utama untuk berkomunikasi satu sama lainnya dan orang yang bisa mendengar. Bahasa isyarat adalah metode komunikasi visual internasional yang digunakan sebagai alternatif cara komunikasi bagi mereka yang kurang beruntung dalam memiliki indera berbicara atau mendengarnya. Perbedaan ini menimbulkan kesenjangan dalam para kaum tuna wicara dan tuna rungu untuk menjalani kehidupan sehari – harinya, khususnya dalam kehidupan pekerjaan mereka. Dalam keadaan kita yang sekarang ini, terlebih dengan terjadinya pandemic Covid-19, penyebaran lapangan pekerjaan menjadi tidak merata terhadap mereka yang disabilitas untuk mendapatkan pekerjaan yang layak. Untuk mengatasi *social distancing* sekarang hampir seluruh perusahaan di Indonesia menggunakan metode daring atau *online* untuk melaksanakan pekerjaannya bahkan untuk melaksanakan wawancara untuk mencari pekerja yang sesuai dengan kriteria perusahaan. Para pengguna bahasa isyarat tentunya mengalami kesenjangan dan kerugian dengan dilaksanakannya metode ini untuk mendapatkan suatu pekerjaan. Dengan menggunakan sistem pengenalan bahasa isyarat ini, orang – orang yang tidak paham bahasa isyarat bisa dengan lebih mudah memahami apa yang disampaikan oleh mereka yang menggunakan bahasa isyarat. Sehingga dengan itu harapannya adalah mereka yang tidak bisa menggunakan indra berbicaranya dengan baik dan benar dapat memiliki kesempatan yang lebih untuk dapat bersaing dengan para pekerja lainnya dan bisa melaksanakan wawancara dengan perusahaan harapan dengan memanfaatkan sistem pengenalan Bahasa Isyarat yang telah dirancang.

*Sign Language Recognition* adalah suatu program yang dibuat dengan tujuan menerjemahkan bahasa isyarat yang ditunjukkan ke kamera dalam waktu nyata menjadi teks agar dapat lebih mudah dimengerti dan dipahami, sehingga tidak perlu mengingat seluruh arti dari bahasa isyarat tetapi hanya tinggal membaca saja. *Sign Language Recognition* ini digunakan untuk lebih menghubungkan antara mereka yang memiliki disabilitas pendengaran atau berbicara kepada masyarakat lainnya yang ingin saling berkomunikasi tetapi tidak memahami bahasa isyarat sepenuhnya. Penelitian ini dilakukan dengan harapan untuk mempermudah proses pencarian kerja bagi para kaum tuna rungu dan tuna wicara yang tentunya mengalami kesulitan setelah masa pandemi Covid – 19 untuk bersaing dengan kaum normal dengan memanfaatkan *image processing* dan *object detection*. Untuk itu, penelitian ini akan berfokus pada dua algoritma yang paling cocok untuk *object detection* secara *real-time* yaitu algoritma SSD dan YOLO.

Pada penelitian ini perancangan sistem pengenalan Bahasa Isyarat akan dilakukan dengan menggunakan dua algoritma atau model yang berbeda untuk dijadikan perbandingan. Kedua algoritma yang akan digunakan adalah algoritma SSD MobileNet FPNLite 320 x 320 yang merupakan salah satu model dari TensorFlow 2 Detection Model Zoo dan algoritma YOLOv5 dimana kedua algoritma tersebut adalah algoritma yang dapat diimplementasikan untuk deteksi objek secara *real-time* dari suatu citra ataupun video. YOLO dan SSD memiliki pengaplikasiannya masing – masing dimana SSD memberikan tingkat presisi yang lebih tinggi dibandingkan YOLO sehingga YOLO lebih cocok digunakan untuk deteksi objek dimana kita bisa mengabaikan adanya sedikit ketidakakurasian dari sistem, ,sedangkan SSD akan lebih menguntungkan untuk deteksi objek yang membutuhkan tingkat akurasi dan presisi yang tepat dan benar. Maka dari itu penelitian ini akan menentukan apakah versi YOLO yang terbaru yaitu YOLOv5 bisa dibandingkan dengan model SSD terbaru namun dengan nilai COCO mAP dan kecepatan yang relatif kecil dibandingkan dengan model SSD lainnya yang tersedia pada TensorFlow 2 Detection Model Zoo.



## **1.2. Tujuan Penelitian**

Tujuan dari penelitian ini adalah sebagai berikut:

1. Mengimplementasikan algoritma SSD dalam mendeteksi bahasa isyarat dalam waktu nyata
2. Mengimplementasikan algoritma YOLOv5 dalam mendeteksi bahasa isyarat dalam waktu nyata
3. Membandingkan kinerja model SSD dan YOLOv5 dalam deteksi bahasa isyarat dalam waktu nyata
4. Mengevaluasi model terbaik diantara SSD dan YOLOv5 untuk deteksi bahasa isyarat sehingga penelitian untuk topik yang serupa dapat berjalan dengan lebih efektif dan efisien.

## **1.3. Manfaat Penelitian**

Manfaat dari penelitian ini adalah sebagai berikut:

1. Meningkatkan keefektifan berkomunikasi antara pengguna bahasa isyarat dengan mereka yang tidak bisa memahami bahasa isyarat dengan mudah
2. Menerjemahkan bahasa isyarat menjadi suatu teks untuk mempermudah pemahaman bahasa isyarat
3. Meningkatkan proses pembelajaran bahasa isyarat baik secara pribadi atau dari suatu instansi.
4. Mengurangi kesenjangan antara kaum tunarungu dengan mereka yang tidak memiliki disabilitas dalam pemilihan pada lapangan pekerjaan
5. Meningkatkan kesempatan bagi kaum tunarungu untuk mendapatkan pekerjaan yang baik dan layak dengan sistem daring atau *online* yang sudah diimplementasikan di Indonesia.

## **1.4. Batasan Masalah**

Batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Dataset yang digunakan adalah bahasa isyarat menggunakan Bahasa Isyarat Indonesia saja, tidak mencakup seluruh variasi bahasa pada bahasa isyarat di dunia
2. Penelitian ditargetkan untuk mendeteksi gerakan tangan dari satu orang saja dalam satu waktu pengambilan gambar.
3. Dataset yang digunakan adalah dataset dengan topik pekerjaan.
4. Pengambilan gambar harus dilakukan pada *background* yang polos.

### 1.5. Sistematika Penulisan

Sistematika Penulisan pada penelitian ini dibagi ke dalam 5 bab, yaitu:

#### BAB I PENDAHULUAN

Bab I Pendahuluan berisi bahasan tentang latar belakang penelitian, tujuan dan manfaat dari penelitian, dan batasan masalah yang ingin diatasi dari penelitian.

#### BAB II DASAR TEORI

Bab II Dasar Teori berisikan tentang teori dasar dari seluruh hal yang berhubungan dengan penelitian dan menjadi referensi dari penelitian ini.

#### BAB III PERANCANGAN SISTEM

Bab III Perancangan Sistem membahas sistem yang akan diimplementasikan pada program yang akan dibuat.

#### BAB IV EVALUASI DAN ANALISIS HASIL PENELITIAN

Bab IV Evaluasi dan Analisis Hasil penelitian berisikan Hasil penelitian dan pengujian yang telah dilakukan, dan menganalisa hasil tersebut apakah sesuai dengan tujuan dari penelitian

#### BAB V KESIMPULAN DAN SARAN

Bab V Kesimpulan dan Saran menjelaskan kesimpulan dari hasil penelitian yang telah dilakukan dan saran yang dapat diberikan untuk meningkatkan penelitian selanjutnya setelah penelitian selesai.



## **BAB 2**

### **LANDASAN TEORI**

#### **2.1. Bahasa Isyarat**

Bahasa isyarat adalah bahasa berbasis visual yang menggunakan berbagai kombinasi dari gerakan tangan, gestur tubuh, gerakan bibir dan ekspresi wajah untuk menyampaikan suatu isi pikiran dari seseorang. Bahasa isyarat bisa ditunjukkan oleh seseorang melalui gelengan kepala, raut wajah, atau tunjukkan saja. Bahasa isyarat berisikan kombinasi dari sinyal manual yang didukung oleh ekspresi wajah atau mungkin gerakan bibir yang menggambarkan suatu kata secara alfabet manual kapanpun komunikasi secara vokal tidak bisa dilakukan.

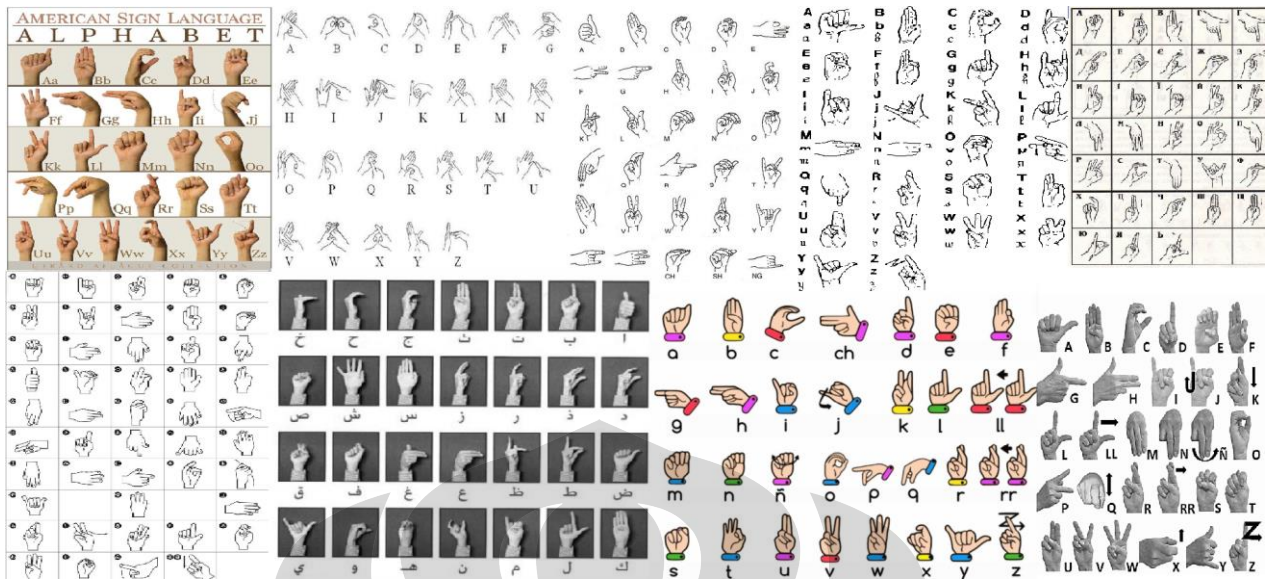
Di Cina dan Jepang di mana bahasa yang digunakan memiliki karakter yang sama namun dibaca berbeda, bisa berkomunikasi dengan menggunakan bahasa isyarat dengan cara memperhatikan pergerakan tangannya. Sebelum ditemukannya penggunaan bahasa modern, ditemukan bukti penggunaan bahasa isyarat pada benua seperti Afrika, Australia dan Amerika Utara [2]. Ini menandakan bahwa pada masa sebelum manusia mengandalkan kepintaran untuk berkomunikasi secara normal menggunakan suara dan bahasa, manusia purba hanya bisa menyampaikan isi pikirannya dengan gestur tubuh sederhana dan menunjuk sesuatu yang dia inginkan. Hal ini telah diimplementasikan dalam hidup manusia oleh para leluhur kita di mana kemudian menjadi dasar dalam pengembangan bahasa manusia seperti bentuk dari gestur tersebut dijadikan suatu simbol yang nantinya dijadikan suatu huruf kemudian berubah menjadi suatu kumpulan bahasa yang unik bagi setiap negara di dunia.

Walau sekarang sudah banyak digunakan bahasa manusia yang lebih modern yaitu dengan menggunakan suara, namun bahasa isyarat juga masih memiliki perannya di kehidupan manusia pada zaman sekarang. Secara tidak sadar manusia juga menggunakan tubuhnya untuk membantunya dalam menyampaikan pikirannya sambil berbicara menggunakan mulutnya. Hal sederhana seperti ini juga sudah termasuk penggunaan bahasa isyarat, di mana bahasa isyarat sangat membantu menyampaikan isi pikiran seseorang menggunakan gestur tubuhnya.

Bahasa isyarat juga digunakan untuk komunikasi jarak jauh saat suara tidak bisa terdengar, atau bisa juga digunakan saat target lawan bicara kita tidak menggunakan bahasa yang sama dengan kita. Bahasa isyarat sudah menjadi bagian dari kehidupan kita sehari – hari yang tidak akan bisa kita lepaskan dari bagian hidup manusia.

Penggunaan bahasa isyarat menjadi lebih berguna lagi di dunia medis di mana ada sebagian manusia yang mengalami ketidakmampuan untuk berbicara atau mendengar sehingga tidak bisa berkomunikasi menggunakan bahasa yang sewajarnya. Orang – orang dengan disabilitas ini atau tuna rungu dan tuna wicara, mengandalkan bahasa isyarat dalam berkomunikasi dengan sesamanya atau dengan orang normal. Dengan adanya bahasa isyarat, manusia dengan ketidakmampuan untuk mendengar atau berbicara bisa mendapatkan kesempatan untuk belajar seperti layaknya manusia normal. Karena kekurangannya, mereka yang tuna wicara dan tuna rungu tidak bisa mendapatkan kesempatan belajar dikarenakan tidak banyak guru yang memiliki pengalaman dan pengetahuan untuk mengajar mereka. Namun dengan adanya bahasa isyarat, kesempatan belajar untuk tuna rungu dan tunwaicara pun semakin meningkat.

