

Sign Language Translator Using Deep Learning

1st Nurul Amelia
Fakultas Teknik Elektro
Universitas Telkom

Bandung, Indonesia
nurulamelia@student.telkomuniversity.
ac.id

2nd Casi Setianingsih
Fakultas Teknik Elektro
Universitas Telkom

Bandung, Indonesia
setiacasie@telkomuniversity.ac.id

3rd Randy Erfa Saputra
Fakultas Teknik Elektro
Universitas Telkom

Bandung, Indonesia
resaputra@telkomuniversity.ac.id

Abstrak—Penelitian ini menjelaskan tentang salah satu permasalahan komunikasi yang terjadi di kehidupan masyarakat. Permasalahan komunikasi tersebut terjadi antara teman tuli dan teman dengar. Teman dengar berkomunikasi secara verbal, sedangkan teman tuli berkomunikasi dengan non-verbal. Untuk teman tuli berkomunikasi dengan menggunakan bahasa isyarat sebagai medianya. Solusi paling umum untuk saat ini adalah dengan menggunakan orang lain sebagai penerjemah untuk berkomunikasi antara teman tuli dan teman dengar. Tetapi, solusi tersebut tidak menjawab solusi yang efektif dikarenakan seorang penerjemah tidak setiap waktu akan tersedia, berbeda dengan program komputer.

Solusi yang diusulkan menerapkan teknik *deep learning* LSTM dan YOLOv5, sistem akan mengartikan bahasa isyarat SIBI. Selain itu, melalui pengembangan aplikasi Android dan pengembangan *Backend* memastikan operasi aplikasi yang andal dan efisien.

Produk berupa aplikasi dengan fitur *deep learning* LSTM untuk fitur deteksi *motion* dengan akurasi 95.56%, YOLOv5 untuk deteksi *realtime* memiliki mAP@0.5IoU 99.5%, dan LSTM untuk *Text-Correction* 92,5%. Kemudian animasi gerakan SIBI untuk fitur pembelajaran, dan implementasi ke Android melalui *backend*. Pada tes reliabilitas aplikasi ditemukan bahwa nilai r_{11} adalah 0,914428422. Oleh karena itu, dapat disimpulkan bahwa pengujian ini memiliki tingkat reliabilitas yang sangat tinggi. Respon waktu dalam hasil tes server juga menunjukkan dikisaran 1-5 detik dan *memory usage* rata-rata 130mb.

Kata kunci— SIBI, inklusivitas, Machine Learning, LSTM, YOLOv5, Android, Backend, Animasi

I. PENDAHULUAN

Tunarungu adalah orang yang kehilangan kemampuan untuk mendengar sehingga tidak dapat mendengar bunyi secara sempurna, bahkan tidak dapat mendengar sama sekali [1]. Menurut laporan BPS, pada tahun 2011 yang telah diperbarui pada Juni 2021, jumlah teman tuli di Bandung sebanyak 219 [2]. Kemudian, jumlah teman tuli di Jawa Barat sebanyak 4.019 yang diambil dari laporan BPS pada tahun 2011 yang telah diperbarui pada Juni 2021 [1]. Untuk Indonesia presentase teman tuli sebesar 7,03% [3]. Disisi lain, terdapat perbedaan bahasa antara teman tuli dan teman dengar. Berbeda dengan teman dengar yang bisa menggunakan komunikasi secara verbal, teman tuli tidak bisa berkomunikasi secara verbal melainkan menggunakan komunikasi dengan bahasa isyarat. Bahasa isyarat merupakan bahasa yang tidak memanfaatkan suara dalam berkomunikasi, melainkan memanfaatkan komunikasi manual, bahasa tubuh dan gerak bibir. Dalam berkomunikasi

tersebut akan mengkombinasikan bahasa tersebut dengan bentuk tangan, orientasi dan gerak tangan, lengan dan tubuh, serta ekspresi wajah untuk mengutarakan pikirannya [4]. Sehingga, terdapat keterbatasan komunikasi antara teman tuli dan teman dengar.

Di Indonesia, terdapat 2 standar bahasa isyarat yaitu BISINDO dan SIBI. Namun, penggunaan SIBI dianggap lebih efektif untuk berkomunikasi dan penataan kalimat dikarenakan dalam SIBI terdapat imbuhan dalam kata yaitu awalan dan akhiran seperti me-, ter-, dan lainnya. SIBI memberikan dampak baik terhadap kemampuan pengorganisasian karangan siswa di SLB. Sedangkan dalam BISINDO tidak menjawab solusi komunikasi yang efektif dikarenakan penataan kalimat yang kurang sesuai dan tanpa imbuhan (diambil dari wawancara Ibu Rini selaku guru di SLB Negeri Cicendo).

