

# APLIKASI PENERJEMAH BAHASA ISYARAT BISINDO MENGGUNAKAN METODE YOLOv5 BERBASIS MOBILE

Anggi Vivi Febrina Silalahi<sup>1</sup>, Agustina Setianing Budi<sup>2</sup>, Januaray Satya Ega<sup>3</sup>, I Wayan Yoga Adi Pranata<sup>4</sup>, Made Sudarma<sup>5</sup>, Duman Care Khrisne<sup>6</sup>, I Made Arsa Suyadnya<sup>7</sup>

<sup>1, 2, 3, 4</sup>Mahasiswa Program Studi Teknik Elektro, Fakultas Teknik, Universitas Udayana

<sup>5, 6, 7</sup>Dosen Program Studi Teknik Elektro, Fakultas Teknik, Universitas Udayana

Kampus Bukit, Jl Raya Kampus Unud Jimbaran, Kuta Selatan, Badung, Bali 80631

anggivivi9@gmail.com<sup>1</sup>, setianingbudi@student.unud.ac.id<sup>2</sup>,

januaraysatya@student.unud.ac.id<sup>3</sup>, yogaadipranata10@gmail.com<sup>4</sup>, sudarmaee@unud.ac.id<sup>5</sup>,

duman@unud.ac.id<sup>6</sup>, arsa.suyadnya@unud.ac.id<sup>7</sup>

## ABSTRAK

Penelitian ini bertujuan untuk mengembangkan aplikasi mobile yang mampu mengenali dan menerjemahkan isyarat BISINDO secara *real-time* menggunakan metode YOLOv5. YOLOv5 (*You Only Look Once* versi 5) adalah algoritma deteksi objek yang efisien dan cepat, sehingga cocok untuk diimplementasikan pada perangkat *mobile*. Dalam pengembangan aplikasi ini, kami mengumpulkan dan melabeli dataset isyarat BISINDO, kemudian melatih model YOLOv5 untuk mendeteksi dan mengenali isyarat tersebut. Hasil uji menunjukkan bahwa model yang diimplementasikan memiliki akurasi yang tinggi dalam mengenali berbagai isyarat BISINDO. Selain itu, aplikasi ini juga dirancang dengan antarmuka pengguna yang intuitif dan mudah digunakan, sehingga pengguna dapat dengan cepat dan mudah berkomunikasi dengan orang yang menggunakan bahasa isyarat. Dengan aplikasi ini, diharapkan dapat memperluas akses komunikasi bagi penyandang tunarungu dan meningkatkan inklusi sosial mereka dalam masyarakat. Penelitian lebih lanjut diperlukan untuk terus menyempurnakan akurasi dan efisiensi aplikasi serta menambahkan lebih banyak fitur yang mendukung berbagai kebutuhan pengguna.

**Kata kunci:** BISINDO, Deteksi Objek, YOLOv5, Aplikasi Mobile, Roboflow, *User Centered Design*, ReactJS, CapacitorJS

## ABSTRACT

*This research aims to develop a mobile application that is able to recognize and translate BISINDO in real-time using the YOLOv5 method. YOLOv5 (You Only Look Once version 5) is an efficient and fast object detection algorithm, making it suitable for implementation on mobile devices. In developing this application, we collected and labeled a typical BISINDO dataset, then trained the YOLOv5 model to detect and recognize these characteristics. The test results show that the implemented model has high accuracy in recognizing various BISINDO characteristics. Apart from that, this application is also designed with an intuitive and easy-to-use user interface, so that users can quickly and easily communicate with people using sign language. With this application, it is hoped that it can expand communication access for people with hearing impaired and increase their social inclusion in society. Further research is needed to continue improving the app's accuracy and efficiency and adding more features that support different user needs.*

**Key Words:** BISINDO, Object Detection, YOLOv5, Mobile Application, Roboflow, User Centered Design, ReactJS, CapacitorJS

## 1. PENDAHULUAN

Bahasa isyarat adalah bahasa yang menggunakan bahasa tubuh, gerak bibir dan komunikasi manual. Keberadaan bahasa isyarat dapat membantu komunikasi antara penderita tunarungu dengan sesama penderita tunarungu atau penderita tunarungu dengan orang normal atau sebaliknya [1]. Bahasa isyarat juga merupakan jenis bahasa yang tidak bergantung pada tulisan atau suara manusia [2]. Bahasa isyarat sangat penting bagi mereka yang memiliki keterbatasan dan terhambat dalam mendengar, sehingga untuk berkomunikasi dengan penyandang tunarungu adalah dengan menggunakan bahasa isyarat.

Dalam hal ini, bahasa isyarat, khususnya BISINDO berfungsi sebagai alat penting bagi penyandang tunarungu untuk berkomunikasi dengan lingkungan di sekitar mereka. Namun, penggunaan BISINDO di Indonesia masih menjadi hal yang asing bagi masyarakat umum sehingga sering terjadi kesalahpahaman saat penyandang tunarungu berkomunikasi dengan masyarakat umum [3].

Untuk mengatasi permasalahan kurangnya pemahaman masyarakat umum terhadap BISINDO, diusulkan solusi berupa sebuah aplikasi penerjemah BISINDO berbasis mobile yang dapat diakses melalui *smartphone* Android. Nantinya, setiap gerakan BISINDO dari penyandang tunarungu akan dideteksi melalui kamera secara *real-time* atau dengan cara memasukkan gambar ke dalam aplikasi, kemudian model akan menerjemahkan gerakan tersebut menjadi teks. Selain itu, pada aplikasi ini akan terdapat fitur kamus BISINDO yang bisa digunakan untuk mempelajari gerakan BISINDO untuk abjad, angka, dan kata.

Penelitian ini akan menggunakan dataset berupa gambar untuk mengetahui bagaimana gerakan dari beberapa kata pada BISINDO. Dataset terdiri dari 15 kata yang memiliki gerakan khusus pada BISINDO dan sering digunakan untuk berkomunikasi sehari-hari. Kata-kata yang

digunakan, yaitu ambil, apa kabar, di mana, di sana, di sini, iya, jam, kamu, nama, rumah, sama, saya, siapa, tidak, dan tolong. Kata yang digunakan ini menandakan variasi gestur dalam komunikasi bahasa isyarat. Dengan penggunaan kata yang mencakup berbagai gestur, sistem dapat belajar untuk mengenali dan menerjemahkan beragam isyarat yang digunakan dalam komunikasi sehari-hari [4]. Setiap kata terdiri dari 200 gambar, sehingga jumlah gambar pada dataset sebanyak 3000 gambar. Jumlah gambar yang digunakan untuk melatih model tidak memiliki jumlah ideal yang pasti, tetapi semakin banyak gambar yang digunakan, maka semakin baik juga performa model yang dilatih [5]. Nantinya, dataset yang telah dibuat akan digunakan untuk melatih *deep learning* dengan model YOLOv5 untuk mendeteksi dan mengenali gerakan BISINDO. Kemudian, perancangan antarmuka aplikasi pada penelitian ini menggunakan metode *User Centered Design* agar aplikasi mudah digunakan dan dimengerti oleh pengguna karena metode ini fokus pada kebutuhan pengguna [6]. Selanjutnya, model YOLOv5 dan desain antarmuka aplikasi yang telah dibuat diterapkan ke dalam perancangan aplikasi penerjemah BISINDO berbasis mobile menggunakan metode *Prototype*.