Bahasa Isyarat digunakan pada seluruh negara dengan variasinya masing – masing. Sama halnya dengan bahasa Indonesia, bahasa Inggris, bahasa Cina, dan bahasa India yang berbeda satu dengan yang lainnya, pada bahasa isyarat juga disesuaikan dengan negara asal dari bahasa tersebut. Berikut adalah bahasa isyarat yang digunakan berbagai macam negara di seluruh dunia.



Gambar 2.1 Bahasa Isyarat Dari Seluruh Dunia [3]

Gambar 2.1 adalah berbagai macam bahasa isyarat yang digunakan diseluruh dunia, dari atas kiri adalah *American Sign Language (ASL)*, *British, Australian, and New Zealand Sign Language (BANZSL)*, *Chinese Sign Language (CSL)*, *French Sign Language (LSF)*, *Japanese Sign Language (JSL) Syllabary*, *Arabic Sign Language*, *Spanish Sign Language (LSE)*, *Mexican Sign Language (LSM)*, dan *Ukrainian Sign Language (USL)*.

### 2.1.1. Bahasa Isyarat di Indonesia

Pada Indonesia, para kaum tuna rungu dan tuna wicara berkomunikasi menggunakan bahasa isyarat yang mengacu kepada dua sistem yang ada yaitu Bahasa Isyarat Indonesia (BISINDO) dan Sistem Isyarat Bahasa Indonesia (SIBI) [4]. BISINDO adalah bahasa isyarat yang dikembangkan oleh kaum tuna rungu sendiri melalui Gerakan Kesejahteraan Tuna Rungu Indonesia (GERKATIN). SIBI dikembangkan oleh orang normal yang bukan merupakan penderita tuna rungu di mana SIBI menggunakan bahasa isyarat yang sama dengan yang digunakan di Amerika yaitu *ASL*. Berikut adalah perbedaan abjad yang digunakan pada BISINDO dan SIBI.



Gambar 2.2 Abjad BISINDO (Kiri) dan SIBI (Kanan) [5]

BISINDO yang merupakan buatan dari kaum tuna rungu sendiri memiliki abjad yang dibuat berdasarkan pengamatan mereka sendiri. Sedangkan SIBI yang dibentuk oleh kepala SLB (Sekolah Luar Biasa) sudah diresimkan oleh pemerintah dan digunakan pada pembelajaran di SLB. BISINDO disampaikan dengan menggunakan dua tangan sedangkan SIBI disampaikan dengan hanya menggunakan satu tangan, tetapi dianggap lebih rumit diandingkan BISINDO karena mengandung kosakata yang baku dan rumit [6].



Gambar 2.3 Kata “Saya” (kiri) dan “Anda” (kanan) dalam SIBI [7]

## 2.2. Machine Learning

*Machine Learning* adalah salah satu bentuk *artificial intelligence (AI)* yang memungkinkan suatu sistem komputer untuk beradaptasi terhadap suatu keadaan dan memiliki kemampuan belajar seperti layaknya seorang manusia. *Machine learning* adalah ilmu yang mempelajari cara membuat algoritma yang bisa terus belajar dan bisa berlatih untuk meningkatkan akurasi dari waktu ke waktu tanpa pemrograman tertentu sehingga sistem atau program yang sudah dibuat tersebut bisa membuat suatu keputusan atau memprediksi suatu hasil berdasarkan input yang diberikan sesuai dengan hasil dari latihan tersebut. Semakin bagus algoritma yang digunakan maka akurasi dari keputusan dan prediksi yang dihasilkan juga akan semakin meningkat [8]. Cara kerja *machine learning* dapat difokuskan kedalam tiga poin utama yaitu proses pengambilan keputusan, fungsi eror, dan proses optimasi model.

Pengambilan keputusan pada *machine learning* berdasarkan prediksi dan klasifikasi dari input data yang bisa diberi nama atau tidak diberi nama dan algoritma dari mesin nantinya akan mengestimasi dan menganalisa pola yang ada pada data, dari pola tersebut kemudian bisa dilakukan prediksi untuk data input selanjutnya. Fungsi eror pada *machine learning* adalah fungsi yang digunakan untuk evaluasi prediksi dari suatu model, kesalahan prediksi dari algoritma yang dijalankan oleh mesin ditunjukkan oleh seberapa besar nilai akurasi. Lalu proses optimasi model adalah saat algoritma dari mesin menentukan sendiri melalui pengulangan dan optimasi berdasarkan latihan yang dilakukan dari data sedemikian rupa sehingga terbentuk suatu model yang bisa digunakan sebagai pacuan yang senantiasa berkembang secara otomatis yang kemudian menjadi titik berat peningkatan akurasi algoritma [9].

## 2.3. Deep Learning

*Deep Learning* adalah suatu metode machine learning yang menggunakan algoritma dengan pola pikir seperti manusia yaitu *Artificial Neural Networks (ANN)* yang merupakan jaringan saraf dengan tiga atau lebih lapisan ANN. Dengan jaringan saraf buatan ini, algoritma mesin dirancang untuk dapat beradaptasi



terhadap input data dalam jumlah yang sangat besar dan cocok untuk menyelesaikan permasalahan yang rumit bagi algoritma *machine learning*.

*Deep learning* lebih banyak digunakan karena dengan menggunakan *deep learning* mesin dapat memproses data seperti teks dan gambar dalam jumlah yang banyak sekaligus. Hasil yang diberikan oleh algoritma tersebut akan lebih berkualitas dibandingkan dilakukan secara manual atau dengan menggunakan algoritma *machine learning* yang lainnya dengan biaya operasional yang lebih sedikit. Berdasarkan dari struktur jaringan, ada dua jenis arsitektur yang digunakan dalam *deep learning* yaitu RNN (*Recurrent Neural Network*) dan CNN (*convolutional neural network*). Kedua arsitektur tersebut digunakan untuk keadaan dan tugas yang berbeda di mana pelatihan data sangat berpengaruh terhadap hasil akhir performa dari algoritma tersebut.

#### 2.4. OpenCV

OpenCV adalah suatu *open-source library* yang digunakan pada *machine learning* dan *image processing* yang bisa digunakan untuk mengidentifikasi suatu objek, muka, bahkan tulisan tangan dari manusia melalui gambar atau video dalam waktu nyata [10]. Dengan menggunakan OpenCV algoritma mesin dapat mengambil data menggunakan kamera secara langsung dan bisa mendeteksi objek yang kemudian bisa digunakan dalam penerapan algoritma *machine learning* dengan menggunakan *labeling* terhadap objek pada gambar tersebut yang kemudian bisa diproses dan dilatih sesuai algoritma mesin.

*Library* yang digunakan memiliki lebih dari 2500 algoritma yang memiliki algoritma *machine learning*. Algoritma tersebut bisa digunakan untuk mendeteksi wajah, identifikasi objek, klasifikasi tindakan manusia dari video, dan mengikuti objek yang bergerak. *Library* dari OpenCV sudah digunakan dan diimplementasikan sepenuhnya oleh perusahaan ternama seperti Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, dan Toyota. [11]

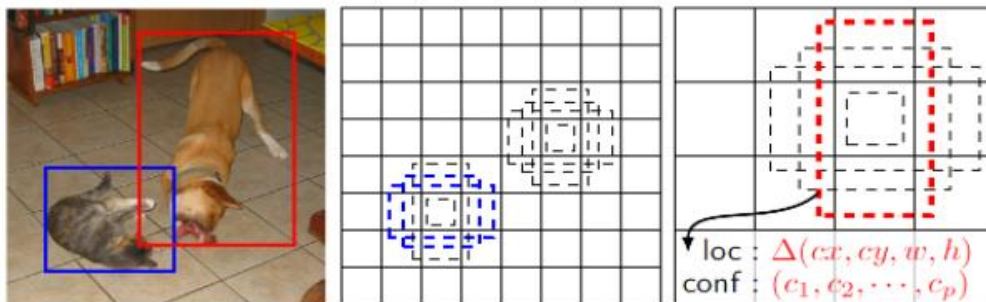
## 2.5. Tensorflow

Tensorflow adalah suatu *library* untuk *machine learning* yang dikembangkan oleh Google yang digunakan untuk mengembangkan algoritma *machine learning*. Framework Tensorflow dibuat menggunakan Python front-end API dan C++ untuk menggabungkan model dan algoritma yang digunakan oleh *machine learning* dan *deep learning*.

Tensorflow menggunakan grafik dataflow untuk mewakili seluruh komputasi dan keadaan algoritma *machine learning* dan juga operasi perhitungan matematika, parameternya, dan input *pre-processing*. Grafik dataflow yang dihasilkan menunjukkan komunikasi antara komputasi secara eksplisit sehingga membuat komputasi mandiri yang dihasilkan oleh komputer untuk dapat dijalankan secara paralel dan untuk partisi proses komputasi antar perangkat.

## 2.6. SSD

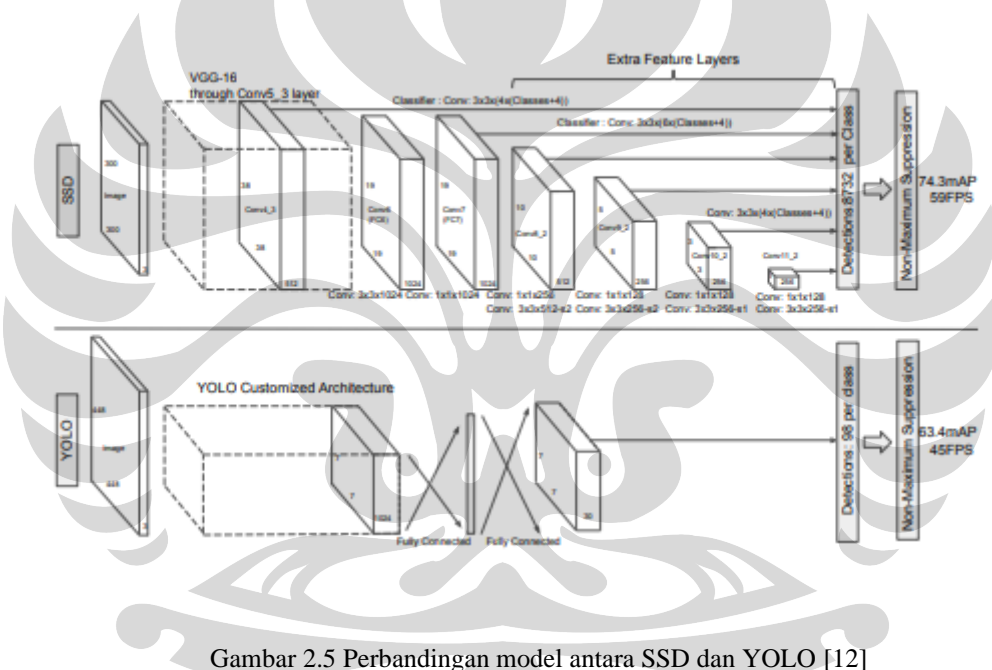
*Single-Shot Detector* (SSD) adalah algoritma yang digunakan untuk mendeteksi suatu objek pada suatu gambar yang memiliki akurasi dan kecepatan yang melebihi dari algoritma *You Only Look Once* (YOLO). YOLO adalah algoritma yang mendeteksi dan mengenali suatu objek dari suatu gambar dalam waktu nyata di mana YOLO mendeteksi objek secara *real-time* menggunakan CNN. SSD adalah algoritma yang telah melampaui kecepatan dan akurasi dari YOLO, di mana SSD300 telah mencapai 74.3% mAP dengan 59 FPS sedangkan SSD500 telah mencapai 76.9% mAP pada 22 FPS dibandingkan dengan YOLOv1 yang memiliki 63.4% mAP pada 45 FPS.



Gambar 2.4 MultiBox Detector [12]

Gambar 2.4 menunjukkan cara kerja SSD menggunakan *MultiBox Detector* untuk mendeteksi objek pada suatu gambar. Berdasarkan input gambar dan label yang diberikan saat *training*, SSD akan mendeteksi objek tersebut pada gambar yang ada dengan memberikan kotak pada objek tersebut.

Model yang digunakan oleh SSD merupakan peningkatan dari model pada YOLO di mana SSD menggunakan *multi-scale feature maps* untuk deteksi gambar dengan menambahkan *convolutional* layer diakhir network layer pada arsitektur standar yang digunakan pada klasifikasi gambar dengan kualitas yang tinggi. SSD juga menggunakan *convolutional predictor* untuk deteksi pada setiap layer.



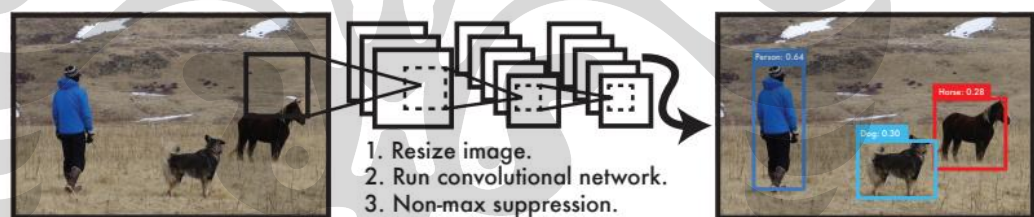
Gambar 2.5 Perbandingan model antara SSD dan YOLO [12]

Pada model SSD terdapat bermacam – macam lapisan yang memiliki perannya masing - masing di mana pada lapisan tertentu terdapat lilitan yang digunakan untuk deteksi objek seperti pada gambar 2.5. misalnya pada lilitan 4\_3 di mana memiliki ukuran  $38 \times 38 \times 512$ . Maka akan terdapat 4 *bounding boxes* di mana setiap *bounding box* akan memiliki output sebesar jumlah kelas + 4, maka outputnya adalah  $38 \times 38 \times 4 \times (21 + 4) = 144,400$  jumlah output dan jumlah *bounding boxes* sebesar  $38 \times 38 \times 4 = 5776$  *bounding boxes*. Perhitungan ini dilakukan terhadap seluruh lapisan dan lilitan dari setiap lapisan tersebut yang memiliki jumlah total sebesar 8732 *bounding boxes*. *Bounding boxes* inilah yang

menjadi perbandingan dengan YOLO di mana YOLO hanya memiliki sebanyak 98 *boxes* saja [12]. Pada SSD juga digunakan perincian perhitungan data saat *training* di mana dengan memperhitungkan jumlah *box* dan menghitung lossnya maka akan bisa ditemukan hasil yang lebih efisien dan hasil pelatihan yang lebih stabil.