Adanya keterbatasan komunikasi antara teman tuli dan teman dengar, maka perlu dibuat sistem untuk berkomunikasi dan mempelajari bahasa isyarat. Solusi paling umum untuk saat ini adalah dengan menggunakan orang lain sebagai penerjemah untuk berkomunikasi antara teman tuli dan teman dengar. Tetapi, solusi tersebut tidak menjawab solusi yang efektif dikarenakan seorang penerjemah tidak setiap waktu akan tersedia, berbeda dengan program komputer. Sehingga, dalam solusi masalah tersebut dibutuhkan alternatif untuk proses pembelajaran bahasa isyarat. Salah satunya dapat menggunakan pengembangan suatu aplikasi sebagai media pembelajaran bahasa isyarat berbasis Android. Untuk implementasinya digunakan penerapan menggunakan algoritma YOLOv5 yang digunakan untuk mendeteksi gerakan alfabet SIBI secara *real-time*. Selain itu, terdapat beberapa integrasi lainnya pada aplikasi yang diterapkan oleh empat rekan saya, seperti SIBI *Motion Detection* menggunakan LSTM oleh Fikri Putra Hidayat, *Text to Animation* menggunakan LSTM oleh Rizqi Alpriansyah, *Backend* dan server oleh Hanifah Marta Ardilah, dan Android oleh Luthfi Yafi Alfiansyah. Sehingga, dengan adanya sistem ini dapat memudahkan antara teman tuli dan teman dengar untuk melakukan komunikasi dan mempelajari bahasa isyarat SIBI.

II. KAJIAN TEORI

A. Bahasa Isyarat

Bahasa isyarat adalah bahasa yang membutuhkan gerakan badan dan mimik muka yang berfungsi sebagai simbol bahasa untuk berkomunikasi. Bahasa isyarat tidak membutuhkan bunyi atau tulisan dalam penerapannya [5]. Bahasa isyarat dibuat secara khusus yang digunakan untuk kaum tunarungu dan beberapa yang membutuhkan gerakan bahasa isyarat.

B. Sistem Isyarat Bahasa Indonesia

Sistem Isyarat Bahasa Indonesia (SIBI) adalah bahasa isyarat yang biasanya digunakan oleh kalangan teman tuli yang berfungsi sebagai media yang membantu teman tuli dan teman dengar untuk berkomunikasi. SIBI merupakan perkembangan dari bahasa isyarat luar negeri yang ditambah dengan isyarat-isyarat lokal Indonesia dan buatan. Penggunaan SIBI menggunakan gerakan satu tangan seperti *American Sign Language* (ASL) karena SIBI diadaptasi dari ASL. SIBI merupakan bahasa isyarat yang telah ditetapkan oleh pemerintah dalam Undang-Undang Nomor 2 Tahun 1989 dan telah dibakukan oleh Menteri Pendidikan dan Kebudayaan pada 30 Juni 1994 sebagai media pembelajaran standar yang digunakan pada Sekolah Luar Biasa (SLB) [6].

C. Machine Learning

Machine Learning adalah salah satu sub-bidang atau cabang dari *Artificial Intelligence* yang mempunyai solusi lebih efisien dengan pengembangan yang sangat cepat dari tahun ke tahun dimulai dari masalah *regression*, *classification*, *clustering* sampai dengan *anomaly detection*. Misalnya pengenalan gestur berdasarkan video digital atau citra. Adapun perkembangan tersebut membuat kreatifitas peneliti untuk mengembangkan model dan menghasilkan *prototype* produk teknologi cerdas. *Machine Learning* adalah bagian dari *Artificial Intelligence* yang banyak digunakan untuk memecahkan suatu masalah. *Machine Learning* dapat didefinisikan sebagai algoritma matematika yang terdefinisi dengan baik yang diambil dari pembelajaran pada struktur data dan akan menghasilkan prediksi di masa yang akan datang [7].

D. You Only Look Once

You Only Look Once atau biasa disingkat dengan YOLO adalah algoritma yang dapat mendeteksi objek secara *real-time*. Salah satu model dari YOLO adalah YOLOv5. YOLOv5 mempunyai empat versi yaitu *small* (s), *medium* (m), *large* (l), dan *extra large* (x) dimana setiap varian membutuhkan waktu yang berbeda untuk melakukan *training* dan tingkat akurasi yang berbeda. Semakin tinggi tingkat versi, maka YOLOv5 akan memberikan akurasi yang semakin tinggi. Pada penelitian ini, YOLOv5 yang digunakan adalah YOLOv5s. YOLO merupakan algoritma yang dapat mendeteksi objek secara cepat, tetapi kurang baik dalam pendeteksian untuk objek kecil. Sehingga, dengan adanya model YOLOv5 dikembangkan untuk menutupi kekurangan YOLO yang ada. Pada YOLOv5 terdapat tiga jenis peningkatan data yaitu *scaling*, *color space adjustment*, dan *mosaic enhancement*. Kemudian, digunakan R-CNN untuk memperkuat pendeteksian pada objek kecil dan kemampuan tinggi dalam pendeteksian ukuran gambar yang berbeda. Selain itu, untuk menambahkan keakuratan prediksi objek jauh secara cepat pada YOLOv5 dirancang dengan penambahan R-FCN.