Dengan demikian, penelitian ini akan menghasilkan sebuah aplikasi yang bisa menerjemahkan BISINDO dengan baik, serta memiliki tampilan antarmuka yang interaktif dan responsif bagi pengguna. Selain itu, aplikasi ini akan membantu komunikasi antara penyandang tunarungu dengan masyarakat umum dan menarik minat masyarakat umum untuk mempelajari BISINDO.

## 2. KAJIAN PUSTAKA

### 2.1 Dataset

Dataset merupakan sebuah basis data yang disimpan dalam memori yang memiliki seluruh karakteristik, fitur, dan fungsi pada basis data konvensional. Di dalam sebuah dataset dapat ditemukan banyak tabel yang memiliki relasi antara satu sama lain, termasuk *foreign key* dan

integritas referensial. Dataset digunakan untuk merepresentasikan data dan hubungannya di dalam memori. Dataset memiliki struktur mirip dengan data yang terdapat di dalam basis data dan berisikan sejumlah tabel data dan *entry* data [7].

## 2.2 Roboflow

Roboflow merupakan sebuah platform cloud yang menawarkan berbagai fitur penting untuk pengembangan aplikasi *computer vision*, meliputi pengumpulan data, pra-pemrosesan data, dan pelatihan model [8].

## 2.3 YOLOv5

You Only Look Once (YOLO) merupakan salah satu metode untuk melakukan pendeteksian objek berbasis *Convolutional Neural Network* yang menggunakan pendekatan jaringan syaraf tunggal (Single neural network). Jaringan ini dapat memprediksi setiap bounding box menggunakan fitur dari semua gambar sehingga dapat memprediksi bounding box dan probabilitas dalam satu evaluasi secara langsung [9].

## 2.4 UI/UX (User Interface dan User Experience)

*User Interface* merupakan elemen visual dan penting dalam membangun sebuah aplikasi yang langsung berinteraksi dengan pengguna, contohnya seperti tombol, menu, ikon, dan lain sebagainya. Sementara itu, *User Experience* merupakan bagaimana seseorang melihat dan merespons saat menggunakan suatu produk, sistem, atau layanan. *User Experience* mengukur seberapa puas dan nyaman pengguna ketika menggunakan suatu produk, sistem, atau layanan [10].

## 2.5 User Centered Design (UCD)

*User Centered Design* merupakan metode pengembangan sistem secara interaktif yang berfokus pada kegunaan dan kebutuhan pengguna. Metode ini melibatkan pengguna dari awal sampai akhir perancangan dan dilakukan secara iterasi [6].

## 2.6 Purposive Sampling

*Purposive sampling* merupakan metode pengambilan sampel secara tidak acak dan dilakukan berdasarkan pertimbangan sifat atau identitas tertentu dengan tujuan agar informasi yang diperoleh lebih representatif [11].

## 2.7 Aplikasi Mobile

Aplikasi *mobile* merupakan sebuah perangkat lunak yang dirancang untuk digunakan pada perangkat *mobile*, seperti *smartphone* atau tablet PC, di mana aplikasi tersebut bisa berjalan secara independen tanpa memerlukan perangkat tambahan. Aplikasi *mobile* memudahkan pengguna untuk mengakses informasi secara cepat dan efisien dibandingkan dengan menggunakan PC karena tidak memerlukan usaha yang lebih besar dibanding menggunakan PC [12].

## 2.8 Metode Prototype

Metode prototype merupakan metode pengembangan yang cocok digunakan untuk aplikasi yang mengadopsi model *deep learning* karena dapat melakukan uji model sesuai dengan kebutuhan dan melakukan evaluasi dengan cara mengganti model dengan model lainnya [13].

## 2.9 Validasi Ahli

Validasi ahli merupakan proses mendapatkan umpan balik dan evaluasi dari para ahli di bidang tertentu untuk menilai kualitas, akurasi, dan relevansi suatu tes atau instrumen Validasi ahli bertujuan untuk memastikan kualitas dan validitas suatu instrumen, dalam konteks penelitian ini validasi ahli memastikan kualitas dan validitas dari dataset yang akan digunakan dalam melatih YOLOv5 [14].

## 2.10 mAP50 (Mean Average Precision at 50)

mAP memberikan gambaran umum bagaimana kinerja dari sebuah deteksi objek dengan cara menghitung nilai rata-rata dari nilai *Average Precision* untuk semua kelas objek [9]. Untuk menghitung mAP, dibutuhkan suatu kelas tertentu ( $k$ ) dalam menghitung AP (*Average Precision*) yang sesuai seperti pada persamaan 1.

$$mAP = \frac{1}{n} \sum_{k=1}^{k=n} AP_k \quad (1)$$

$n$  menunjukkan jumlah kelas dan  $AP_k$  menunjukkan rata-rata *precision* kelas  $k$ .

Pada penelitian ini menggunakan mAP50, di mana mAP50 ini merupakan nilai rata-rata *precision* yang diukur pada tingkat batas *Intersection Over Union* (IoU) sebesar 0.50. IoU adalah ukuran yang mengukur tumpang tindih antara kotak pembatas yang diprediksi dan kotak

pembatas kebenaran dasar. Ini memainkan peran mendasar dalam mengevaluasi keakuratan lokalisasi objek. Parameter ini memberikan gambaran tentang akurasi model dengan mempertimbangkan kemudahan dalam mendeteksi objek dengan fokus pada IoU sekitar 0.50 [15].

(a) *Confusion Matrix*

*Confusion matrix* merupakan ringkasan hasil prediksi dari keseluruhan klasifikasi. Jumlah klasifikasi yang benar dan yang salah dikumpulkan dengan nilai hitung, kemudian dipecah oleh setiap kelas sehingga matriks ini akan memberikan informasi yang dibuat oleh *classifier* dan jenis kesalahannya. Pengelompokan pada *confussion matrix* dibagi menjadi empat, yaitu aktual bernilai positif dan diprediksi positif (*True Positive*), aktual bernilai negatif dan diprediksi negatif (*True Negative*), aktual bernilai negatif dan diprediksi positif (*False Positive*), serta aktual bernilai negatif dan diprediksi negatif (*False Negative*) [9]. Pengelompokan ini dapat dilihat pada gambar 1.