## 2.7. You Only Look Once

You Only Look Once (YOLOv5) adalah suatu algoritma yang dirancang dengan tujuan untuk mendeteksi dan mengenali objek secara *real-time* atau dalam waktu nyata dengan mengkategorikan objek yang telah dideteksi oleh kamera dan memisahkan antara objek tersebut dengan menggunakan *frame* yang disebut sebagai suatu *bounding box*. Suatu *neural network* tunggal memprediksi terdapatnya probabilitas *bounding box* dan *class secara langsung* dari gambar secara utuh dalam satu proses evaluasi gambar tersebut. Karena seluruh proses deteksi merupakan suatu bagian dari satu jaringan, maka dapat dioptimasi secara langsung dengan menggunakan *end to end* pada *detection performance* [13].



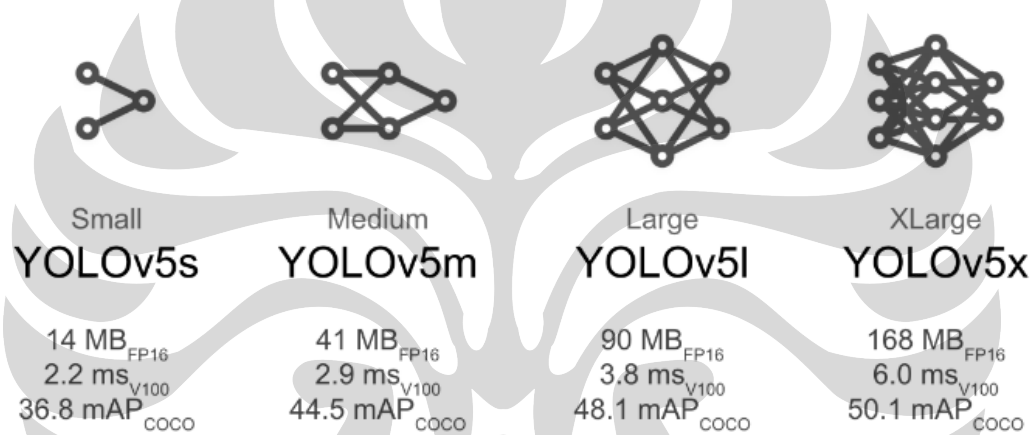
Gambar 2.6 YOLO Detection System [13]

Cara kerja YOLO bisa dilihat seperti pada Gambar 2.6. Suatu *convolutional network* bekerja untuk mendeteksi beberapa *bounding box* secara bersamaan dan probabilitas dari *class* untuk setiap kotak yang ada pada gambar. Sistem memulai dengan menyesuaikan ukuran citra dengan model dengan menggunakan metode *resize* yang kemudian dengan menggunakan citra baru tersebut akan dilakukan proses *convolutional network* dan hasilnya akan dideteksi sesuai dengan *confidence* dari model.

Cara kerja YOLO sangatlah cepat dimana deteksi suatu *frame* tidak membutuhkan *pipeline* yang kompleks. YOLO memiliki kemampuan untuk

mendeteksi suatu video secara *real-time* dengan latensitas kurang dari 25 ms dan memiliki hasil *mean average precision* lebih dari dua kali sistem lainnya dalam mendeteksi objek.

Algoritma yang dibuat pertama kali oleh Joseph Redmon pada tahun 2015 ini, dikembangkan oleh banyak peneliti hingga sekarang dan pada 18 Mei 2020, Glenn Jocher memperkenalkan YOLOv5 yang mengimplementasikan framework Pytorch [14]. Dalam YOLOv5 terdapat beberapa pilihan model yang dapat digunakan dan disesuaikan dengan kebutuhan. Semakin baik model yang digunakan maka akan semakin lama pula waktu yang dibutuhkan untuk model bekerja. Berikut adalah beberapa pilihan model yang tersedia pada YOLOv5.



Gambar 2.7 Seleksi Model YOLOv5 [15]

Gambar 2.7 menunjukkan seleksi model YOLOv5 yang dapat digunakan untuk kebutuhan object detection. Model YOLOv5s dan YOLOv5m digunakan untuk *mobile deployment* sedangkan YOLOv5l dan YOLOv5x biasa digunakan untuk *cloud deployment*.

## 2.8. Precision, Recall, dan F1 Score

*Precision* dan *recall* merupakan parameter yang digunakan untuk mengukur ketepatan dan kebenaran dari suatu sistem dalam mendeteksi suatu objek dengan benar. Dalam proses perhitungannya *precision*, *recall*, dan *F1 Score* menggunakan parameter TP, TN, FP, dan FN yaitu :

- TP (*True Positive*) adalah saat terdapat objek pada citra dan sistem berhasil mendeteksi objek dengan benar dan memiliki output sesuai dengan nama objek pada citra.
- TN (*True Negative*) adalah saat benar tidak terdapat objek pada citra dan sistem tidak mendeteksi adanya objek pada citra.
- FP (*False Positive*) adalah saat sistem mendeteksi citra dan memiliki output sesuai dengan nama objek pada citra namun objek tidak dideteksi dengan tepat
- FN (*False Negative*) adalah saat seharusnya ada objek yang dideteksi pada citra, namun sistem tidak berhasil mendeteksi objek tersebut sama sekali

Dengan menggunakan parameter diatas, *precision* digunakan untuk mengukur seberapa akurat suatu model dengan membandingkan berapa banyak hasil yang diprediksi sebagai positif dan berapa diantara mereka yang memang benar positif. Rumus perhitungan yang digunakan untuk menghitung *precision* adalah sebagai berikut :

$$Precision = \frac{TP}{TP + FP} \quad (2.1)$$

Sedangkan *Recall* digunakan untuk mengukur berapa banyak TP yang ditangkap oleh model dengan dibandingkan juga terhadap berapa banyaknya FN. Dengan menghitung *Recall* kita bisa memastikan tidak ada kasus dimana adanya objek yang seharusnya positif tetapi diprediksi sebagai negatif oleh model. Rumus perhitungan untuk menghitung *recall* adalah

$$Recall = \frac{TP}{TP + FN} \quad (2.2)$$

*F1 Score* adalah nilai yang digunakan sebagai acuan untuk mendapatkan nilai banding antara *precision* dan *recall* dengan menggunakan rumus berikut :

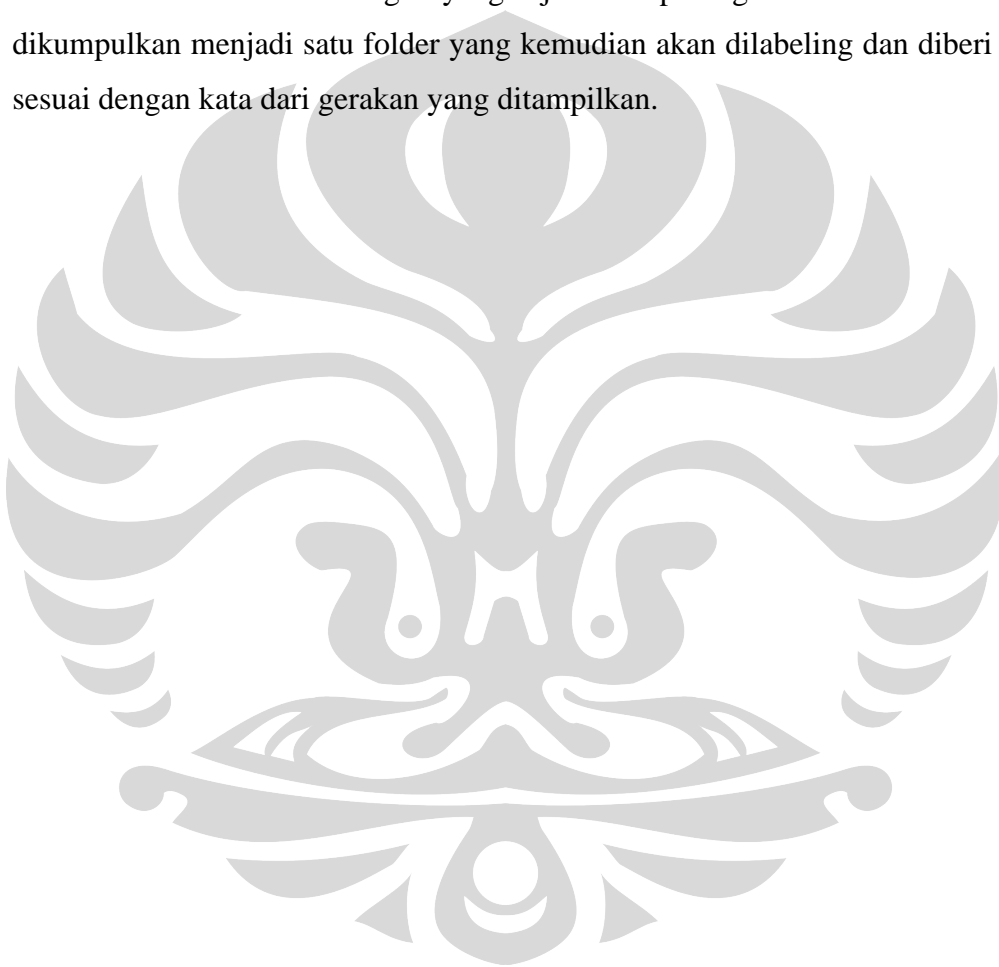
$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (2.3)$$

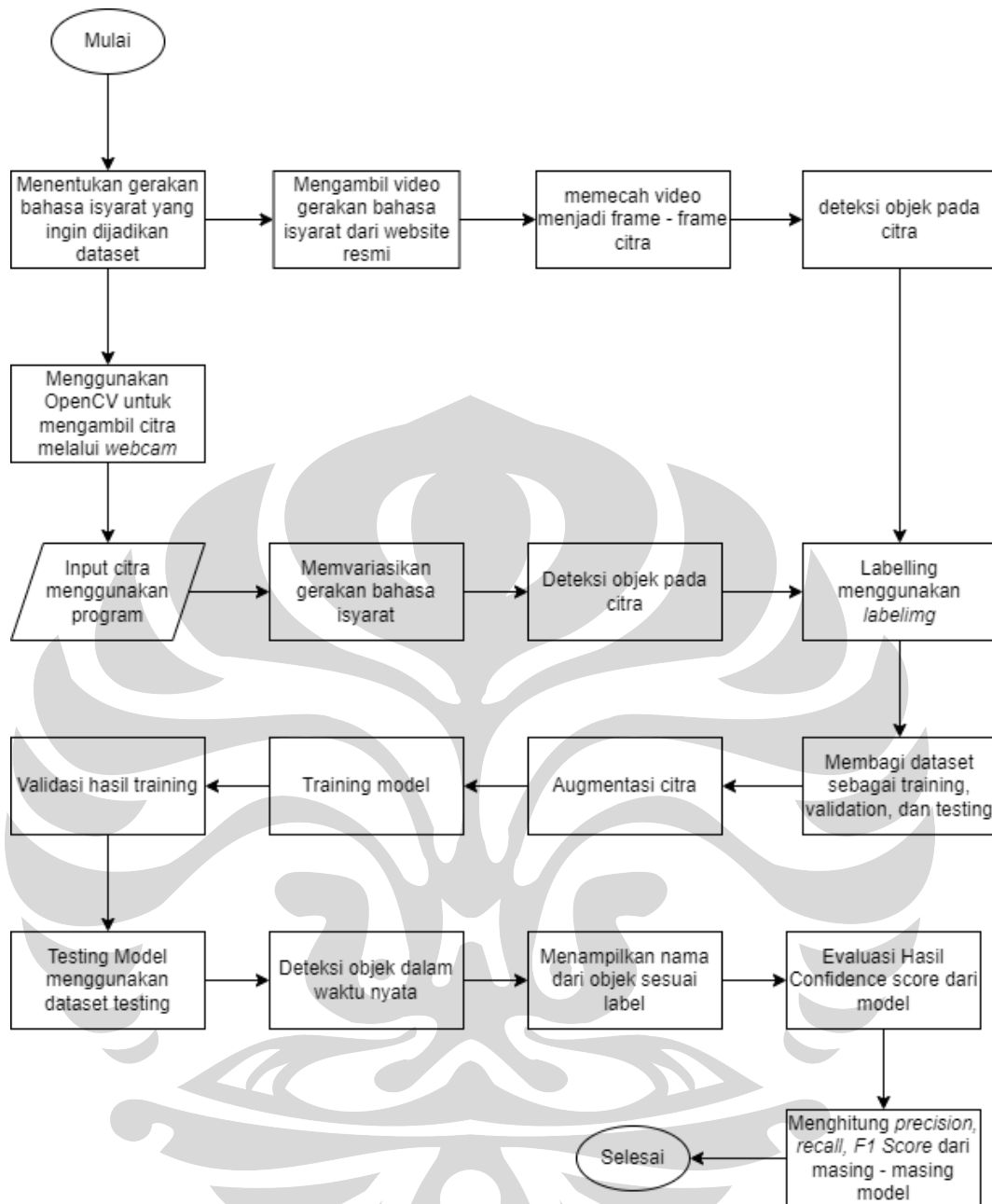
## **BAB 3**

### **PERANCANGAN SISTEM**

#### **3.1. Diagram Alur**

Dalam melaksanakan penelitian, proses berjalannya penelitian ini digambarkan dengan diagram alur. Diagram alur dari penelitian yang akan dilakukan adalah sesuai dengan yang dijelaskan pada gambar 3.1. Dataset akan dikumpulkan menjadi satu folder yang kemudian akan dilabeling dan diberi nama sesuai dengan kata dari gerakan yang ditampilkan.