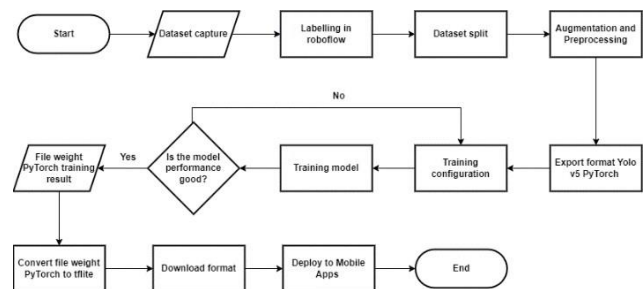
E. Parameter Performa

Parameter performa adalah parameter yang digunakan untuk mengoptimalkan model. Hasil ini didapatkan dari hasil *training* yang dilakukan. Beberapa parameter performa yang

digunakan dalam penelitian ini adalah *precision*, *recall*, *F1 score*, dan *mAP@0.5IoU*.

III. METODE

A. Gambaran Sistem



Gambar 3. 1 Gambaran sistem

Gambar 3.1 merupakan sub-sistem proses *deep learning* dari pengambilan data sampai model di *deploy* ke *mobile apps*. Langkah-langkah tersebut diperlukan untuk membangun dan menggunakan sistem *deep learning* untuk deteksi objek menggunakan model YOLOv5, yang nantinya akan diimplementasikan dalam aplikasi Android. Proses dimulai dengan mengidentifikasi kebutuhan sistem dan persiapan awal sebelum melakukan pelabelan dan pelatihan model. Hal ini mencakup perencanaan dataset, *tools* yang akan digunakan, dan spesifikasi model yang ingin dihasilkan. Data yang diperlukan untuk melatih model diambil dari webcam laptop dan diambil dari penelitian milik Siroojuddin Apendi. Setelah dataset terkumpul, langkah selanjutnya adalah melakukan pelabelan, dengan membuat *react box* pada objek setiap gambar dengan anotasi yang tepat. Dataset yang telah diberi label kemudian dibagi menjadi tiga bagian yaitu data pelatihan (*training*), data validasi (*validation*), dan data test (pengujian). Sebelum melatih model, dilakukan proses *preprocessing* dan *augmentation*, kemudian dataset harus diekspor ke dalam format yang sesuai dengan model YOLOv5. Selanjutnya, model YOLOv5 harus didefinisikan dan disesuaikan dengan memilih hyperparameter yang tepat, seperti *learning rate*, *batch size*, dan lainnya. Model ini kemudian dilatih menggunakan dataset yang telah diproses dan dibagi sebelumnya. Setelah pelatihan selesai, model akan menghasilkan parameter bobot yang disimpan dalam bentuk file PyTorch. File bobot ini merepresentasikan hasil pelatihan model dan akan digunakan untuk evaluasi lebih lanjut serta akan dikonversi ke format yang sesuai untuk implementasi pada aplikasi Android, sehingga file bobot PyTorch yang dihasilkan harus dikonversi ke format TFLite. Setelah file model TFLite dibuat, model tersebut harus didownload ke sistem untuk nantinya diintegrasikan ke dalam aplikasi Android. Implementasi ke aplikasi Android dilakukan oleh rekan saya Luthfi Yafi Alfiansyah.

B. Analisis Kebutuhan Sistem

Berdasarkan dari gambaran sistem, maka dibutuhkan beberapa aspek yang digunakan untuk melakukan implementasi agar sistem dapat berjalan. Berikut ini aspek yang dibutuhkan:

1. Analisis dataset yang digunakan

Pada setiap model klasifikasi gambar menggunakan *deep learning* dibutuhkan dataset yang digunakan untuk melatih model untuk mengenali objek. Dataset yang digunakan pada penelitian ini berjumlah 26 *class* yang terdiri dari abjad a – z. Dataset yang dibutuhkan untuk membuat sistem ini setiap kelasnya akan membutuhkan 40 frame, dengan kombinasi beberapa data milik pribadi dan diambil dari penelitian milik Siroojuddin Apendi. Kelas berjumlah 26 sehingga total dataset keseluruhan adalah 1.040 gambar. Pengambilan dataset secara pribadi dirancang dengan menggunakan OpenCV dengan latar belakang berwarna putih dengan tujuan untuk mendapatkan akurasi yang tinggi, karena memberikan kontras yang jelas. Berikut ini, contoh dari dataset yang digunakan:



GAMBAR 3. 2
Contoh dataset huruf H dan C

Sebelum dataset diolah lebih lanjut, maka perlu dilakukan pelabelan pada dataset. Pelabelan pada penelitian ini menggunakan *platform* roboflow. Dataset akan dibagi menjadi tiga bagian yaitu *training*, *validation*, dan *test*. Data *training* adalah data yang digunakan untuk melatih model dan mengajarkan model untuk mengenali pola dan menyesuaikan parameter agar dapat melakukan tugas yang diinginkan serta untuk mendeteksi kelas yang belum pernah dilihat sebelumnya. Data *training* memiliki rasio yang lebih tinggi daripada *validation* dan *test*, hal tersebut dikarenakan data *training* sebagai penyedia data, sehingga semakin banyak data *training*, model akan mempunyai peluang untuk mengeksplorasi variasi dataset dan mengenali pola yang lebih kompleks. Data *validation* digunakan untuk mengukur performa model selama proses training. Kemudian, data *test* digunakan sebagai hasil evaluasi untuk menguji performa model pada data yang belum pernah dilihat.

Sebelum data diolah pada model, data akan diolah terlebih dahulu dengan melakukan *preprocessing* dan *augmentation*. Hal tersebut perlu dilakukan untuk mengatasi ketidakseimbangan dataset, pencegahan *overfitting* dan peningkatan generalisasi sehingga dapat membantu mengenali objek dengan skala, kondisi, atau pose yang berbeda. Proses *preprocessing* akan diimplementasikan ke seluruh gambar, sedangkan untuk proses *augmentation* hanya akan diimplementasikan ke gambar *training*.

TABEL 3. 1
Preprocessing dan Augmentation

<i>Preprocessing</i>	<i>Augmentation</i>
<i>Auto-orient</i>	<i>Flip : Horizontal</i>
<i>Resize to 640x640</i>	<i>Crop minimum zoom : 0%, maximum zoom : 20%</i>
	<i>Rotation between -5° and +5°</i>
	<i>Shear ±5 horizontal, ±5 vertical</i>
	<i>Grayscale on 15% of images</i>
	<i>Saturation between -25% and +25%</i>

2. Analisis perangkat keras yang digunakan

Pada penelitian tugas akhir ini perangkat keras yang digunakan adalah laptop. Laptop digunakan untuk melakukan pelabelan sampai dengan pelatihan model. Berikut ini adalah detail spesifikasi laptop yang digunakan :

TABEL 3. 2
Perangkat keras yang digunakan

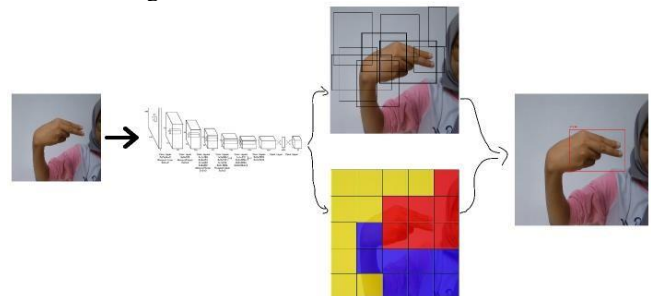
Type	Hardware/Tools	Spesifikasi
Laptop	Acer Swift	Inter® Core™ i5-1035G1 RAM 8.00 GB OS Windows 11 Home Single Language CPU @ 1.00GHz 1.19 GHz GPU NVIDIA GeForce MX250
Cloud Server	Google Colaboratory	GPU Tesla V100-SXM2-16GB, CUDA 11.1 (cu118)

3. Analisis perangkat lunak yang digunakan

Pada penelitian ini digunakan juga perangkat lunak sebagai penunjang untuk pengimplementasian sistem yang dibuat. Berikut ini spesifikasi perangkat lunak yang digunakan :

- Bahasa pemrograman Python versi 3.8.
- Sistem Operasi Windows 11 Home Single Language.
- Pengolahan dataset Roboflow.
- Anaconda Visual Studio sebagai uji coba secara *real time*.
- Anaconda Jupyter Notebook untuk pengambilan dataset.