		ACTUAL	
		Class 1 (Positive)	Class 2 (Negative)
PREDICTION	Class 1 (Positive)	TP	FP
	Class 2 (Negative)	FN	TN

Gambar 1. *Confusion Matrix*

(b) *Precision*

*Precision* merupakan salah satu matriks evaluasi yang digunakan untuk memprediksi seberapa sering model tersebut memunculkan persentase prediksi nilai benar. *Precision* akan memberikan informasi tentang seberapa akurat model dalam mengklasifikasi *instance* positif. Jika nilai *precision* tinggi maka model cenderung membuat sedikit kesalahan dengan mengidentifikasi *instance* positif. Sebaliknya, jika nilai *precision* rendah, maka model lebih rentan membuat kesalahan dalam mengklasifikasi *instance* sebagai nilai positif [15]. Adapun rumus yang digunakan untuk menghitung *precision* dapat dilihat pada persamaan 2.

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (2)$$

(c) *Recall*

*Recall* merupakan matriks evaluasi yang digunakan untuk mengukur sejauh mana model klasifikasi berhasil mengidentifikasi semua *instance* positif yang sebenarnya. *Recall* memberikan informasi tentang kemampuan model untuk mengingat atau mendeteksi *instance* positif yang ada [15]. Adapun rumus yang digunakan untuk menghitung *recall* dapat dilihat pada persamaan 3.

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (3)$$

(d) *F1-Score*

*F1-Score* dilakukan dengan menghitung nilai dari *False Positive* (FP) dan *False Negative* (FN). *F1-Score* memberikan bobot lebih tinggi pada nilai FN dan FP, sementara nilai *True Negative* (TN) tidak mempengaruhi skor. Nilai terbaik dari *F1-Score* adalah 1.0 dan nilai terburuknya adalah 0. Jika *F1-Score* memiliki nilai yang baik, maka dapat menunjukkan bahwa metode klasifikasi dan deteksi yang dibangun memiliki nilai *precision* dan *recall* yang baik [16]. Adapun rumus yang digunakan untuk menghitung *F1-Score* dapat dilihat pada persamaan 4.

$$F1-Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (4)$$

2.11 *System Usability Scale (SUS)*

*System Usability Scale (SUS)* merupakan alat survei yang digunakan untuk mengukur tingkat kegunaan suatu sistem, sehingga dapat menentukan apakah sistem memiliki manfaat bagi pengguna atau tidak. Metode ini menggunakan kuesioner dengan 10 butir pernyataan dan menggunakan skala *likert* 1-5 untuk pengukuran jawaban [6]. Nilai SUS diperoleh dengan menghitung skor rata-rata dari nilai responden menggunakan persamaan 5.

$$Nilai\ SUS = \frac{\sum x}{n} \quad (5)$$

Di mana  $\sum x$  merupakan jumlah skor SUS dan  $n$  merupakan jumlah responden.

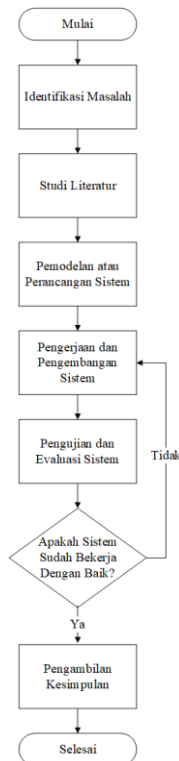
2.12 *Black Box Testing*

*Black box testing* merupakan metode pengujian fungsional dari suatu aplikasi yang telah dirancang. *Black box testing* digunakan untuk mengidentifikasi dan memperbaiki kesalahan dalam beberapa kondisi, seperti fungsi yang berjalan tidak sesuai perancangan, kesalahan kinerja

pada aplikasi, dan kesalahan pada tampilan antarmuka aplikasi [6].

### 3. METODE PENELITIAN

Penelitian ini dilaksanakan di Program Studi Teknik Elektro Fakultas Teknik Universitas Udayana, Kampus Bukit Jimbaran. Waktu pelaksanaan dimulai dari bulan September 2023 hingga April 2024. Tahapan yang dilakukan untuk melaksanakan penelitian ini dapat dilihat pada gambar 2.



Gambar 2. Diagram alir tahapan penelitian

Berikut penjelasan pada Gambar 2 :

#### Langkah 1. Identifikasi Masalah

Tahapan pertama pada penelitian ini adalah melakukan identifikasi masalah terhadap hal yang dijadikan topik penelitian dan melakukan perumusan masalah dengan mengidentifikasi permasalahan yang didapatkan. Perumusan masalah pada penelitian ini adalah bagaimana cara mengembangkan aplikasi penerjemah BISINDO berbasis mobile yang bertujuan untuk membantu komunikasi antara penyandang tunarungu dengan masyarakat umum.

#### Langkah 2. Studi Literatur

Tahapan kedua, yaitu melakukan studi literatur untuk mengumpulkan data dan referensi yang relevan dengan penelitian. Referensi didapatkan dari jurnal, buku, artikel, maupun buku digital yang berkaitan dengan permasalahan penelitian, seperti perancangan dataset, implementasi model YOLOv5, perancangan UI/UX dengan metode UCD, dan pengembangan aplikasi *mobile* menggunakan metode *prototype*.

#### Langkah 3. Pemodelan Atau Perancangan Sistem

Tahapan ketiga, yaitu melakukan pemodelan atau perancangan sistem. Pada tahap ini dilakukan pengumpulan gambar untuk dataset, perancangan model YOLOv5, perancangan *wireframe* UI/UX, membuat gambaran aplikasi menggunakan *Unified Modelling Language* (UML), dan perancangan *prototype* aplikasi menggunakan contoh model YOLOv5. Tahapan ini berguna untuk memberikan gambaran bagaimana sistem aplikasi akan dikembangkan nantinya.

#### Langkah 4. Pengerjaan dan Pengembangan Sistem

Tahapan keempat, yaitu pengerjaan dan pengembangan sistem. Dalam tahap ini dilakukan perancangan dataset, pelatihan model YOLOv5 menggunakan dataset BISINDO, merancang UI/UX berdasarkan *wireframe* menggunakan metode UCD, serta pengembangan aplikasi dengan mengintegrasikan hasil model YOLOv5 dan hasil rancangan UI/UX ke dalam aplikasi *mobile*.

#### Langkah 5. Pengujian dan Evaluasi Sistem

Tahapan kelima yaitu pengujian dan evaluasi pada sistem yang telah dibuat dengan menggunakan mAP pada model *deep learning*, metode System Usability Scale (SUS) untuk mengevaluasi UI/UX, dan metode *black box testing* dengan teknik *equivalence partitions* untuk mengevaluasi aplikasi yang telah dirancang. Jika sistem yang dibuat belum berjalan dengan baik, maka akan dilakukan evaluasi serta kembali ke tahapan pengerjaan dan pengembangan sistem.

Langkah 6. Pengambilan Kesimpulan

Tahapan keenam yaitu membuat kesimpulan dari aplikasi yang dibangun dan diambil dari hasil pengujian serta evaluasi sistem yang telah dilakukan.