Gambar 3.1 Diagram Alur Penelitian

Gambar 3.1 menjelaskan alur pembuatan sistem di mana dimulai dari mengambil gambar menggunakan OpenCV, sesuai dengan angka yang sudah ditentukan diawal kamera akan mengambil gambar sejumlah angka tersebut untuk setiap jenis kata. Variasi dari suatu objek akan dilakukan saat pengambilan gambar dan kemudian akan dilakukan augmentasi terhadap seluruh dataset yang telah dikumpulkan, dan kemudian seluruh gambar tersebut akan dilabel dan diberi nama.



Setelah dataset terkumpul, maka *training* akan dimulai dan hasil akan dievaluasi. Setelah *training* selesai program akan dijalankan dan kamera akan menyala dan mendeteksi gambar dalam waktu nyata. Objek yang terdeteksi oleh kamera dan dianggap sesuai dengan label yang telah dibuat akan ditandai dan ditunjukkan namanya sesuai label yang telah dibuat lalu evaluasi hasil akhir.

### 3.2. Dataset

#### 3.2.1. Pengumpulan Dataset

Dataset yang digunakan dalam pembuatan sistem pengenalan bahasa isyarat adalah berbagai macam kata dalam Bahasa Isyarat Bahasa Indonesia yang biasa digunakan dalam suatu topik pembicaraan saat wawancara untuk mendaftar dalam suatu pekerjaan, dengan masing – masing kata jenisnya divariasikan sehingga sistem bisa mengenali suatu kata dalam keadaan yang berbeda. Data diambil dari website resmi yang digunakan sebagai kamus berbahasa isyarat Indonesia [7] dan dibantu oleh gambar yang diinput sendiri oleh penulis. Untuk setiap kata akan dikumpulkan citra sebanyak 15 - 20 citra untuk *training*, 1 citra untuk *validasi* dan 3 citra untuk *testing*. Kemudian setiap citra yang telah dikumpulkan tersebut akan dilakukan augmentasi data sebanyak 3 kali sehingga menjadi total 45 - 60 citra untuk 1 jenis kata dan untuk keseluruhan karena digunakan ada 27 jenis kata yang berbeda maka jumlah total citra yang digunakan untuk *training* adalah 1389 citra untuk *training*, 54 citra untuk *validasi*, dan 81 citra untuk *testing*. Dalam paper A. Ghafar, J. Sari menggunakan 5 citra yang berbeda-beda tiap model Bahasa isyarat dengan posisi jarak yang telah ditentukan dimana citra yang digunakan memiliki jumlah sebanyak 130 citra yang dibagi menjadi citra

*training* dan testing, dimana 104 citra digunakan untuk citra *training* dan 26 citra digunakan untuk citra testing untuk melakukan pencocokan. [16]



Gambar 3.2 Dataset Topik Pekerjaan "Ahli", "Apa", "Daftar" (kiri ke kanan)[7]



Gambar 3.3 Dataset Topik Pekerjaan "Tidak", "Ya", "Saya", "Rumah", "Kerja" (kiri ke kanan)[7]

Gambar 3.2 dan 3.3 menunjukkan contoh bseberapa dataset yang digunakan dalam penelitian ini. Dataset yang digunakan adalah kata yang berhubungan dengan topik pekerjaan dimana kata – kata yang sering diucapkan dalam suatu sesi wawancara pekerjaan. Model akan mendeteksi objek yaitu tangan dari orang yang melakukan gestur gerakan bahasa isyarat tersebut dan menunjukkan apa arti dari gerakan yang dilakukan tersebut. Sumber dari dataset yang diperoleh diambil dari dua sumber berbeda yaitu :

1. Memecah Gerakan dari video yang diambil dari website resmi Bahasa Isyarat Indonesia "Kamus SIBI" [7] dengan menggunakan program Python dari satu Gerakan per kata yang digunakan menjadi frame per 15 milidetik dilihat dari Gerakan yang dihasilkan antar jarak waktu tersebut. Pada website tersebut terdapat fitur pencarian untuk mendapatkan kata

yang ingin dicari, namun karena tidak bisa dilihat ada berapa banyak kata dalam website tersebut, peneliti hanya menggunakan kata yang berhubungan dengan topik yang dipilih yaitu pekerjaan. Jumlah kata yang digunakan untuk penelitian ini dengan topik pekerjaan adalah 27 kata berbeda.

2. Dataset berasal dari citra yang diambil sendiri oleh peneliti menggunakan program pengambil gambar otomatis yang dibuat oleh peneliti dengan menggunakan looping dan webcam menggunakan OpenCV yaitu sebanyak 15 citra perkata yang kemudian hanya digunakan citra yang membentuk Gerakan sesuai dengan Gerakan pada Kamus SIBI [7].



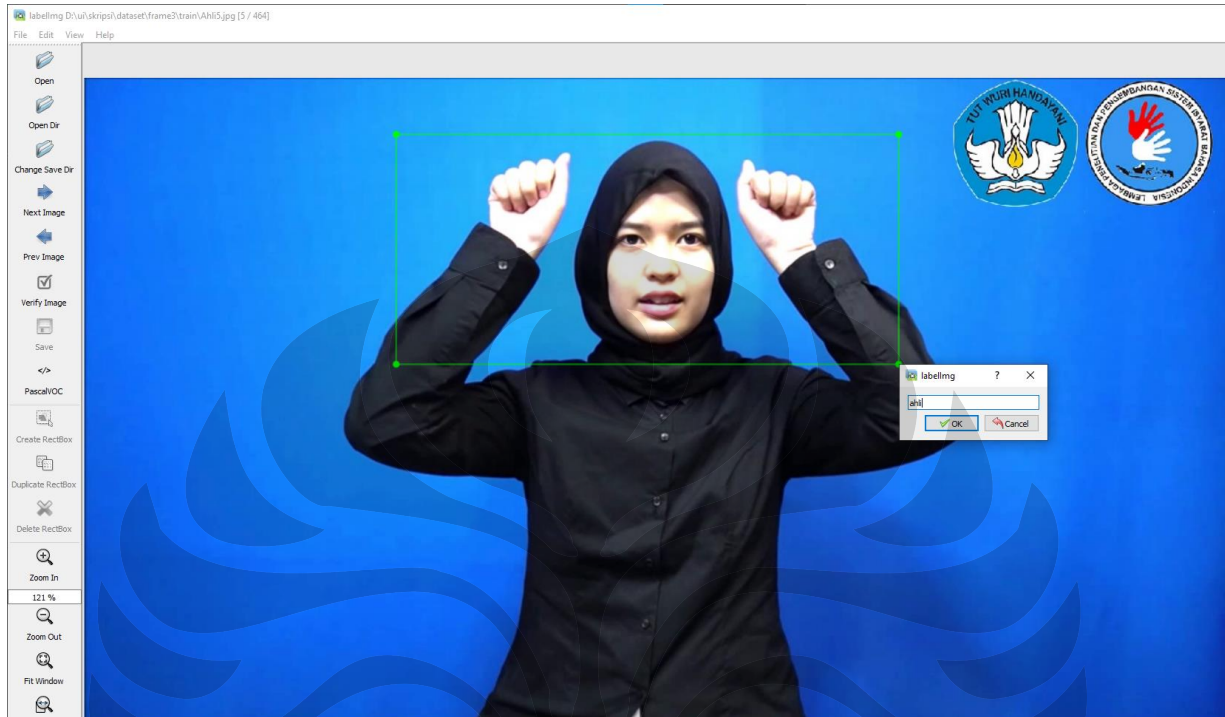
Gambar 3.4 Dataset Topik Pekerjaan "Ahli", "Apa", "Daftar" (kiri ke kanan)

Dataset yang terkumpul memiliki resolusi yang berbeda dimana dataset yang diambil dari website resmi tersebut memiliki resolusi 1280 x 720 pixel dan 1920 x 1080 pixel, dan dataset yang dibuat sendiri oleh peneliti memiliki resolusi sebesar 640 x 480 pixel. Untuk mengatasi hal ini, maka pada program setiap model yang digunakan maka citra akan diresize menjadi ukuran 320 x 320 pixel menyesuaikan dengan model SSD MobileNet FPNLite 320 x 320 yang hanya mendukung penggunaan gambar dengan resolusi tersebut. Maka dari itu untuk Model Yolov5 akan menggunakan resolusi yang sama yaitu 320 x 320 pixel.

### 3.2.2. *Preprocessing*

Setelah menyiapkan dataset, peneliti melakukan *preprocessing* data untuk mempermudah pemrosesan citra pada tahap selanjutnya dimana seluruh citra yang telah dikumpulkan untuk *training* yaitu sebanyak 15 - 20 citra untuk 27 kata yang berbeda sehingga berjumlah 463 citra dilakukan proses *labelling* kepada seluruh dataset *training* dan *validation* yang sudah terkumpul. Proses *labelling* yang

dilakukan oleh peneliti menggunakan bantuan aplikasi labelimg dari python yaitu dengan memberikan kotak terhadap objek pada gambar untuk setiap individu gambar.



Gambar 3.5 Proses *Labelling* gambar menggunakan Labelimg

Gambar 3.5 menunjukkan proses *labelling* yang dilakukan oleh peneliti untuk salah satu contoh dataset yang digunakan untuk proses *training*. Pada proses *labelling* pertama setelah mengumpulkan seluruh dataset yang ingin digunakan untuk training menjadi 1 folder yang sama, folder tersebut akan diakses menggunakan aplikasi labelimg yang kemudian dilakukan penandaan secara manual dengan mengkotakkan objek yang ingin dideteksi sebagai objek nantinya oleh model, dalam hal ini objek yang ingin dideteksi adalah tangan dari pengguna Bahasa Isyarat. Setelah objek selesai diberikan kotak sesuai dengan seperti di Gambar 3.5, selanjutnya adalah memberikan nama dari objek tersebut dalam hal ini yaitu “ahli” dan kemudian setelah mengklik tombol “ok” maka citra telah berhasil diberikan label yang disimpan sebagai suatu koordinat dari letak kotak penanda tersebut.

Setelah proses *labelling* selesai dilakukan, selanjutnya adalah dilakukan augmentasi terhadap masing – masing dari citra *training* tersebut. Augmentasi yang dilakukan terhadap citra *training* adalah sebagai berikut :

1. *Flip*
2. *Crop*
3. *Rotation*
4. *Shear*
5. *Grayscale*
6. *Hue*
7. *Saturation*
8. *Brightness*
9. *Exposure*
10. *Blur*
11. *Noise*

Augmentasi data dilakukan untuk setiap citra dari 463 citra *training* yang telah terkumpul sehingga menghasilkan 3 citra untuk setiap citra awal, dimana diakhir proses augmentasi telah terkumpul 45 - 60 citra untuk masing – masing kata yang menghasilkan sebanyak 1389 citra untuk *training*. Sedangkan untuk *validasi* dan *testing* tidak dilakukan proses augmentasi data.

### **3.3. Perancangan Model**

Kedua model yang digunakan adalah model SSD MobileNet v2 FPNLite 320 x 320 dan YOLOv5s. Model SSD MobileNet v2 320 x 320 yang digunakan diambil langsung dari website github TensorFlow 2 Detection Model Zoo [17]. Model SSD yang digunakan memiliki kecepatan 22ms dan COCO mAP sebesar 22.2. Model YOLOv5s yang digunakan diambil langsung dari website github YOLOv5 yang menggunakan PyTorch dengan kecepatan 6.4ms, dan COCO mAP sebesar 37.2 [18]

### 3.4. Skenario Pengujian

Penelitian akan dilaksanakan skenario dengan melakukan *Training* berdasarkan gambar yang telah direkam sebelumnya menggunakan OpenCV. Semua gambar yang telah direkam kemudian akan dilabelling diberi nama sesuai dengan nama dari gerakan Bahasa Isyarat tersebut dan kemudian menggunakan dataset yang sudah jadi tersebut dan dibagi menjadi dataset *training*, *validation*, dan *testing*. Setelah dataset sudah selesai dikelompokkan maka akan dilakukan *training* dengan menggunakan 2 model yang berbeda yaitu model SSD MobileNet FPNLite 320 x 320 dan menggunakan model YOLOv5. Setelah *training* dilaksanakan hingga selesai, selanjutnya adalah melakukan testing terhadap gambar atau video dari seseorang yang melakukan Gerakan sesuai dengan dataset yang tersedia.