C. Proses Algoritma YOLO



GAMBAR 3. 3
Cara kerja YOLO

Sebelum di proses oleh algoritma YOLOv5, gambar akan diubah ukurannya menjadi 640x640 dan kemudian akan dibagi menjadi sebuah kisi-kiri (grid), 1 grid terdapat 1 metrik. Dari setiap grid akan dilakukan proses konvolusi-konvolusi untuk mendapatkan garis-garis prediksi yang digunakan YOLOv5 untuk mendeteksi objek yang terdapat didalam grid. Kemudian, akan dilakukan penentuan untuk *bounding box* dengan metrik pada setiap grid seperti dibawah ini:

$$y = \begin{bmatrix} p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c \end{bmatrix} \quad (3.1)$$

Rumus 3.1 merupakan metrik yang terdapat pada deteksi YOLOv5. p_c adalah jumlah objek yang berada didalam kolom (pada grid), b_x dan b_y menunjukan posisi sumbu x dan y dari kotak pembatas yang mengelilingi objek, b_h dan b_w

adalah tinggi dan lebar dari kotak pembatas yang mengelilingi objek yang terdeteksi, dan c menunjukkan total kelas objek yang diidentifikasi oleh model. Pada penelitian ini, jumlah kelas sebanyak 26 kelas, sehingga nilai c pada metrik terdapat 26 yaitu dari C1 sampai C26. Setelah mendapatkan *bounding box* disetiap grid, maka dilakukan proses :

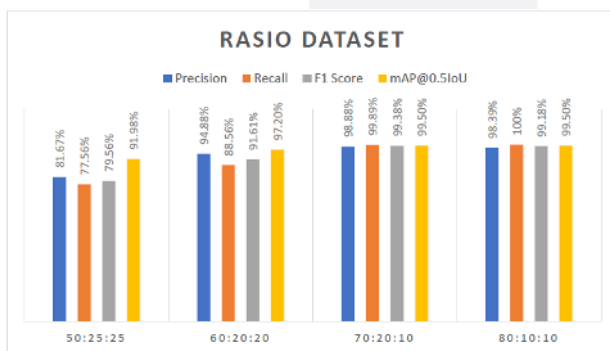
- Menghilangkan prediksi dengan probabilitas rendah dan menjadikan 1 *bounding box*.
- Kemudian, menggunakan IoU dan NMS untuk menghasilkan prediksi akhir.

Confidence score direpresentasikan IoU, dimana IoU berfungsi untuk mengevaluasi area yang beririsan antara dua *bounding box*, yaitu *bounding box* hasil prediksi dan *bounding box ground truth* (kebenaran). Semakin beririsan atau semakin dekat jarak antara *bounding box* prediksi dan *bounding box ground truth*, maka dapat digunakan untuk menilai skor IoU yang diperoleh baik atau buruk [8]. Hasil dari *precision*, *recall*, dan lainnya merupakan hasil dari menerapkan IoU. Sedangkan, NMS adalah teknik yang digunakan dalam pendeteksi objek yang bertujuan untuk memilih nilai *confidence score* terbaik [9].

IV. HASIL DAN PEMBAHASAN

A. Hasil Pengujian Partisi Dataset

Dataset yang telah disimpan akan dilakukan proses *training* dengan tujuan untuk mencari model terbaik dari skenario yang diberikan. Metode yang digunakan adalah *train-validation-test split*. Jumlah keseluruhan dataset adalah 1040, dengan jumlah kelas sebanyak 26. Dataset displit menjadi data *train*, *validation*, dan *test*. Terdapat empat skenario pengujian dataset yaitu rasio 50:25:25, 60:20:20, 70:20:20, 80:10:10. Hasil *training* dari empat skenario pengujian dataset, dapat dilihat pada gambar 4.1.



GAMBAR 4. 1

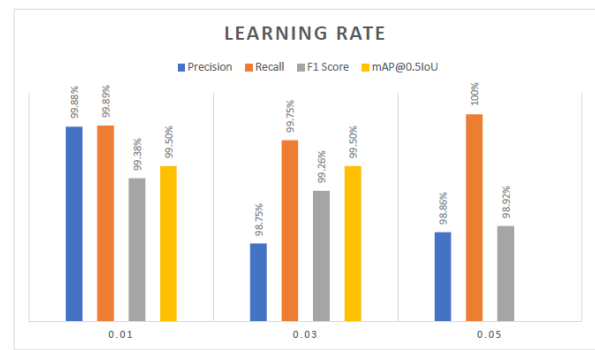
Grafik hasil *training* skenario rasio dataset

Dari 4 pengujian rasio *dataset*, model terbaik dengan rasio 70:20:10. Rasio 80:10:10 memiliki hasil metrik evaluasi yang sama pada mAP dengan rasio 70:20:10, tetapi pada rasio 80:10:10 model terlalu lama dalam proses deteksi pada saat dilakukan percobaan secara *real-time*, dibandingkan dengan dataset rasio 70:20:10. Selain itu, model sulit untuk mendeteksi gerakan yang dilakukan secara *real-time*.

B. Hasil Pengujian Learning Rate

Setelah, melakukan proses pengujian dataset, langkah selanjutnya adalah melakukan pengujian *learning rate*. Terdapat tiga skenario pengujian *learning rate* dengan nilai

0.01, 0.03, dan 0.05. Rasio dataset yang digunakan dalam pengujian ini yaitu rasio 70:20:10 yang merupakan hasil paling optimal dari rasio dataset lainnya.