## 4. HASIL DAN PEMBAHASAN

### 4.1 Aplikasi Penerjemah Bahasa Isyarat BISINDO Berbasis Mobile

Penelitian ini menghasilkan solusi untuk mengatasi permasalahan kurangnya pemahaman masyarakat umum terhadap BISINDO. Solusi tersebut berupa sebuah aplikasi penerjemah bahasa isyarat BISINDO berbasis mobile dengan nama Sekata yang beroperasi pada *smartphone* Android. Nama Sekata secara harfiah diambil dari fitur utama dari aplikasi ini yang bisa mendeteksi satu kata dari beberapa gerakan BISINDO pada satu waktu. Selain itu, Sekata memiliki arti sependapat yang mencerminkan tujuan aplikasi ini untuk menjadi jembatan komunikasi dan menyamakan pemahaman antara penyandang tunarungu dengan masyarakat umum. Aplikasi Sekata mampu mendeteksi dan mengenali beberapa kata dari gerakan khas BISINDO melalui kamera secara *real-time* atau dengan cara memasukkan gambar ke dalam aplikasi, kemudian model akan menerjemahkan gerakan tersebut menjadi teks.

### 4.2 Pengumpulan dan Pra-Pemrosesan Dataset Untuk Penerjemahan BISINDO

Topik ini membahas pengumpulan dan pra-pemrosesan dataset yang berfokus untuk mengumpulkan dan melakukan pra-pemrosesan data yang digunakan untuk melatih serta menguji model *deep learning*. Adapun tahapan yang dilakukan pada topik ini adalah sebagai berikut.

#### (a) Pengumpulan Data

Pengumpulan data dilakukan melalui pencarian referensi tentang gerakan Bahasa Isyarat Indonesia (BISINDO) dan wawancara dengan beberapa guru di SLB N 1 Badung untuk mengetahui kata-kata BISINDO yang biasa digunakan sehari-hari.

#### (b) Pelabelan Data

Proses pelabelan dilakukan menggunakan aplikasi berbasis web, yaitu *Roboflow* untuk memberikan label pada data gambar. Data gambar yang terdiri dari

15 gerakan BISINDO untuk kata diberikan label dengan kelas Ambil, Apa-kabar, Dimana, Di-sana, Di-sini, Iya, Jam, Kamu, Nama, Rumah, Sama, Saya, Siapa, Tidak, dan Tolong.

#### (c) Pembagian Dataset dan *Resize* Gambar

Dataset yang berjumlah 3000 gambar dibagi menjadi tiga bagian, yaitu data latih (*train*), data uji (*test*), dan data validasi. Pembagian dataset dilakukan dengan membagi data latih sebesar 70%, data uji sebesar 20%, dan data validasi sebesar 10% sehingga data latih berjumlah 2100 gambar, data uji berjumlah 600 gambar, dan data validasi berjumlah 300 gambar. Setelah itu, dilakukan perubahan ukuran gambar (*resize*) agar berukuran 640x640 untuk menyesuaikan proses latih data model YOLOv5 yang menggunakan ukuran 640x640.

#### (d) Hasil dan Validasi Dataset

Proses ini melibatkan konversi anotasi dataset ke dalam bentuk TXT yang dapat dibaca oleh model YOLOv5 dan dikemas menjadi file ZIP yang terdiri dari tiga bagian, yaitu data latih (*train*), data uji (*test*), dan data validasi. Dalam setiap bagian ZIP terdapat gambar-gambar sumber, file anotasi yang sesuai, dan file konfigurasi YAML untuk menginstruksikan model bagaimana cara menginterpretasikan data serta melakukan proses pelatihan.

Sebelum melakukan validasi dataset, dilakukan penyusunan dataset berisi gerakan BISINDO yang ditiru dari gambar guru SLB N 1 Badung. Setelah dilakukan pengambilan gambar, dilakukan diskusi dengan guru – guru untuk mengevaluasi ketepatan gerakan dan latar belakang gambar. Evaluasi menghasilkan latar belakang yang homogen serta tambahan sudut pengambilan gambar untuk Gerakan kata “sama” dan “siapa”. Untuk kata “saya”, ibu jari diletakkan di dada. Dataset yang telah dievaluasi divalidasi ulang dan dinyatakan sesuai dengan kaidah BISINDO yang ditandai dengan pemberian surat pernyataan oleh SLB N 1 Badung.

### 4.3 Penerapan YOLOv5 Untuk Penerjemahan BISINDO

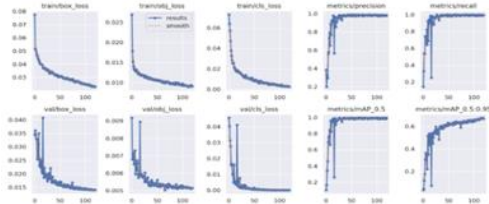
Topik ini berfokus untuk merancang dan melatih model YOLOv5 menggunakan dataset yang dirancang pada topik pertama agar bisa mendeteksi gerakan BISINDO dari beberapa kata dengan baik serta seberapa besar tingkat akurasi pada aplikasi Sekata. Adapun proses yang dilakukan pada topik ini adalah sebagai berikut.

(a) *Training Data*

*Training* dilakukan secara berulang-ulang agar mendapatkan model yang memiliki performa terbaik dalam mengenali 15 kelas yang telah dibuat, yaitu Ambil, Apa-kabar, Di-mana, Di-sana, Di-sini, Iya, Jam, Kamu, Nama, Rumah, Sama, Saya, Siapa, Tidak, dan Tolong. Jumlah *epoch* yang digunakan dalam penelitian ini sebanyak 120 *epoch* yang artinya algoritma YOLOv5 akan terus-menerus melakukan *training* dimulai dari *epoch* ke-1 sampai dengan *epoch* ke-120.

(b) Hasil *Training Data*

Adapun hasil *training data* yang berupa grafik dapat dilihat pada gambar 3.



Gambar 3. Grafik hasil *training data*

Berdasarkan gambar 3, dapat dilihat bahwa *train/box\_loss*, *train/obj\_loss*, dan *train/cls\_loss* mengalami penurunan *loss* (kerugian) seiring dengan banyaknya *training* dengan nilai *loss* yang mendekati angka 0. Begitupun dengan *val/box\_loss*, *val/obj\_loss*, dan *val/cls\_loss* yang mengalami penurunan *loss* seiring dengan banyaknya *validation* dengan nilai *loss* yang mendekati angka 0. Hal tersebut menunjukkan bahwa model yang dilatih dapat memahami dan mendeteksi sebuah objek.

Grafik *metrics/precision*, *metrics/recall*, *metrics/mAP\_0.5*, dan *mAP\_0.5:0.95* mengalami kenaikan angka seiring dengan bertambahnya jumlah *epoch* pelatihan dengan skor yang mendekati

angka 1, ini artinya memiliki kinerja yang baik dalam mendeteksi objek.