Dataset yang digunakan adalah citra yang direkam dengan ukuran 1280 x 720 piksel dan 640 x 480 piksel yang kemudian seluruh citra tersebut akan diresize menjadi 320 x 320 piksel, parameter yang digunakan adalah besar akurasi yang dihasilkan oleh mesin setelah *training* dan testing selesai. Setelah mesin selesai *training* dan testing saat sistem dijalankan maka deteksi objek akan mulai mendeteksi bahasa isyarat sesuai dengan kata yang ada di dataset yang telah diinput. Evaluasi kinerja sistem akan dapat dilihat saat objek bahasa isyarat yang ditampilkan di mesin bersesuaian dengan dataset yang telah dimasukkan sebelumnya. Saat objek yang dideteksi oleh sistem telah bersesuaian maka layar akan menampilkan *bounding box* terhadap objek yang dideteksi yaitu tangan sebagai tanda bahwa objek tersebut sudah terdeteksi dan nama dari kata tersebut akan terlihat sebagai bagian dari *bounding box* tersebut. Penelitian ini akan membandingkan keefektifan akurasi dan efisiensi kecepatan objek terdeteksi yang dihasilkan oleh mesin menggunakan model SSD dan YOLO lalu menentukan model mana yang lebih cocok untuk melaksanakan tugas dalam mendeteksi Gerakan pada Bahasa Isyarat. Perbandingan kedua model tersebut akan dilakukan dengan menggunakan perangkat keras sebagai berikut

1. CPU : Intel Core i7 Gen 10th
2. RAM : 16 GB
3. *Storage* : 512 GB SSD & 256 GB SSD

4. OS : Windows 10 Pro 64-bit
5. GPU : NVIDIA GTX 1660 SUPER
6. Kamera : NYK Nemesis A50 (Crusher) Full HD 1080

### 3.5. Parameter Evaluasi

Perbandingan dari kedua model yang digunakan antara model SSD MobileNet FPNLite 320 x 320 dan YOLOv5 adalah penelitian akan membandingkan besar akurasi, *precision*, *recall*, *F1 Score* dari kedua model tersebut setelah *training* dan *testing*. Penelitian ini juga akan membandingkan berapa kali kata dapat diprediksi dengan benar dari 27 kali percobaan sesuai dengan jumlah jenis kata yang digunakan untuk *training*. Evaluasi akan dilakukan dengan menggunakan parameter yang sama untuk kedua model. Dimana penelitian ini akan mengevaluasi hasil *training* dari kedua model yang digunakan, dan membandingkan kecepatan yang diperlukan untuk memproses data, Jumlah Loss saat *training* dengan menggunakan epoch dan batch size yang sama, dan besar akurasi dari setiap kata saat *testing* setelah *training* dilakukan hingga selesai, dan sebagai pembandingan penelitian ini juga akan mencatat berapa kali setiap model gagal mendeteksi suatu kata yang sama dan dibuat menjadi suatu perbandingan untuk kedua model tersebut.

#### 3.5.1. Evaluasi Akurasi, *Precision*, *Recall*, dan *F1 Score* Model

Penelitian ini akan menggunakan parameter *Precision*, *Recall*, dan *F1* sebagai pembandingan dari hasil penelitian antara model SSD MobileNet v2 FPNLite 320 x 320 dan YOLOv5s, dimana proses perhitungan dari *Precision*, *Recall*, dan *F1* kedua model akan dilakukan dengan menggunakan hasil deteksi dari 3 dataset *testing* yang tersedia dengan menentukan apakah deteksi objek pada citra tergolongkan TP, TN, FP, dan FN.

Dengan menggunakan berapa jumlah TP, FP, dan FN penelitian ini akan menghitung besar *Precision*, *Recall*, dan *F1* dari model untuk dijadikan

perbandingan seberapa tepat kedua model dalam mendeteksi objek setelah training selesai dilakukan.





## BAB 4

### EVALUASI DAN ANALISIS HASIL PENELITIAN

#### 4.1. Implementasi Sistem

Sistem pengenalan Bahasa Isyarat menggunakan model SSD MobileNet v2 FPNLite 320 x 320 dan YOLOv5 dilakukan dengan menggunakan versi *software* dan *library* sebagai berikut:

1. Python 3.8.5 sebagai library utama program yang digunakan untuk penelitian
2. IPython 8.3.0 digunakan sebagai library untuk Jupyter notebook
3. Tensorflow 2.9.0 Library yang digunakan untuk komputasi aljabar dan segala proses perhitungan pada program
4. Keras 2.9.0 digunakan oleh algoritma SSD
5. OpenCV 4.5.5.64 digunakan untuk mengambil gambar yang digunakan untuk penelitian
6. Pandas 1.4.2 untuk memproses dataset yang digunakan pada penelitian
7. Numpy 1.22.3 digunakan untuk menghitung proses matematis pada program
8. Matplotlib 3.5.2 digunakan untuk visualisasi data dalam bentuk tabel dan grafik
9. Roboflow 0.2.4 digunakan untuk memproses dataset untuk proses augmentasi
10. Torch 3.1.0 digunakan untuk pengaplikasian penggunaan CPU dan GPU pada proses training
11. Seaborn 0.11.2 untuk analisis data dan visualisasi data
12. Requests 2.27.1 yang digunakan untuk memproses dataset yang telah diaugmentasi oleh Roboflow pada model YOLOv5

Dataset serta model yang dikembangkan akan disimpan dalam perangkat keras yang digunakan untuk proses penelitian.

## 4.2. Hasil Penelitian

Penelitian yang dilakukan berdasarkan pemaparan pada bab tiga, dilakukan dengan nilai epoch yang sama untuk kedua model SSD MobileNet v2 FPNLite 320 x 320 dan YOLOv5s. Dengan menggunakan dataset yang sama, epoch yang sama, dan resolusi citra yang sama untuk *training* pada kedua model. Kedua model menunjukkan hasil yang serupa, namun terdapat beberapa perbedaan dan dapat terlihat bahwa terdapat salah satu model yang lebih baik dibandingkan yang lainnya. Berikut merupakan perbandingan dari proses *training* yang dilakukan pada kedua model.

### 4.2.1. Analisis *Loss* dan *Learning Rate* Model

Tabel 4.1 Evaluasi Hasil *Training* Model SSD FPNLite 320 x 320

epoch	loss/ clasification loss	loss/ localization loss	loss/ regularization_loss	loss/total loss	learning rate
100	0,72261536	0,39293328	0,15355366	1,2691023	0,0319994
200	0,6489808	0,40213713	0,15386882	1,20498675	0,0373328
1000	0,39879623	0,18736778	0,15577719	0,7419412	<b>0,08</b>
2000	0,21922627	0,124170415	0,15541244	0,498809125	0,07991781
3000	0,1744589	0,057357382	0,1522331	0,384049382	0,0796716
4000	0,10811846	0,04903763	0,14846607	0,30562216	0,07926236
5000	0,14008714	0,045465555	0,14480637	0,330359065	0,078691795
6000	0,103425354	0,05334888	0,1400541	0,296828334	0,07796223
7000	0,08387594	0,025046866	0,13527259	0,244195396	0,07707667
8000	<b>0,07951175</b>	0,03853824	0,13078646	0,24883645	0,07603875
9000	0,105506234	0,04513723	0,12638944	0,277032904	0,07485275
9500	0,060747217	0,030078378	0,124284625	<b>0,21511022</b>	0,074205705
9900	0,08624473	<b>0,024159525</b>	0,122492895	0,23289715	0,073662736
10000	0,15893276	0,032693155	<b>0,12199713</b>	0,313623045	0,07352352

Tabel 4.1 menunjukkan evaluasi dari hasil *training* Model SSD Mobilenet v2 FPNLite 320 x 320 dengan epoch sebesar 10000. Pada program yang dibuat menggunakan model SSD Mobilenet v2 FPNLite 320 x 320 proses *training* tidak menggunakan *early stopping* sehingga proses *training* tetap dilaksanakan hingga selesai walaupun perubahan pada model tidak lagi signifikan. Proses *training* dari

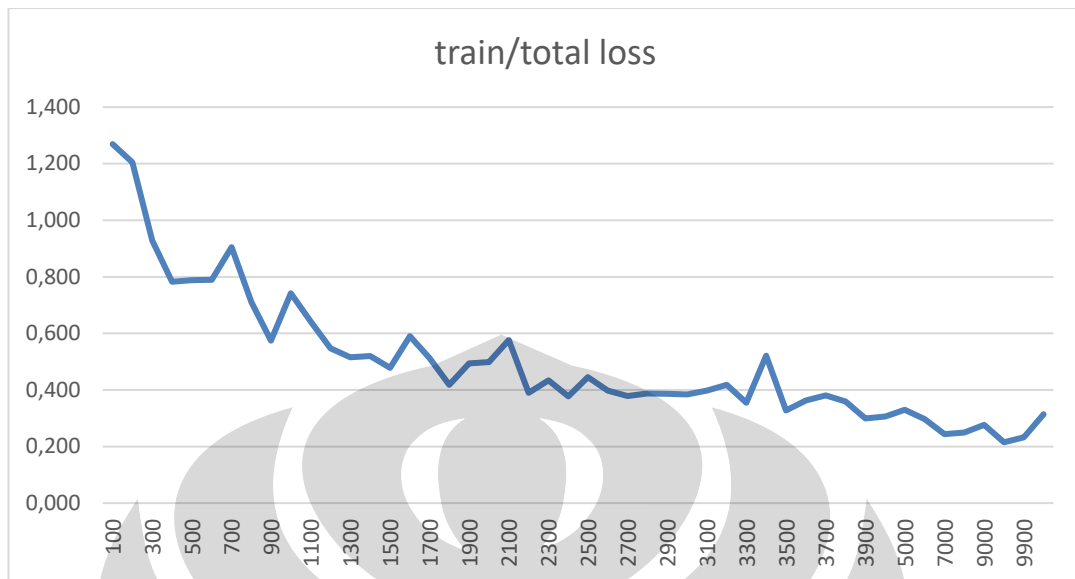
Model SSD Mobilenet v2 FPNLite 320 x 320 dilaksanakan selama 1 jam 4 menit. Program menunjukkan hasil dari setiap Langkah *training* setiap 100 epoch, sehingga hasil evaluasi yang dihasilkan merupakan perubahan yang terjadi pada model setiap 100 epoch telah berlalu. Karena terdapat 10000 epoch, maka peneliti mencantumkan perubahan yang terjadi pada model setiap 1000 epoch. Learning rate dari model SSD Mobilenet v2 FPNLite 320 x 320 memiliki nilai terbesar pada saat epoch 1000 yaitu 0,08. Learning rate dari model mengalami kenaikan drastis saat epoch 100 hingga 1000, dimana nilai learning rate pada saat epoch 100 adalah 0,032. Setelah melalui titik tertinggi bagi learning rate model pada epoch 1000, untuk epoch selanjutnya learning rate hanya mengalami penurunan saja hingga akhirnya learning rate dari model berhenti di angka 0,0735 pada epoch 10000, yaitu epoch terakhir dari percobaan. Rata – rata kecepatan dari proses *training* untuk setiap 1 epoch adalah 0,39503 detik. Dimana untuk 100 epoch pertama kecepatan proses *training* adalah 0,893 detik dan untuk epoch 200 dan seterusnya, kecepatan *training* tidak pernah melebihi 0,398 detik. Pada saat Learning rate dari model mencapai nilai tertinggi pada 0,08 di epoch ke 1000. Total loss yang dimiliki oleh model pada saat itu adalah 0,742 dimana nilai total loss masih terus berkurang hingga nilai paling kecil adalah pada epoch 9500 dengan nilai total loss 0,215 dengan perincian yaitu classification loss sebesar 0,060, localization loss sebesar 0,03, dan regularization loss sebesar 0,124. Nilai Total Loss dari Model SSD MobileNet v2 FPNLite 320 x 320 memiliki nilai paling kecil pada saat epoch 9500 yaitu sebesar 0,21. Namun untuk keefesiensian program, dengan menggunakan 7000 epoch saja, model bisa mendapatkah hasil yang hampir serupa dengan menggunakan epoch 10000. Pada epoch 7000 model memiliki nilai loss dan learning rate yang tidak jauh berbeda dengan epoch 10000. Dan dengan menggunakan epoch 7000 waktu yang diperlukan untuk proses training juga akan berkurang dengan estimasi sebesar 20 menit lebih cepat.

Tabel 4.2 Evaluasi Hasil *Training* Model YOLOv5s

epoch	train/ box_loss	train/ obj_loss	train/ cls_loss	train/ total_loss	x/lr0
0	0,096613	0,014858	0,084933	0,196404	0.003295
1	0,066863	0,014685	0,079686	0,161234	0.0066277
50	0,03481	0,0072638	0,0090112	0,051085	<b>0.0099515</b>
100	0,030002	0,0064443	0,0058831	0,0423294	0.009902
150	0,027128	0,006098	0,0047654	0,0379914	0.0098525
200	0,025566	0,0059554	0,0043493	0,0358707	0.009803
250	0,023356	0,0054062	0,0035334	0,0322956	0.0097535
300	0,022667	<b>0,0051882</b>	0,004296	0,0321512	0.009704
350	<b>0,021835</b>	0,0054113	0,0034056	<b>0,0306519</b>	0.0096545
390	0,022828	0,0052778	<b>0,0032537</b>	0,0313595	0.0096149