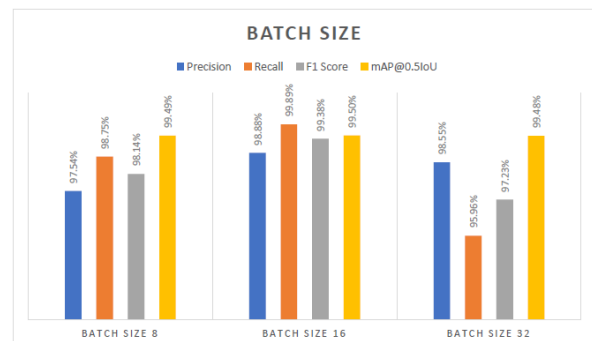
GAMBAR 4. 2

Grafik hasil *training* skenario *learning rate*

Dari hasil evaluasi yang dilakukan *learning rate* 0.01 merupakan hasil yang paling optimal untuk model. Dari hal tersebut dapat disimpulkan bahwa *learning rate* paling kecil pada model yang diujikan lebih optimal dibandingkan dengan *learning rate* yang lebih besar.

C. Hasil Pengujian Batch Size

Setelah pengujian *learning rate*, selanjutnya adalah melakukan pengujian *batch size*. Rasio dataset 70:20:10 dan *learning rate* 0.01 yang telah optimal sebelumnya akan digunakan pada pengujian ini. Pengujian dilakukan sebanyak tiga skenario yaitu dengan nilai 8, 16, dan 32.



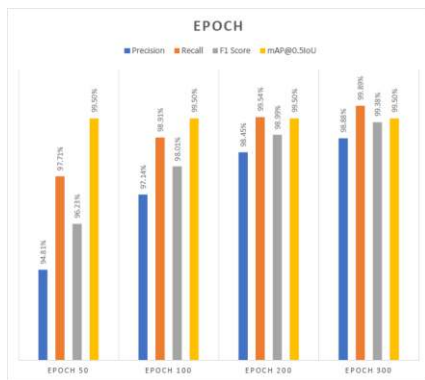
GAMBAR 4. 3

Grafik hasil *training* skenario *batch size*

Dari hasil pengujian yang didapatkan dengan tiga skenario *batch size*, metrik evaluasi yang paling optimal adalah *batch size* 16. Hasil mAP@0.5IoU yang dihasilkan tidak jauh berbeda dalam tiga skenario tersebut. Tetapi, setelah dilakukan pengujian secara *real-time batch size* 8 dan *batch size* 32 sulit untuk mengenali gambar yang berbeda dari data pelatihan, hal tersebut dikarenakan model mengalami kesulitan dalam melakukan generalisasi data yang belum pernah dilihat sebelumnya karena pada *batch size* 8 iterasi yang dilakukan terlalu detail sedangkan pada *batch size* 32 kurangnya variasi setiap iterasi dalam pengamatan yang dilakukan.

D. Hasil Pengujian Epoch

Setelah mendapatkan hasil yang optimal dari pengujian rasio *dataset*, *learning rate* dan *batch size*, selanjutnya dilakukan pengujian menggunakan *epoch*. Terdapat 4 skenario pengujian *epoch* yang dilakukan yaitu *epoch* 50, *epoch* 100, *epoch* 200, dan *epoch* 300.



GAMBAR 4. 4

Grafik hasil *training* skenario *epoch*

Dari hasil pengujian yang dilakukan, *epoch* 300 adalah skenario paling optimal dibandingkan dengan skenario lainnya. Hasil mAP@0.5IoU yang dihasilkan untuk seluruh pengujian memiliki nilai yang sama. Tetapi, pada saat melakukan pengujian dengan menggunakan dataset berupa gambar, untuk skenario *epoch* yang lainnya, terdapat gambar yang tidak terdeteksi dan gambar tidak terdeteksi dengan benar.

E. Hasil Pengujian *resize* gambar

Setelah mendapatkan hasil yang optimal dari pengujian rasio *dataset*, *learnin* rate, *batch size*, dan *epoch*, selanjutnya dilakukan pengujian menggunakan *resize* gambar. Skenario yang dilakukan ada dua yaitu menggunakan *resize* 320 dan *resize* 640.



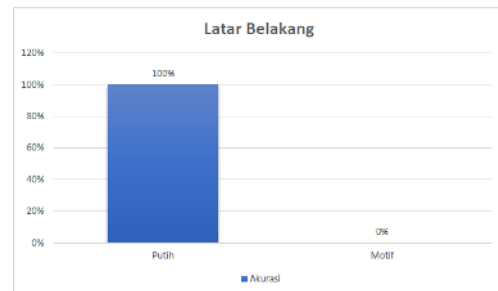
GAMBAR 4. 5

Grafik hasil *training* skenario *resize* gambar

Dari hasil pengujian skenario yang dilakukan, *resize* gambar 640 yang paling optimal dari *resize* 320. Pada saat melakukan pengujian secara *real time*, pada *resize* 320 beberapa huruf tidak dapat terdeteksi dengan benar. Sehingga, model yang paling optimal adalah menggunakan rasio *dataset* 70:20:10, *learning rate* 0.01, *batch size* 16, *epoch* 300, dan *resize* gambar 640.