Dari hasil *training data* ini juga didapatkan nilai *precision* dan *recall* untuk setiap kelas. Adapun nilai *precision* dan *recall* untuk kelas Ambil, Apa-kabar, Di-mana, Di-sana, Di-sini, Iya, Jam, Kamu, Nama, Rumah, Sama, Saya, Siapa, Tidak, dan Tolong dapat dilihat pada tabel 1.

Tabel 1. Nilai *precision* dan *recall* setiap kelas

Kelas	Precision	Recall
Ambil	0.845	0.85
Apa-kabar	0.983	1
Di-mana	0.983	1
Di-sana	0.987	1
Di-sini	0.983	1
Iya	0.99	1
Jam	0.985	1
Kamu	0.983	1
Nama	0.985	1
Rumah	1	1
Sama	0.988	1
Saya	0.985	1
Siapa	0.988	1
Tidak	0.951	0.972
Tolong	0.984	1
Total/Rata-rata	0.975	0.988

(c) Evaluasi Kinerja Model YOLOv5

Kinerja model YOLOv5 yang telah dilatih dievaluasi menggunakan perhitungan *F1-Score*. Adapun hasil dari perhitungan *F1-Score* adalah sebagai berikut.

$$\begin{aligned}
 F1-Score &= 2 \times \frac{Precision \times Recall}{Precision + Recall} \\
 &= 2 \times \frac{0.975 \times 0.988}{0.975 + 0.988} = 0.981
 \end{aligned}$$

Berdasarkan hasil perhitungan di atas, maka didapatkan *F1-Score* rata-rata sebesar 0.981, hal ini menunjukkan model yang dirancang memiliki kualitas yang baik karena sudah mendekati nilai tertinggi yang bisa dicapai.

(d) Hasil Pengujian

Hasil pengujian ini mempertimbangkan *confidence threshold* yang merupakan nilai probabilitas yang digunakan untuk menentukan apakah suatu

deteksi dianggap benar atau tidak, sesuai dengan tingkat kepercayaan yang diberikan oleh model. Pada model YOLOv5 ini *confidence threshold* yang digunakan sebesar 0.25. Nilai ini merupakan nilai *default* yang diberikan oleh YOLOv5. Menurut dokumentasi resmi YOLOv5, direkomendasikan untuk menggunakan nilai *hyperparameter default* terlebih dahulu sebelum memodifikasinya. Pengujian dilakukan pada tiga kondisi, yaitu pengujian pada *Google Colab*, pengujian pada fitur deteksi langsung di aplikasi Sekata, dan pengujian pada fitur deteksi gambar di aplikasi Sekata. Nantinya model akan mendeteksi gerakan BISINDO dan akan menandainya dengan *bounding box* dan memberi keterangan kelas beserta tingkat *confidence* pendeteksian.

Pengujian pada *Google Colab* menggunakan data *test* yang tidak termasuk dalam *data training* atau validasi, artinya data ini tidak pernah dilihat model sebelumnya. Model akan mendeteksi gambar yang diunggah ke *Google Colab*. Hasilnya, model mampu mendeteksi semua kelas dengan baik.

Pengujian pada fitur deteksi langsung di aplikasi Sekata dilakukan dengan cara mengarahkan kamera *smartphone* ke arah peraga gerakan BISINDO untuk sebuah kata dan akan dideteksi oleh model. Pengujian pada fitur deteksi langsung dilakukan dengan jarak terdekat 1 meter dan jarak terjauh 2 meter. Hasilnya, model mampu mendeteksi semua kelas dengan baik pada jarak 1 meter. Namun, pada jarak 2 meter, model melakukan kesalahan deteksi pada kelas jam, kamu, dan nama, bahkan tidak mampu mendeteksi kelas di sini dan saya.

Pengujian pada fitur deteksi gambar di aplikasi Sekata dilakukan dengan cara memasukkan gambar ke dalam aplikasi Sekata dan akan dideteksi oleh model. Pengujian pada fitur deteksi gambar dilakukan dengan intensitas cahaya 100%, 50%, dan 20%. Hasilnya, model mampu mendeteksi seluruh kelas dengan baik pada intensitas cahaya 100%. Namun, pada intensitas cahaya 50% model tidak mampu mendeteksi kelas di sini dan siapa. Sedangkan, pada intensitas cahaya 20% model melakukan kesalahan deteksi untuk

kelas ambil, di sini, jam, kamu, tidak, dan tolong.

#### 4.4 Perancangan UI/UX dengan Metode *User Centered Design* (UCD) Untuk Aplikasi Sekata

Topik ini bertujuan untuk merancang tampilan antarmuka aplikasi dengan metode UCD yang berfokus pada kebutuhan pengguna agar pengguna mudah dan nyaman ketika menggunakan aplikasi. Sebelum masuk ke dalam proses metode UCD, dilakukan perancangan desain dan *mockup* awal tampilan antarmuka aplikasi. Setelah itu, dilakukan perancangan menggunakan metode UCD yang tahapannya sebagai berikut.

##### (a) *Plan The Human Centered Process*

Pada tahap ini dilakukan pengumpulan informasi melalui studi literatur untuk mendapatkan pemahaman tentang penerapan metode UCD dalam merancang tampilan antarmuka aplikasi.

##### (b) *Specify The Context of Use*

Pada tahap ini, dilakukan identifikasi siapa yang akan menggunakan aplikasi. Masyarakat umum pun dipilih sebagai pengguna aplikasi agar meningkatkan pemahaman dan kesadaran masyarakat umum terkait dengan Bahasa Isyarat Indonesia (BISINDO). Agar mendapatkan data yang lebih representatif mengenai kebutuhan aplikasi, tahap ini menggunakan *purposive sampling* dan mengambil sampel guru yang mengajar di SLB N 1 Badung karena memahami kebutuhan dan tantangan saat berkomunikasi dengan penyandang tunarungu.

##### (c) *Specify User and Organizational Requirements*

Pada tahap ini, guru SLB N 1 Badung diberikan kuesioner secara langsung yang berisikan pertanyaan dan *mockup* awal tampilan antarmuka aplikasi berdasarkan *wireframe* untuk mengetahui apa saja kebutuhan dan saran pengguna yang nantinya akan diterapkan pada aplikasi ini. Data didapatkan dari 20 responden yang merupakan guru SLB N 1 Badung.

##### (d) *Produce Design Solution*

Pada tahap ini dilakukan perancangan tampilan antarmuka aplikasi berdasarkan data kebutuhan pengguna



yang telah dianalisis pada tahap sebelumnya. Tahap ini akan memperbarui desain dan *mockup* awal tampilan antarmuka aplikasi dan menambah beberapa fitur sehingga menghasilkan tampilan antarmuka aplikasi yang sesuai dengan kebutuhan pengguna.