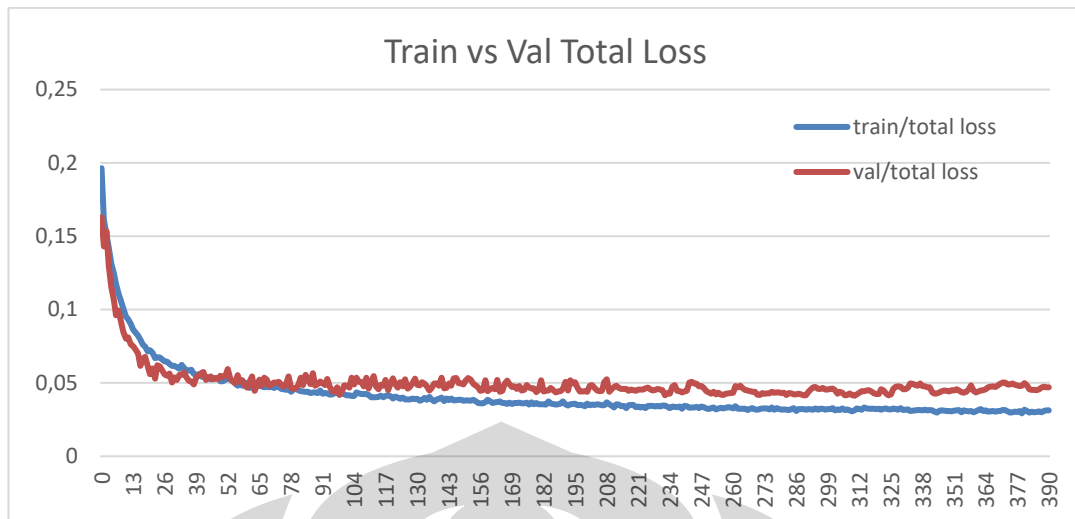
Tabel 4.2 menunjukkan evaluasi dari hasil *training* Model YOLOv5s dengan epoch sebesar 10000. Pada model YOLOv5s digunakan *early stopping* sehingga proses *training* akan selesai dengan sendirinya saat model menentukan sudah tidak terjadi perubahan yang signifikan lagi pada model. Model YOLOv5s melakukan proses *training* selama 1 jam 15 menit. Program menunjukkan hasil untuk setiap epoch *training*. Karena terdapat 390 epoch maka peneliti mencantumkan perubahan yang terjadi pada model setiap 50 epoch. Proses *training* pada model YOLOv5s dimulai pada epoch 0 dengan box loss *training* sebesar 0,097, objek loss *training* sebesar 0,0149, dan classification loss *training* sebesar 0,0849. Learning rate dari model YOLOv5s yaitu lr0, dimana lr0 adalah *initial learning rate*. Learning rate dari model YOLOv5s memiliki nilai 0,0033 pada epoch 0 yaitu saat *training* dimulai dan nilai dari learning rate mengalami kenaikan setiap epochnya hingga mencapai angka 0,0099 pada epoch 3 hingga 100 dan dari situ nilai *learning rate* menurun perlahan hingga berhenti pada angka 0,0096 pada epoch 390. Diakhir proses *training*, *total loss* dari model YOLOv5s memiliki nilai 0,031 dan *learning rate* sebesar 0,0096. Berdasarkan data Tabel 4.1 dan 4.2, dapat dilihat bahwa learning rate dari model YOLOv5s lebih rendah dibandingkan model SSD MobileNet v2 FPNLite 320 x 320 yaitu 0,009 : 0,08. Hal inilah yang menjadi salah satu faktor yang mempengaruhi kecepatan proses learning dari kedua model tersebut.

#### 4.2.2. Analisis Grafik Total Loss



Gambar 4.1 Grafik Total Loss Training Model SSD MobileNet v2 FPNLite 320 x 320

Gambar 4.1 adalah grafik yang dihasilkan menggunakan data dari Total Loss yang dimiliki oleh Model SSD MobileNet v2 FPNLite 320 x 320 pada proses *training*. Grafik menunjukkan bahwa model SSD MobileNet v2 FPNLite 320 x 320 memiliki saturasi pada epoch 2000. Dimana saat epoch 3900, nilai loss dari model SSD tidak pernah melebihi 0.4 dan dari epoch 4000 hingga 10000 nilai total loss yang dimiliki oleh model saat proses training tidak terlalu jauh berbeda yaitu berkisaran pada 0.2 hingga 0.35.



Gambar 4.2 Grafik Total Loss Training dan Validasi Model YOLOv5s

Gambar 4.2 adalah grafik yang dihasilkan menggunakan data dari Total Loss yang dimiliki oleh Model YOLOv5s pada proses *training* dan *validation*. Grafik menunjukkan bahwa model YOLOv5 nilai loss mengalami 10 epoch pertama. Kemudian nilai total loss turun perlahan – lahan hingga berakhir di epoch 390 dengan nilai total loss pada proses *training* adalah 0,031 dan pada proses *validation* adalah 0,047. Nilai loss yang dimiliki oleh model YOLOv5s pada training lebih rendah dibandingkan pada validasi namun jarak antar keduanya sangatlah kecil yaitu 0,016. Maka dapat disimpulkan bahwa nilai total loss dari model YOLOv5s sangatlah rendah.

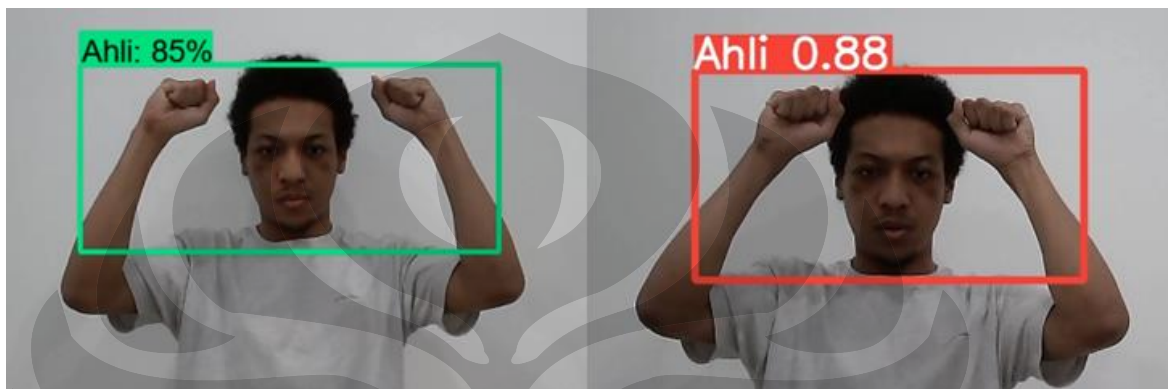


Gambar 4.3 Perbandingan grafik Total Loss Training Model SSD MobileNet v2 FPNLite 320 x 320 vs YOLOv5s

Gambar 4.3 menunjukkan perbandingan grafik total loss antara Model SSD MobileNetv2 FPNLite 320 x 320 dan YOLOv5s. Karena epoch yang digunakan oleh model YOLOv5s jauh lebih sedikit dibandingkan model SSD MobileNetv2 FPNLite 320 x 320 maka grafik yang dihasilkan oleh model YOLOv5s terlihat lebih kecil dibandingkan SSD MobileNetv2 FPNLite 320 x 320. Nilai total loss dari model YOLOv5s juga jauh lebih kecil dibandingkan SSD MobileNet v2 FPNLite 320 x 320 dimana pada epoch 0 sudah ditunjukkan bahwa model YOLOv5s memulai lossnya dari kisaran 0,1 loss, sedangkan loss pada model SSD MobileNet v2 FPNLite 320 x 320 memiliki loss sebesar 1. Dan pada akhir proses training loss yang dimiliki oleh model YOLOv5s hampir mendekati 0 sedangkan model SSD MobileNet v2 FPNLite 320 x 320 masih memiliki nilai loss mendekati 0,3.

#### 4.2.3. Hasil Percobaan *Testing*

Dengan menggunakan 3 citra dataset *testing* yang telah dibuat oleh peneliti, dilakukan percobaan dengan mendeteksi objek pada citra pada kedua Model yang digunakan untuk penelitian. Hasil deteksi objek menggunakan Model SSD MobileNet v2 FPNLite 320 x 320 ditunjukkan dalam persen sedangkan Model YOLOv5 ditunjukkan dalam angka decimal.



Gambar 4.4 Percobaan deteksi objek “Ahli” dengan menggunakan Model SSD MobileNet v2 FPNLite 320 x 320 (atas) dan YOLOv5s (bawah)

Gambar 4.4 menunjukkan hasil testing dari percobaan kedua model setelah training selesai menggunakan kamera secara real – time dengan memperagakan kata “Ahli”. Model SSD MobileNet v2 FPNLite 320 x 320 menunjukkan hasil *confidence score* sebesar 85% sedangkan model YOLOv5 memberikan hasil *confidence score* sebesar 88% untuk kata “Ahli”. Ini menunjukkan bahwa kedua model dengan menggunakan dataset yang sama akan memiliki hasil yang serupa.



Tabel 4.3 Hasil Pengujian Dataset Testing I

No	Kata	SSD MobileNet	YOLOv5
1	Ahli	Benar	Benar
2	Akhiran-nya	Benar	Benar
3	Alamat	Benar	Benar
4	Anda	Benar	Benar
5	Apa	Benar	Benar
6	Awalan-Pe	Benar	Benar
7	Bulan	Salah (FN)	Benar
8	Cepat	Benar	Benar
9	Daftar	Benar	Benar
10	Dekat	Benar	Benar
11	Gaji	Benar	Benar
12	Hari	Benar	Benar
13	Ingin	Benar	Benar
14	Jauh	Benar	Benar
15	Karena	Benar	Benar
16	Kerja	Benar	Benar
17	Lupa	Salah (FN)	Benar
18	Maaf	Benar	Benar
19	Minat	Benar	Benar
20	Orang	Benar	Benar
21	Prestasi	Benar	Benar
22	Rumah	Benar	Benar
23	Saya	Benar	Benar
24	Sini	Salah (FN)	Benar
25	Tanggap	Benar	Benar
26	Tidak	Benar	Benar
27	Ya	Benar	Benar
Total Akurasi		24 Benar	27 Benar

Tabel 4.3 menunjukkan Hasil pengujian dari kedua model terhadap dataset testing I untuk 27 kata bahasa isyarat yang diuji. Model SSD MobileNet v2 FPNLite 320 x 320 dari 27 objek berhasil mendeteksi 24 objek dengan benar dimana kata “Bulan”, “Lupa”, dan “Sini” mendapatkan hasil *False Negative* dimana berarti objek gagal mendeteksi objek yang seharusnya dideteksi dengan benar.



Gambar 4.5 Dataset Testing I “Bulan”, “Lupa”, “Sini” pada model SSD MobileNetv2 FPNLite 320 x 320

Gambar 4.5 menunjukkan saat Dataset Testing I “Bulan”, “Lupa”, dan “Sini” menunjukkan hasil *False Negative* saat dilakukan testing menggunakan Model SSD MobileNet v2 FPNLite 320 x 320. Hal ini terjadi dikarenakan model SSD gagal mendeteksi objek yang dianggap tidak serupa dengan apa yang telah dipelajari oleh model SSD selama proses *training*. Hal ini mungkin terjadi karena adanya ketidaksesuaian gerakan yang dideteksi oleh model pada ketiga gerakan tersebut antara dataset yang digunakan untuk *testing* dan dengan apa yang dimiliki model saat *training*. Dari hasil pengujian pada dataset testing I didapatkan *confidence score* model sebesar  $(24/27) \times 100\% = 88,89\%$ . Sedangkan Model YOLOv5 berhasil mendeteksi 27 objek dengan benar sehingga memiliki *confidence score* model sebesar  $(27/27) \times 100\% = 100\%$ .

Tabel 4.4. Hasil Pengujian Dataset Testing II

No	Kata	SSD MobileNet	YOLOv5
1	Ahli	Benar	Benar
2	Akhiran-nya	Benar	Benar
3	Alamat	Benar	Benar
4	Anda	Benar	Benar
5	Apa	Benar	Benar
6	Awalan-Pe	Benar	Benar
7	Bulan	Benar	Benar
8	Cepat	Benar	Benar
9	Daftar	Benar	Benar
10	Dekat	Benar	Benar
11	Gaji	Benar	Benar
12	Hari	Salah (FN)	Benar
13	Ingin	Salah (FN)	Benar
14	Jauh	Benar	Benar
15	Karena	Benar	Benar
16	Kerja	Benar	Benar
17	Lupa	Salah (FN)	Benar
18	Maaf	Benar	Benar
19	Minat	Benar	Benar
20	Orang	Benar	Benar
21	Prestasi	Benar	Benar
22	Rumah	Benar	Benar
23	Saya	Benar	Benar
24	Sini	Salah (FN)	Benar
25	Tanggap	Benar	Benar
26	Tidak	Benar	Benar
27	Ya	Benar	Benar
Total Akurasi		23 Benar	27 Benar

Tabel 4. 4 menunjukkan Hasil pengujian dari kedua model terhadap dataset testing II untuk 27 kata bahasa isyarat yang diuji. Model SSD MobileNet v2 FPNLite 320 x 320 dari 27 objek berhasil mendeteksi 24 objek dengan benar dimana kata “Hari”, “Ingin”, “Lupa”, dan “Sini” mendapatkan hasil *False Negative*.



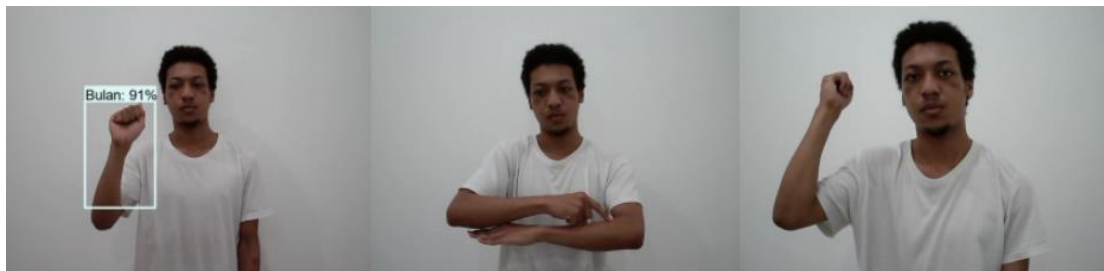
Gambar 4.6 Dataset Testing II “Ingin” dan “Hari” (atas), “Lupa” dan “Sini”(bawah) pada model SSD MobileNetv2 FPNLite 320 x 320

Gambar 4.6 menunjukkan saat Dataset Testing I “Ingin”, “Hari”, “Lupa” dan “Sini” menunjukkan hasil *False Negative* saat dilakukan testing menggunakan Model SSD MobileNet v2 FPNLite 320 x 320. Hal ini terjadi dikarenakan model SSD gagal mendeteksi objek yang dianggap tidak serupa dengan apa yang telah dipelajari oleh model SSD selama proses *training*. Dari hasil pengujian ini maka didapatkan *confidence score* model sebesar  $(23/27) \times 100\% = 85,19\%$ . Sedangkan Model YOLOv5 berhasil mendeteksi 27 objek dengan benar sehingga memiliki *confidence score* model sebesar  $(27/27) \times 100\% = 100\%$ .