F. Hasil Pengujian Latar Belakang

Pengujian inferensi latar belakang dilakukan untuk mengukur kemampuan sistem dalam mengenali dan mengklasifikasi objek berdasarkan latar belakangnya. Pengujian ini menggunakan dua skenario yang berbeda untuk memahami sejauh mana sistem dapat beroperasi dengan baik dalam kondisi yang berbeda. Nilai akurasi yang didapatkan pada pengujian ini berasal dari perhitungan banyaknya kelas yang terdeteksi pada masing-masing jarak, dengan nilai aktual tiga pada setiap kelas. Rumus yang digunakan dalam perhitungan menggunakan persamaan akurasi.



GAMBAR 4. 6

Grafik hasil pengujian latar belakang

Dalam skenario ini, pengujian latar belakang dilakukan dengan menggunakan latar belakang berwarna putih dan bermotif. Hasil pengujian menunjukkan bahwa model yang diuji mendapatkan akurasi 100% dalam mengenali isyarat dengan latar belakang putih. Namun, akurasi untuk mengenali isyarat pada latar belakang bermotif mendapatkan akurasi 0%, yang berarti model tidak mampu mengidentifikasi latar belakang bermotif. Hal tersebut disebabkan pada saat melakukan proses pengambilan dataset, latar belakang yang digunakan seluruhnya berwarna putih. Sehingga, dalam melakukan proses *training* model hanya mengenali warna putih.

G. Hasil Pengujian Jarak

Pengujian inferensi jarak adalah suatu proses evaluasi yang bertujuan untuk mengetahui jarak optimal yang paling tepat untuk sebuah model. Pengujian inferensi jarak dilakukan dengan empat skenario yaitu jarak 30 cm, 50 cm, 75 cm, dan jarak 100 cm.



GAMBAR 4. 7

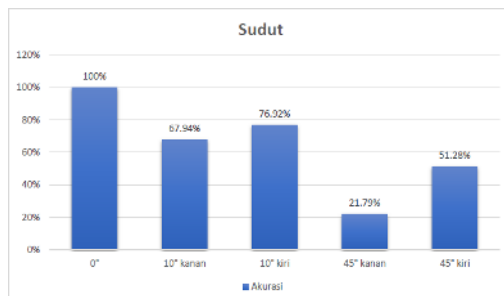
Grafik hasil pengujian jarak

Berdasarkan percobaan yang telah dilakukan dengan menggunakan empat skenario berbeda pada jarak 30 cm, 50 cm, 75 cm, dan 100 cm, didapatkan hasil paling optimal dengan jarak 30 cm. Pada jarak 30 cm, hasil akurasi yang didapatkan mencapai 100%. Terjadinya penurunan akurasi pada hasil pengujian jarak dikarenakan pada jarak yang lebih jauh, resolusi spasial atau kemampuan sistem untuk membedakan detail-detail kecil dalam gerakan isyarat menurun. Bahasa isyarat memiliki gerakan dan bentuk tangan yang kompleks, dan semakin jauh jaraknya, semakin kecil ukuran yang terlihat oleh kamera. Sehingga, sistem kesulitan untuk mengidentifikasi detail-detail penting dalam isyarat, mengakibatkan penurunan akurasi.

H. Hasil Pengujian Sudut Kamera

Pengujian sudut kamera dilakukan untuk mengetahui sudut pandang paling optimal kamera dalam melakukan pendeteksian pada isyarat. Sehingga, dapat membantu untuk mengidentifikasi area pandang yang akan menyebabkan

kesalahan dalam melakukan deteksi isyarat. Pengujian ini dilakukan dengan empat skenario yaitu pada sudut 0°, 10° kanan, 10° kiri, 45° kanan, dan 45° kiri.



GAMBAR 4. 8
Grafik hasil pengujian sudut kamera

Dari hasil pengujian yang telah dilakukan dengan menggunakan lima skenario sudut yang berbeda yaitu sudut 0°, sudut 10° kanan, sudut 10° kiri, sudut 45° kanan, dan sudut 45° kiri, hasil paling optimal terdapat pada pengujian dengan sudut 0°. Skenario dengan sudut 0° mendapatkan akurasi mencapai 100%.