(e) *Evaluate Design Against User Requirement*

Pada tahap ini dilakukan evaluasi untuk menguji kemudahan dan kenyamanan tampilan antarmuka aplikasi yang telah dirancang menggunakan metode *System Usability Scale* (SUS). Pengujian ini melibatkan masyarakat umum dan guru SLB N 1 Badung sebagai pengguna dan dilakukan dengan cara meminta responden mencoba aplikasi, setelah itu mengisi kuesioner yang berisi 10 pernyataan metode SUS secara daring. Setelah mendapatkan hasil kuesioner SUS, data tersebut diolah sesuai dengan ketentuan dari metode SUS. Adapun hasil pengolahannya dapat dilihat pada tabel 2.

Tabel 2. Hasil pengolahan kuesioner SUS

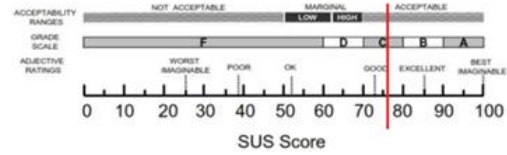
Responden	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Jumlah	Nilai (*2,5)
Responden 1	2	3	3	3	3	3	3	3	3	3	29	72,5
Responden 2	3	3	3	4	3	3	3	3	3	3	31	77,5
Responden 3	3	3	3	3	3	3	3	3	3	2	29	72,5
Responden 4	3	1	3	1	4	1	3	3	3	3	25	62,5
Responden 5	3	2	3	3	3	3	3	2	3	2	27	67,5
Responden 6	3	3	3	1	3	3	3	3	3	3	28	70
Responden 7	3	3	3	3	3	1	3	3	3	3	28	70
Responden 8	3	4	3	4	3	3	3	4	3	4	34	85
Responden 9	3	3	3	3	3	3	3	3	3	4	31	77,5
Responden 10	3	3	3	3	3	3	3	3	3	3	30	75
Responden 11	3	1	3	1	3	3	2	2	4	4	24	60
Responden 12	4	3	4	4	3	3	4	4	4	3	36	90
Responden 13	3	3	4	1	3	3	3	3	3	1	27	67,5
Responden 14	3	3	3	3	3	3	3	3	3	1	28	70
Responden 15	3	4	4	4	4	4	4	4	4	1	36	90
Responden 16	3	4	4	1	4	4	4	4	4	1	33	82,5
Responden 17	3	3	3	4	2	3	3	4	3	1	29	72,5
Responden 18	3	3	4	3	3	4	3	3	4	3	33	82,5
Responden 19	3	3	4	3	3	3	4	4	3	4	34	85
Responden 20	3	4	4	4	3	4	3	4	3	3	35	87,5
Responden 21	4	4	3	3	4	4	4	4	2	3	35	87,5
Responden 22	3	3	4	3	3	2	3	3	3	3	30	75
Responden 23	4	4	3	3	4	3	4	4	4	4	37	92,5
Responden 24	3	3	3	1	3	2	3	3	3	2	26	65
Responden 25	3	3	3	1	3	3	3	3	4	2	28	70
Responden 26	1	3	3	3	3	2	3	3	3	3	27	67,5
Responden 27	3	3	3	4	3	3	3	3	3	3	31	77,5
Responden 28	3	4	4	4	4	4	3	3	4	3	36	90
Responden 29	1	3	4	0	4	1	4	3	4	1	25	62,5
Responden 30	3	3	3	4	3	2	3	3	3	3	30	75
TOTAL											2280	

Setelah itu, nilai SUS didapatkan dari membagi total nilai dengan jumlah responden. Adapun perhitungannya sebagai berikut.

$$\begin{aligned}
 \text{Nilai SUS} &= \frac{\sum x}{n} \\
 \text{Nilai SUS} &= \frac{2280}{30} = 76
 \end{aligned}$$

Berdasarkan hasil perhitungan nilai SUS, didapatkan nilai sebesar 76. Selanjutnya, nilai ini diinterpretasikan ke

dalam bagan penilaian *System Usability Scale* (SUS). Adapun hasil interpretasi nilai SUS dapat dilihat pada gambar 4.



Gambar 4. Hasil interpretasi nilai SUS

Berdasarkan gambar 4, dapat dilihat bahwa nilai letak nilai SUS diwakilkan oleh garis berwarna merah yang berarti tampilan antarmuka aplikasi Sekata mendapatkan *acceptability ranges* dengan hasil *acceptable*, *grade scale* dengan hasil C, dan *adjective ratings* dengan hasil *good* bernilai sebesar 76. Hal ini menandakan bahwa tampilan antarmuka aplikasi Sekata memiliki tingkat *usability* yang baik.

4.5 Perancangan Aplikasi Sekata Berbasis Mobile dengan Metode Prototype

Topik ini mengimplementasikan model *deep learning* dan desain tampilan antarmuka yang telah dibuat sebelumnya ke dalam aplikasi sehingga menjadi aplikasi penerjemah BISINDO berbasis mobile dengan nama Sekata yang bisa dijalankan di *smartphone* Android. Adapun tahapan yang dilakukan adalah sebagai berikut.

(a) *Communication*

Tahap ini dimulai dengan melakukan diskusi mengenai fitur apa saja yang akan ada di dalam aplikasi. Dari hasil diskusi awal, ditetapkan beberapa fitur utama untuk aplikasi, yaitu deteksi langsung, deteksi gambar, kamus BISINDO, dan tentang aplikasi. Kemudian, dilanjutkan dengan melakukan komunikasi kepada guru SLB N 1 Badung untuk memperoleh masukan lebih lanjut. Hasilnya, terdapat beberapa saran penambahan fitur, yaitu fitur deteksi langsung, deteksi gambar, kamus BISINDO, tentang BISINDO, tentang aplikasi, dan petunjuk aplikasi.

(b) *Quick Plan*

Langkah selanjutnya adalah melakukan perencanaan perancangan aplikasi yang melibatkan analisis teknologi dan analisis pengguna. Teknologi yang dipilih untuk merancang aplikasi ini yaitu, HTML, CSS, JavaScript, dan JSON, serta

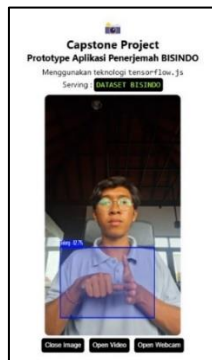
didukung beberapa library dan framework seperti ReactJS, TensorFlow.js, dan CapacitorJS. Kemudian, ditentukan juga bahwa aplikasi yang dirancang akan ditujukan kepada masyarakat umum agar meningkatkan pemahaman tentang BISINDO di kalangan masyarakat umum.

(c) *Modelling Quick Design*

Pada tahap ini dilakukan perancangan gambaran umum dan *Unified Modelling Language* (UML) berupa *Use Case Diagram* serta *Activity Diagram*. Selanjutnya, dilakukan juga penyesuaian pada UML agar dapat menambahkan fitur yang disarankan oleh pengguna, yaitu tentang BISINDO, petunjuk aplikasi, dan pop-up untuk menampilkan profil tim pengembang pada halaman tentang aplikasi. Penyesuaian tersebut untuk memastikan agar semua fitur yang diinginkan dapat diimplementasikan dengan baik dan sesuai dengan kebutuhan pengguna.