Tabel 4.5 Hasil pengujian Dataset Testing III

No	Kata	SSD MobileNet	YOLOv5
1	Ahli	Benar	Benar
2	Akhiran-nya	Benar	Benar
3	Alamat	Benar	Benar
4	Anda	Salah (FP)	Benar
5	Apa	Benar	Benar
6	Awalan-Pe	Benar	Benar
7	Bulan	Benar	Benar
8	Cepat	Benar	Benar
9	Daftar	Benar	Benar
10	Dekat	Benar	Benar
11	Gaji	Benar	Benar
12	Hari	Salah (FN)	Benar
13	Ingin	Benar	Benar
14	Jauh	Benar	Benar
15	Karena	Benar	Benar
16	Kerja	Benar	Benar
17	Lupa	Salah (FN)	Benar
18	Maaf	Benar	Benar
19	Minat	Benar	Benar
20	Orang	Benar	Benar
21	Prestasi	Benar	Benar
22	Rumah	Benar	Benar
23	Saya	Benar	Benar
24	Sini	Benar	Benar
25	Tanggap	Benar	Benar
26	Tidak	Benar	Benar
27	Ya	Benar	Benar
Total Akurasi		24 Benar	27 Benar

Tabel 4.5 menunjukkan Hasil pengujian dari kedua model terhadap dataset testing III untuk 27 kata bahasa isyarat yang diuji. Model SSD MobileNet v2 FPNLite 320 x 320 dari 27 objek berhasil mendeteksi 24 objek dengan benar dimana kata “Anda” mendapatkan hasil *False Positive* dimana model berhasil mendeteksi objek pada citra tetapi kata yang dideteksi tidak sesuai dengan kata apa yang seharusnya dideteksi oleh sistem sedangkan kata “Hari dan “Lupa” mendapatkan hasil *False Negative*.



Gambar 4. 7 Dataset Testing III “Anda”, “Hari”, “Lupa” pada model SSD MobileNetv2 FPNLite 320 x 320

Gambar 4.7 menunjukkan saat Dataset Testing III “Anda” yang menunjukkan hasil *False Positive* dan Dataset Testing III “Hari” dan “Lupa” yang menunjukkan hasil *False Negative* saat dilakukan testing menggunakan Model SSD MobileNet v2 FPNLite 320 x 320. Pada kata “Anda” Model SSD MobileNet v2 FPNLite 320 x 320 memprediksi gerakan tersebut sebagai kata “Bulan” yang memang mirip gerakannya dengan kata “Anda” dimana terdapat dataset *training* untuk kata Bulan yang melakukan gerakan yang mirip dengan kata “Anda”. Sedangkan untuk kata “Hari” dan “Lupa” gagal terdeteksi, hal ini terjadi dikarenakan model SSD gagal mendeteksi objek yang dianggap tidak serupa dengan apa yang telah dipelajari oleh model SSD selama proses *training*. Dari hasil pengujian ini maka didapatkan *confidence score* model sebesar  $(24/27) \times 100\% = 88,89\%$ . Sedangkan Model YOLOv5 berhasil mendeteksi 27 objek dengan benar sehingga memiliki *confidence score* model sebesar  $(27/27) \times 100\% = 100\%$ .

Dari ketiga hasil pengujian pada masing – masing dataset testing, dan dengan menggunakan rumus perhitungan *precision*, *recall*, dan *F1 Score*, maka didapatkan hasil untuk kedua model sebagai berikut :

1. Model SSD MobileNet v2 FPNLite 320 x 320 memiliki hasil :

- Rata – rata *confidence score* sebesar 0,8766
- *Precision* sebesar  $(71/72) = 0,9861$
- *Recall* sebesar  $(71/80) = 0,8875$
- *F1* sebesar  $2 \times \frac{0,9861 \times 0,8875}{0,9861 + 0,8875} = 0,9342$

2. Model YOLOv5s memiliki hasil :

- Rata – rata *confidence score* sebesar 1
- *Precision* sebesar  $(81/81) = 1$

- *Recall* sebesar  $(81/81) = 1$
- *F1* sebesar  $2 \times \frac{1 \times 1}{1 + 1} = 1$

Dengan hasil perhitungan diatas maka dapat dilihat bahwa Model YOLOv5 memiliki keandalan yang lebih tinggi dibandingkan dengan model SSD MobileNet v2 FPNLite 320 x 320 diukur dengan menggunakan perhitungan *Precision*, *Recall*, dan *F1 Score* dimana nilai *F1 Score testing* dari model YOLOv5 memiliki nilai 1, sedangkan untuk Model SSD MobileNetv2 FPNLite 320 x 320 memiliki *F1 Score* sebesar 0,9342. Namun perlu diperhatikan bahwa penelitian ini menggunakan *sample* sebanyak 27 kata dari Bahasa Isyarat Indonesia SIBI sehingga dengan semakin banyaknya kata – kata yang digunakan untuk penelitian, maka kemungkinan untuk nilai *error* akan semakin besar juga dan nilai banding untuk rata – rata *confidence score*, *precision*, *recall*, dan *F1 Score* dari kedua model akan berubah.

#### 4.2.4. Pengujian Kalimat

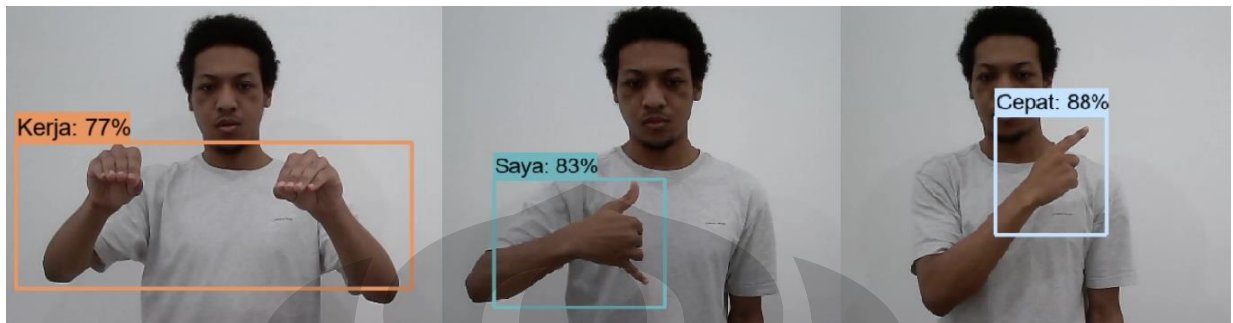
Pengujian pada kedua model secara *real-time* menggunakan kamera dengan menggunakan *library* OpenCV. Untuk memenuhi kebutuhan penelitian, peneliti telah menyiapkan 3 contoh pertanyaan yang mungkin untuk ditanyakan dalam proses wawancara nyata. Pertanyaan akan diajukan oleh penanya menggunakan Bahasa Indonesia dan jawaban akan diberikan oleh pengguna Bahasa Isyarat yang kemudian akan dideteksi secara *real-time* menggunakan sistem yang telah dirancang. Pada bagian ini akan ditunjukkan seberapa tepat objek yang berupa jawaban dari pengguna Bahasa Isyarat berhasil dideteksi oleh Model SSD MobileNet v2 FPNLite 320 x 320 dan Model YOLOv5s.



1. Apa kelebihan anda ?

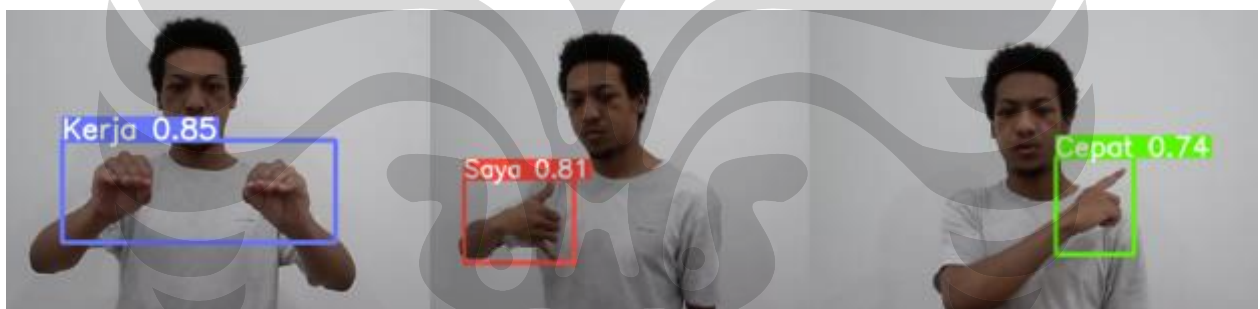
a. “Kerja saya cepat”

- Model SSD MobileNet v2 FPNLite 320 x 320 (Gambar 4.8 menghasilkan rata – rata *confidence score* 82%)



Gambar 4.8 "Kerja" "Saya" "Cepat" (Model SSD MobileNet v2 FPNLite 320 x 320)

- Model YOLOv5s (Gambar 4.9 menghasilkan rata – rata *confidence score* 80%)



Gambar 4.9 "Kerja" "Saya" "Cepat" (Model YOLOv5s)

b. “Saya cepat Tanggap”

- Model SSD MobileNet v2 FPNLite 320 x 320 (Gambar 4.10 menghasilkan rata – rata *confidence score* 81%)



Gambar 4.10 "Saya" "Cepat" "Tanggap" (Model SSD MobileNet v2 FPNLite 320 x 320)



- Model YOLOv5s (Gambar 4.11 menghasilkan rata – rata *confidence score* 80,3%)

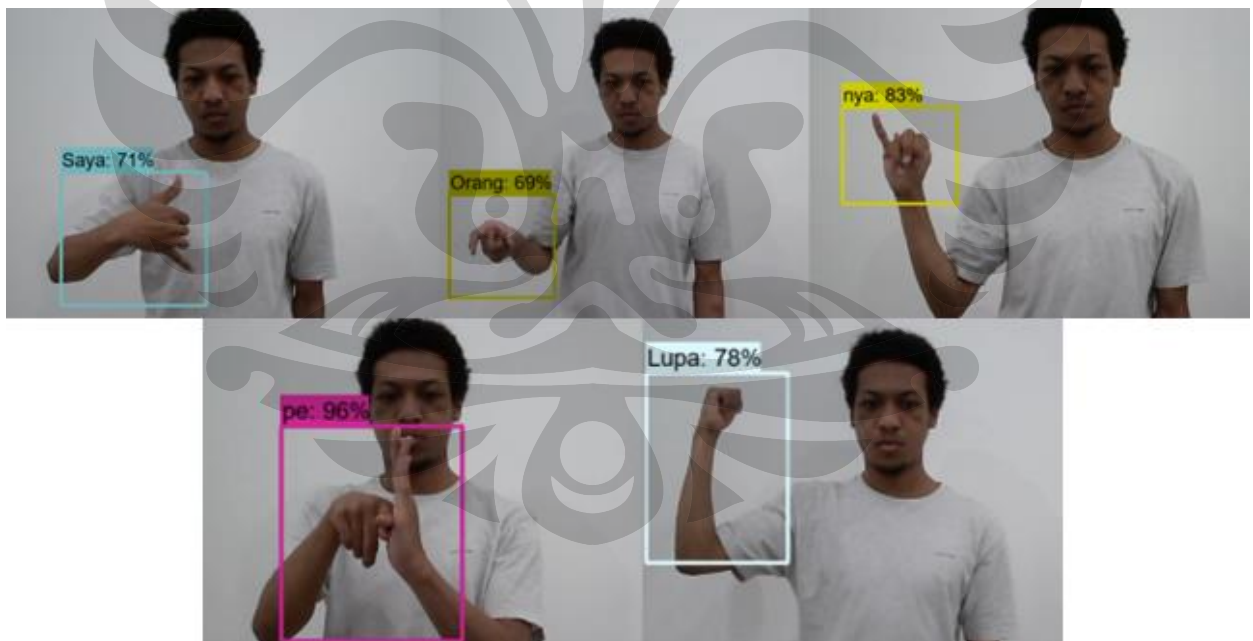


Gambar 4.11 "Saya" "Cepat" "Tanggap" (Model YOLOv5s)