I. Hasil Pengujian Inferensi Cahaya dengan *Lux Light Meter Pro*

Pengujian inferensi cahaya ini dirancang untuk mengevaluasi kinerja sistem dalam mengenali atau mendeteksi cahaya dari berbagai lingkungan. Terdapat empat skenario yang dirancang secara khusus untuk mencakup berbagai kondisi dan situasi yang mungkin terjadi dalam pengenalan cahaya.



GAMBAR 4. 9
Grafik hasil pengujian tingkat cahaya

Pengujian cahaya yang dilakukan mencakup empat skenario yaitu dengan cahaya 170 lux, 50 lux, 10 lux, dan 5 lux. Akurasi mencapai 100% pada pengujian dengan menggunakan cahaya 170 lux dan 50 lux.

V. KESIMPULAN

Berdasarkan hasil penelitian yang dilakukan pada algoritma YOLOv5, pada penelitian ini jumlah kelas sebanyak 26 kelas (abjad a-z), model terbaik terdapat pada konfigurasi parameter dengan rasio dataset 70% *train*, 20% *validation*, dan 10% *test*. Kemudian, *learning rate* 0.01, *batch size* 16, *epoch* 300, dan *resize* gambar 640. Dari model terbaik didapatkan nilai mAP@0.5IoU sebesar 99.5%.

Pengujian model YOLOv5 untuk deteksi gestur SIBI statis secara *real-time* skenario terbaik terdapat pada latar belakang putih, jarak 30 cm, sudut 0°, tingkat cahaya 50 lux

sampai 170 lux. Setiap skenario yang diujikan mendapatkan akurasi mencapai 100%.

REFERENSI

- [1] M. Sari and A. Taher, "Perkembangan Sosial dan Kepribadian Pada Anak Tunarungu (Studi Penelitian di SDLB Kebayakan Takengon, Aceh Tengah)," *Jurnal Ilmiah Mahasiswa Fakultas Ilmu Sosial & Ilmu Politik*, vol. 1, no. 1, hal. 6-8, 2017.
- [2] B. Kabupaten, "Banyaknya Desa Menurut Keberadaan Penyandang Cacat di Kabupaten/Kota Provinsi Jawa Barat (Bukan Tepi Laut)," Badan Pusat Statistik Kabupaten Bekasi, 2021. [Online]. Tersedia: <https://bekasikab.bps.go.id/statictable/2021/06/18/1717/banyaknya-desa-menurut-keberadaan-penyandang-cacat-di-kabupaten-kota-provinsi-jawa-barat-bukan-tepi-laut-2011.html>. [Diakses: 07-Nov-2022].
- [3] P. R. Anggriana, "Stop AUDISME, yes Bahasa isyarat," *The Columnist*, 23-Sep-2022. [Online]. Tersedia: <https://thecolumnist.id/artikel/stop-audisme-yes-bahasa-isyarat-2260>. [Diakses: 07-Nov-2022].
- [4] J. K. Hikmalansya, "Aplikasi Pembelajaran Bahasa Isyarat berbasis android," *Inform : Jurnal Ilmiah Bidang Teknologi Informasi dan Komunikasi*, vol. 1, no. 2, 2016.
- [5] Nugraheni, A.S., Husain, A.P. and Unayah, H. (2023) 'Optimalisasi Penggunaan Bahasa ISYARAT Dengan sibi Dan Bisindo Pada Mahasiswa difabel tunarungu di Prodi Pgmi Uin Sunan Kalijaga', *Jurnal Holistika*, 5(1), p. 28. doi:10.24853/holistika.5.1.28-33.
- [6] K. Perempuan, "Siaran pers," Komnas Perempuan | Komisi Nasional Anti Kekerasan Terhadap Perempuan, 23-Sep-2021. [Online]. Tersedia: [https://komnasperempuan.go.id/siaran-pers-detail/siaran-pers-komnas-perempuan-tentang-peringatan-hari-bahasa-isyarat-internasional-jakarta-23-september-2021#:~:text=Penerapan%20SIBI%20ini%20diresmikan%20dalam,Sekolah%20Luar%20Biasa%20\(SLB\)](https://komnasperempuan.go.id/siaran-pers-detail/siaran-pers-komnas-perempuan-tentang-peringatan-hari-bahasa-isyarat-internasional-jakarta-23-september-2021#:~:text=Penerapan%20SIBI%20ini%20diresmikan%20dalam,Sekolah%20Luar%20Biasa%20(SLB)). [Diakses: 08-Nov-2022].
- [7] J. J. Grefenstette, "Genetic algorithms and machine learning," *Machine Learning*, vol. 3, no. 2-3, hal. 95–99, 198.
- [8] Salim, A. (2020) Intersection over Union, Medium. Available at: <https://medium.com/bisa-ai/intersection-over-union-a8d1532899b3> (Accessed: 01 June 2023).
- [9] S, G., Duraimurugan, N. and Chokkalingam, S.P. (2019) Real-Time Object Detection with Yolo, 8(3S), pp. 578–581.