(d) *Construction of Prototype*

Pada tahap ini dilakukan perancangan *prototype* sekaligus mengimplementasikan model YOLOv5 dan rancangan tampilan antarmuka ke dalam aplikasi. Pengimplementasian YOLOv5 dilakukan dengan mengonversi model menjadi format JSON dan tujuh file *shard* yang berformat bin agar bisa digunakan di lingkungan JavaScript. Tampilan *prototype* aplikasi setelah mengimplementasikan model YOLOv5 dapat dilihat pada gambar 5.



Gambar 5. Tampilan *prototype* aplikasi setelah mengimplementasikan model YOLOv5

Pengimplementasian tampilan antarmuka dilakukan dengan cara mengembangkan desain yang telah

disetujui menjadi kode menggunakan HTML, CSS, JavaScript, dan framework ReactJS. Tampilan *prototype* aplikasi setelah mengimplementasikan tampilan antarmuka dapat dilihat pada gambar 6.



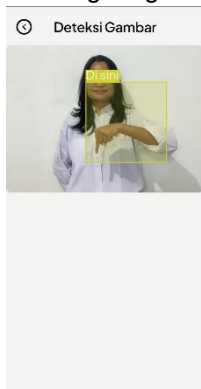
Gambar 6. Tampilan *prototype* aplikasi setelah mengimplementasikan tampilan antarmuka

(e) *Deployment Delivery & Feedback*

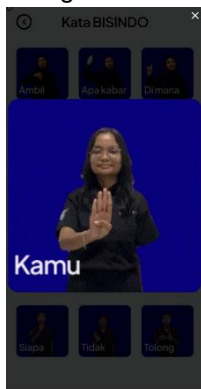
Setelah aplikasi selesai dirancang, tahap selanjutnya adalah pengujian aplikasi untuk memastikan bahwa aplikasi dapat beroperasi sesuai dengan kebutuhan dan tujuan yang diinginkan. Pengujian dilakukan oleh peneliti dengan menggunakan metode *black box testing* yang bertujuan untuk menguji berbagai fungsionalitas pada aplikasi yang telah dirancang. Hasilnya, seluruh fungsional aplikasi yang telah dibangun telah terpenuhi dan berfungsi dengan baik sehingga aplikasi siap untuk digunakan oleh masyarakat umum. Berikut ini merupakan tampilan dari beberapa fungsional utama aplikasi Sekata yang dapat dilihat pada gambar 7, gambar 8, dan gambar 9.



Gambar 7. Tampilan fungsi deteksi langsung



Gambar 8. Tampilan fungsi deteksi gambar



Gambar 9. Tampilan fungsi kamus BISINDO

Dari gambar 7, gambar, 8, dan gambar 9 dapat dilihat bahwa fungsionalitas aplikasi untuk mendeteksi langsung, mendeteksi gambar, dan menampilkan kamus BISINDO berfungsi dengan baik.

## 5. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa:

1. Berhasil mengumpulkan dan memproses dataset gerakan tangan untuk penerjemahan BISINDO menggunakan *Roboflow* yang divalidasi oleh guru SLB N 1 Badung. Dataset ini siap digunakan untuk melatih model YOLOv5 untuk mengenali gerakan BISINDO.
2. Berhasil merancang model YOLOv5 untuk menerjemahkan BISINDO menggunakan 3000 dataset gambar gerakan kata BISINDO dan berhasil mengenali 15 kelas yang dibuat, yaitu Ambil, Apa-kabar, Di-mana, Di-sana, Di-sini, Iya, Jam, Kamu, Nama, Rumah, Sama, Saya, Siapa, Tidak, dan Tolong pada pengujian menggunakan *Google Colab*, Deteksi langsung aplikasi sekata, dan deteksi gambar aplikasi sekata. Perancangan model YOLOv5 menggunakan 32 *batch* dan 120 *epoch* mendapatkan nilai rata-rata mAP50 sebesar 0.982 dan evaluasi kinerja model menggunakan *F1-Score* sebesar 0.981.
3. Pengujian pengaruh jarak pendeteksian terhadap fitur deteksi langsung pada aplikasi Sekata dilakukan dengan jarak terdekat sebesar 1 meter dan jarak terjauh sebesar 2 meter. Hasil pengujian menunjukkan bahwa pada jarak 1 meter, model dapat mendeteksi semua kelas dengan baik pada kelas Ambil, Apa kabar, Di mana, Di sana, Di sini, Iya, Jam, Kamu, Nama, Rumah, Sama, Saya, Siapa, Tidak, Tolong. Namun, pada jarak 2 meter, model melakukan beberapa kesalahan deteksi yaitu pada kelas Jam, Kamu, dan Nama, bahkan tidak dapat mendeteksi kelas sama sekali seperti pada kelas Di sini, dan Saya. Dari pengujian tersebut dapat disimpulkan bahwa jarak terjauh pendeteksian agar model dapat melakukan pendeteksian secara optimal sebesar 1 meter.
4. Pengujian pengaruh intensitas cahaya terhadap fitur deteksi gambar pada

aplikasi Sekata dengan menggunakan intensitas cahaya 100%, 50%, dan 20%. Hasil pengujian menunjukkan bahwa pada intensitas cahaya 100%, model dapat mendeteksi semua kelas dengan baik pada kelas Ambil, Apa kabar, Di mana, Di sana, Di sini, Iya, Jam, Kamu, Nama, Rumah, Sama, Saya, Siapa, Tidak, dan Tolong. Namun, pada intensitas cahaya 50% model tidak dapat mendeteksi kelas Di sini dan Siapa. Kemudian, pada Intensitas cahaya 20%, model mengalami kesalahan deteksi pada kelas Apa kabar, bahkan tidak dapat melakukan pendeteksian sama sekali pada kelas Ambil, Di sini, Jam, Kamu, Tidak, dan Tolong. Oleh karena itu, dapat disimpulkan bahwa performa deteksi model dipengaruhi oleh intensitas cahaya.

5. Perancangan tampilan antarmuka aplikasi berhasil dilakukan menggunakan metode *User Centered Design* (UCD). Tampilan antarmuka dievaluasi menggunakan metode *System Usability Scale* (SUS) dan mendapatkan nilai sebesar 76 yang berarti tampilan antarmuka aplikasi Sekata mendapatkan *acceptability ranges* dengan hasil *acceptable*, *grade scale* dengan hasil C, dan *adjective ratings* dengan hasil *good* bernilai sebesar 76. Hal ini menandakan bahwa tampilan antarmuka aplikasi Sekata memiliki tingkat *usability* yang baik sehingga tampilan antarmuka yang telah dirancang mudah digunakan oleh pengguna.
6. Aplikasi Sekata berbasis mobile berhasil dirancang menggunakan metode *prototype*. Perancangan ini mengimplementasikan model YOLOv5 dan tampilan antarmuka ke dalam aplikasi menggunakan HTML, CSS, JavaScript, dan JSON, serta didukung beberapa library dan framework seperti ReactJS, TensorflowJS, dan CapacitorJS. Selanjutnya, aplikasi diuji menggunakan metode *black box testing*

oleh peneliti dan hasilnya seluruh fungsionalitas pada aplikasi telah terpenuhi dan berfungsi dengan baik sehingga aplikasi siap untuk digunakan oleh masyarakat umum.