## 2. Apa kekurangan anda ?

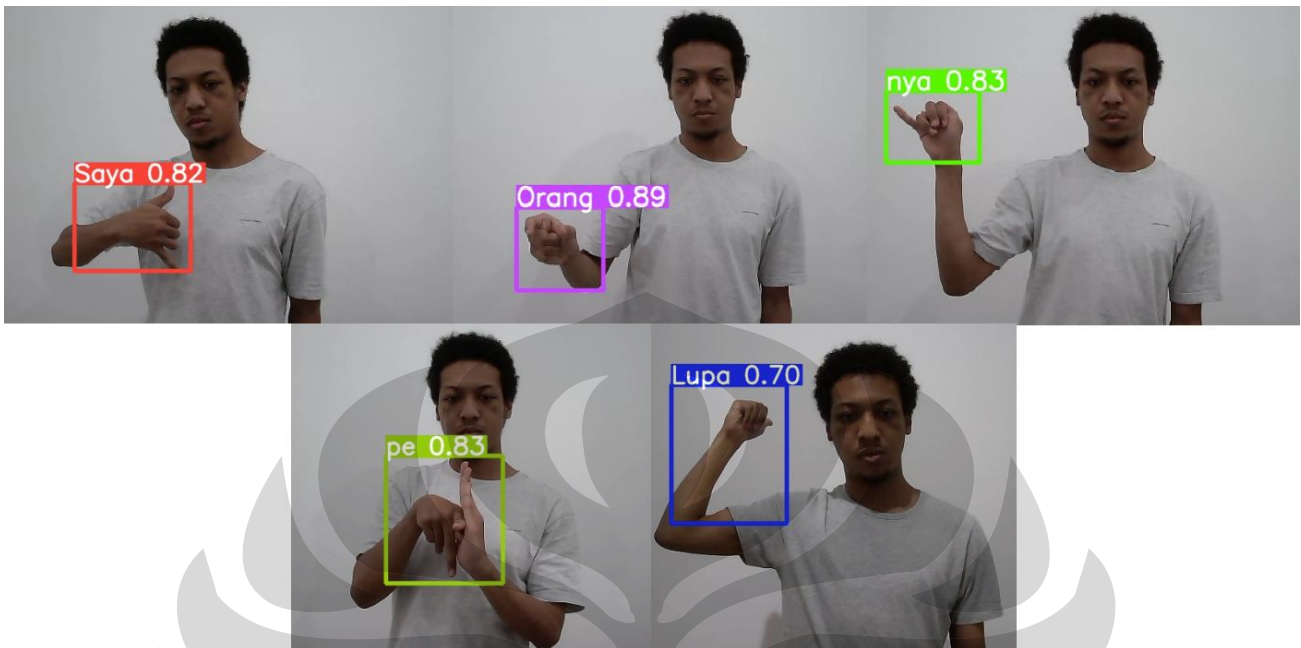
“Saya orang-nya pe-lupa”

- Model SSD MobileNet v2 FPNLite 320 x 320 (Gambar 4.12 menghasilkan rata – rata *confidence score* 79,4%)



Gambar 4.12 "Saya" "Orang" "-Nya" "Pe-" "Lupa" (Model SSD MobileNet v2 FPNLite 320 x 320)

- Model YOLOv5s (Gambar 4.13 menghasilkan rata – rata *confidence score* 81,4%)

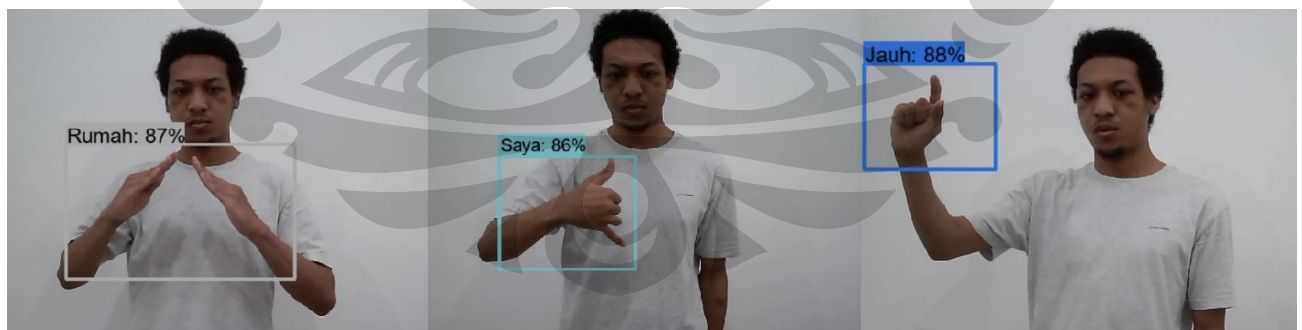


Gambar 4.13 "Saya" "Orang" "-Nya" "Pe-" "Lupa" (Model YOLOv5s)

3. Apa tantangan terberat anda untuk bekerja di perusahaan ini ?

“Rumah saya jauh”

- Model SSD MobileNet v2 FPNLite 320 x 320 (Gambar 4.14 menghasilkan rata – rata *confidence score* 87%)



Gambar 4.14 "Rumah" "Saya" "Jauh" (Model SSD MobileNet v2 FPNLite 320 x 320)

- Model YOLOv5s (Gambar 4.15 menghasilkan rata – rata *confidence score* 82,3%)



Gambar 4.15 "Rumah" "Saya" "Jauh" (Model YOLOv5s)

Gambar 4.8, 4.9, 4.10, 4.11, 4.12, 4.13, 4.14, dan 4.15 menunjukkan hasil deteksi Bahasa Isyarat menggunakan kedua model SSD MobileNet v2 FPNLite 320 x 320 dan YOLOv5 secara *real-time* dalam menjawab pertanyaan yang biasa ditanyakan saat proses wawancara pekerjaan. Pada gambar dapat dilihat akurasi dari sistem bahwa akurasi paling rendah dari model SSD MobileNet v2 FPNLite 320 x 320 ada pada angka 69% yaitu pada kata “Orang” sedangkan pada Model YOLOv5 akurasi paling rendah ada pada angka 70% yaitu pada kata “Tanggap”.

Dengan menggunakan nilai rata – rata *confidence score* untuk setiap kalimat, maka didapatkan hasil rata – rata *confidence score* untuk setiap model dari 4 kalimat yang telah berhasil diuji adalah Model SSD MobileNet v2 FPNLite 320 x 320 memiliki nilai rata – rata *confidence score* sebesar 82,35% sedangkan Model YOLOv5s memiliki nilai rata – rata *confidence score* sebesar 81%. Dimana dapat dilihat bahwa Model SSD MobileNet v2 FPNLite memiliki *confidence score* yang lebih tinggi dibandingkan Model YOLOv5s dalam pengujian pembuatan kalimat.

Pengambilan contoh gambar menggunakan metode *real-time* menunjukkan bahwa kedua model SSD MobileNet v2 FPNLite 320 x 320 dan YOLOv5 berhasil mendeteksi semua kata yang digunakan untuk menjawab contoh pertanyaan, ini menandakan bahwa kedua model dapat digunakan untuk keperluan penelitian yaitu untuk membantu pengguna Bahasa Isyarat dalam melaksanakan sesi wawancara saat melamar kerja.

## BAB 5

### KESIMPULAN DAN SARAN

Penelitian sistem pengenalan Bahasa Isyarat menggunakan Model SSD MobileNet v2 FPNLite 320 x 320 dan YOLOv5s dapat disimpulkan bahwa :

1. Sistem Pengenalan Bahasa Isyarat dalam waktu nyata dapat direalisasikan menggunakan algoritma SSD.
2. Sistem Pengenalan Bahasa Isyarat dalam waktu nyata dapat direalisasikan menggunakan algoritma YOLOv5.
3. *Confidence score* sistem saat *testing* dari Model YOLOv5s lebih tinggi dibanding Model SSD MobileNet v2 FPNLite 320 x 320 yaitu 100% : 87,66%.
4. Nilai *F1 Score* dari Model YOLOv5 lebih tinggi dibandingkan Model SSD MobileNet v2 FPNLite 320 x 320 yaitu 1 : 0,9342
5. Proses training Model YOLOv5 menunjukkan saturasi penurunan loss saat training setiap epochnya lebih cepat dibandingkan model SSD.
6. Proses training Model SSD Mobilenet v2 FPNLite 320 x 320 untuk setiap epochnya hanya membutuhkan waktu kurang dari 1 detik yang merupakan waktu yang lebih cepat dibandingkan Model YOLOv5.
7. Nilai Total Loss yang dimiliki oleh Model YOLOv5 lebih rendah dibandingkan Model SSD MobileNet v2 FPNLite 320 x 320 meskipun epoch yang digunakan oleh Model SSD MobileNet v2 FPNLite 320 x 320 jauh lebih banyak dibanding Model YOLOv5.
8. Learning Rate dari Model SSD MobileNet v2 FPNLite 320 x 320 lebih tinggi dibanding Model YOLOv5 yaitu 0,08 : 0,009
9. Proses pengujian pembuatan kalimat menggunakan kedua model menunjukkan hasil yang serupa dimana semua kata berhasil dideteksi dan disusun sebagai suatu kalimat.
10. Model terbaik diantara Model SSD dan YOLOv5 untuk deteksi bahasa isyarat adalah Model YOLOv5 karena hasil penelitian menunjukkan bahwa nilai *precision*, *recall*, dan *F1 Score* dari Model YOLOv5 melebihi nilai Model SSD MobileNetv2 FPNLite 320 x 320.

11. Untuk keefensiensian, lebih baik menggunakan Model SSD MobileNet v2 FPNLite 320 x 320 untuk keperluan Bahasa Isyarat untuk wawancara pekerjaan karena waktu yang diperlukan untuk training lebih singkat untuk mendapatkan hasil yang hampir serupa dalam proses pengujian pembuatan kalimat.

Untuk penelitian selanjutnya untuk meningkatkan kinerja dan keefektifan sistem maka peneliti memiliki beberapa saran diantaranya adalah :

1. Memperbanyak *class* dataset sehingga pilihan kata dan kalimat yang bisa dibentuk menggunakan kata berbeda akan semakin bervariasi dan semakin banyak.
2. Pengumpulan dataset untuk penelitian ini hanya dilakukan oleh peneliti saja, baiknya untuk meningkatkan variasi data pengambilan dataset sebaiknya dilakukan oleh lebih dari satu orang.
3. Model masih mengalami kesulitan untuk mendeteksi gerakan yang terlalu Panjang dalam satu urutan gerakan yang sama, maka disarankan untuk memperbanyak dataset untuk setiap frame bagi gerakan Bahasa Isyarat yang memiliki urutan yang banyak.
4. Memberikan Output berupa string dalam bentuk file .txt atau *alert* yang menandakan kalimat yang sudah berhasil disusun menggunakan sistem pendeteksi objek yang sudah dibuat.

## DAFTAR PUSTAKA

- [1] O. M. Sincan and H. Y. Keles, "Using Motion History Images with 3D Convolutional Networks in Isolated Sign Language Recognition," pp. 1–14, 2021, [Online]. Available: <http://arxiv.org/abs/2110.12396>.
- [2] T. E. of E. Britannica, "sign language," *Encyclopedia Britannica*, 2020. <https://www.britannica.com/topic/sign-language> (accessed Dec. 15, 2021).
- [3] ai Media, "Sign Language Alphabets From Around The World," 2020. <https://www.ai-media.tv/sign-language-alphabets-from-around-the-world/> (accessed Dec. 15, 2021).
- [4] A. Ansori Al Nasihudin, "Mengenal 2 Sistem Bahasa Isyarat di Indonesia, Mana yang Lebih Digemari?," 2020. <https://www.liputan6.com/disabilitas/read/4385778/mengenal-2-sistem-bahasa-isyarat-di-indonesia-mana-yang-lebih-digemari> (accessed Dec. 16, 2021).
- [5] Yayasan Peduli Kasih Anak Berkebutuhan Khusus, "Mengenal Bahasa Isyarat," 2017. <https://www.yedulikasihabk.org/2018/11/09/mengenal-bahasa-isyarat/> (accessed Dec. 16, 2021).
- [6] Klobility, "BISINDO dan SIBI: Apa Bedanya?," 2021. <https://www.klobility.id/post/perbedaan-bisindo-dan-sibi> (accessed Dec. 16, 2021).
- [7] Kementerian Pendidikan dan Kebudayaan dengan Lembaga Penelitian dan Pengembangan Sistem Isyarat Bahasa Indonesia, "Kamus SIBI," 2022. <https://pmpk.kemdikbud.go.id/sibi/> (accessed Jan. 06, 2021).
- [8] N. Rahmalia, "Kenalan dengan Machine Learning, Sebuah Cabang Ilmu Kecerdasan Buatan," 2021. <https://glints.com/id/lowongan/machine-learning/> (accessed Dec. 17, 2021).
- [9] IBM Cloud Education, "What is Machine Learning," 2020.

- <https://www.ibm.com/cloud/learn/machine-learning> (accessed Dec. 17, 2021).
- [10] ROOT0x57B, “Detect an Object with OpenCV-Python,” 2020. <https://www.geeksforgeeks.org/detect-an-object-with-opencv-python/> (accessed Dec. 17, 2021).
- [11] OpenCV, “About - OpenCV,” 2022. <https://opencv.org/about/> (accessed Jan. 07, 2022).
- [12] W. Liu *et al.*, “SSD: Single shot multibox detector,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 9905 LNCS, pp. 21–37, 2016, doi: 10.1007/978-3-319-46448-0\_2.
- [13] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, pp. 779–788, 2016, doi: 10.1109/CVPR.2016.91.
- [14] Ultralytics, “YOLOv5 Documentation,” *Ultralytics*, 2020. <https://docs.ultralytics.com/> (accessed Jun. 15, 2022).
- [15] Ultralytics, “Tips for Best Training Results,” *YOLOv5 Documentation*, 2022. <https://docs.ultralytics.com/tutorials/training-tips-best-results/>.
- [16] A. A. Gafar and J. Y. Sari, “Sistem Pengenalan Bahasa Isyarat Indonesia dengan Menggunakan Metode Fuzzy K-Nearest Neighbor,” *J. Ultim.*, vol. 9, no. 2, pp. 122–128, 2018, doi: 10.31937/ti.v9i2.671.
- [17] vighneshbirodgar, “Tensorflow detection model zoo,” *GitHub*, 2020. [https://github.com/tensorflow/models/blob/master/research/object\\_detection/g3doc/tf2\\_detection\\_zoo.md](https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf2_detection_zoo.md) (accessed Feb. 20, 2022).
- [18] Glenn-jocher, “Ultralytics YOLOv5,” *GitHub*, 2022. <https://github.com/ultralytics/yolov5> (accessed Mar. 29, 2022).
- [19] J. Hui, “mAP (mean Average Precision) for Object Detection,” 2018.

<https://jonathan-hui.medium.com/map-mean-average-precision-for-object-detection-45c121a31173>.





**LAMPIRAN**