7. Penelitian ini menghasilkan sebuah aplikasi mobile bernama Sekata yang bisa digunakan pada *smartphone* android untuk mendeteksi dan menerjemahkan BISINDO menggunakan model YOLOv5. Aplikasi ini memanfaatkan kemampuan YOLOv5 dalam mendeteksi objek secara *real-time* untuk menerjemahkan gerakan BISINDO dari beberapa kata menjadi sebuah teks. diharapkan aplikasi ini dapat meningkatkan pemahaman tentang BISINDO di kalangan masyarakat umum.

## 6. DAFTAR PUSTAKA

- [1] D. Gustiar, S. H. Sitorus, and D. M. Midyanti, "Penerjemahan Bahasa Isyarat Menggunakan Metode Generalized Learning Vector Quantization (GLVQ)," *Coding Jurnal Komputer dan Aplikasi*, vol. 8, no. 3, pp. 1–8, Sep. 2020, doi: 10.26418/coding.v8i3.42156.
- [2] A. S. Nugraheni, A. P. Husain, and H. Unayah, "Optimalisasi Penggunaan Bahasa Isyarat Dengan SIBI Dan BISINDO Pada Mahasiswa Difabel Tunarungu Di Prodi PGMI UIN Sunan Kalijaga," *Holistika: Jurnal Ilmiah PGSD*, vol. 5, no. 1, pp. 28–33, May 2021, doi: 10.24853/holistika.5.1.28-33.
- [3] S. Nita and A. D. Cahyanti, "Penggunaan Aplikasi i-Chat Sebagai Sarana Teknologi Pembelajaran Bahasa Isyarat Bagi Masyarakat Awam," *Seminar Nasional Hasil Penelitian dan Pengabdian Kepada Masyarakat UNIPMA*, pp. 174–180, 2019.
- [4] S. S. Sindarto, D. E. Ratnawati, and I. Arwani, "Klasifikasi Citra Sistem Isyarat Bahasa Indonesia (SIBI) dengan Metode Convolutional Neural Network pada Perangkat Lunak berbasis Android," *Jurnal Pengembangan Teknologi dan Ilmu*

- Komputer (JPTIIK)*, vol. 6, no. 5, pp. 2129–2138, May 2022, [Online]. Available: <http://j-ptiik.ub.ac.id>
- [5] R. Imantiyar and D. H. Fudholi, “Kajian Pengaruh Dataset dan Bias Dataset terhadap Performa Akurasi Deteksi Objek,” *Jurnal Petir (Pengkajian dan Penerapan Teknik Informatika)*, vol. 14, no. 2, pp. 258–268, Sep. 2021, doi: 10.33322/petir.v14i2.1350.
- [6] S. Supardianto and A. B. Tampubolon, “Penerapan UCD (User Centered Design) Pada Perancangan Sistem Informasi Manajemen Aset TI Berbasis Web di Bid TIK Kepolisian Daerah Kepulauan Riau,” *Journal of Applied Informatics and Computing*, vol. 4, no. 1, pp. 74–83, Jun. 2020, doi: 10.30871/jaic.v4i1.2108.
- [7] Y. Yahya and M. Mahpuz, “Penggunaan Algoritma K-Means Untuk Menganalisis Pelanggan Potensial Pada Dealer SPS Motor Honda Lombok Timur Nusa Tenggara Barat,” *Infotek : Jurnal Informatika dan Teknologi*, vol. 2, no. 2, pp. 109–118, Aug. 2019, doi: 10.29408/jit.v2i2.1447.
- [8] A. A. S. Arip, N. Sazali, K. Kadirgama, A. S. Jamaludin, F. M. Turan, and N. Ab. Razak, “Object Detection for Safety Attire Using YOLO (You Only Look Once),” *Journal of Advanced Research in Applied Mechanics*, vol. 113, no. 1, pp. 37–51, Jan. 2024, doi: 10.37934/aram.113.1.3751.
- [9] A. Amwim, “Deteksi Dan Klasifikasi Kendaraan Berbasis Algoritma You Only Look Once (YOLO),” Skripsi, Universitas Islam Indonesia, Yogyakarta, 2021. Accessed: Dec. 12, 2023. [Online]. Available: <https://dspace.uii.ac.id/handle/123456789/34154>
- [10] M. A. Rizkhullah and A. Voutama, “Perancangan UI/UX Aplikasi Mobile Untuk Pembelajaran Adaptif Public Speaking Dengan Metode User Centered Design,” *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 12, no. 2, pp. 1250–1256, Apr. 2024, doi: 10.23960/jitet.v12i2.4204.
- [11] J. Rohman, N. A. A. Bashir, J. Ipmawati, and F. F. Laksana, “Permodelan UI/UX Aplikasi Santri Information Management System (SAIMS) Menggunakan Metode User Centered Design (UCD),” *JIKO (Jurnal Informatika dan Komputer)*, vol. 7, no. 1, p. 72, Feb. 2023, doi: 10.26798/jiko.v7i1.702.
- [12] A. Hidayat and N. A. Amri, “Implementasi Aplikasi Profil Institusi Berbasis Android Program Studi Bisnis Internasional Dan Pemasaran Digital,” *Jurnal Manajemen Informatika (JUMIKA)*, vol. 9, no. 1, pp. 11–20, 2022, doi: 10.51530/jumika.v9i1.650.
- [13] A. H. Saputra and D. H. Fudholi, “Realtime Object Detection Masa Siap Panen Tanaman Sayuran Berbasis Mobile Android Dengan Deep Learning,” *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 5, no. 4, pp. 647–655, Aug. 2021, doi: 10.29207/resti.v5i4.3190.
- [14] H. M. B. Garrido, “Expert Validation of a Python Test, Reliability, Difficulty and Discrimination Indices,” *Journal of Education and Development*, vol. 7, no. 1, p. 52, Feb. 2023, doi: 10.20849/jed.v7i1.1320.
- [15] G. Jocher and A. Vina, “YOLO Performance Metrics.” Accessed: Dec. 12, 2023. [Online]. Available: <https://docs.ultralytics.com/guides/yolo-performance-metrics/#class-wise-metrics>
- [16] L. Lusiana, A. Wibowo, and T. K. Dewi, “Implementasi Algoritma Deep Learning You Only Look Once (YOLOv5) Untuk Deteksi Buah Segar Dan Busuk,” *Paspalum: Jurnal Ilmiah Pertanian*, vol. 11, no. 1, pp. 123–130, Mar. 2023, doi: 10.35138/paspalum.v11i1.489.