# BAB II
# LANDASAN TEORI

Bab ini memuat kajian pustaka yang mencakup pembahasan terhadap penelitian-penelitian sebelumnya serta teori-teori yang berkaitan dengan topik yang diteliti. Kajian ini bertujuan untuk memberikan gambaran konteks penelitian, mencegah terjadinya pengulangan studi yang sudah ada, mengidentifikasi celah atau kekurangan pada penelitian terdahulu, serta merumuskan kontribusi baru yang dihasilkan dari penelitian ini.

## 2.1. Penelitian Terdahulu

Dalam sub bab ini akan dijabarkan beberapa penelitian yang relevan terkait pengembangan sistem pengenalan bahasa isyarat, khususnya Bahasa Isyarat Indonesia (BISINDO), dengan memanfaatkan teknologi YOLOv11. Penelitian-penelitian terdahulu diambil dari jurnal nasional maupun internasional dan dijelaskan secara ringkas mencakup judul penelitian, metode yang digunakan, hasil analisis, serta mengidentifikasi kesenjangan penelitian yang ada untuk memperlihatkan relevansi sekaligus perbedaan dengan penelitian yang sedang dilakukan. Adapun ringkasan penelitian terdahulu dapat disajikan pada Tabel 2.1 berikut.

**Tabel 2. 1** Penelitian Terdahulu

| No | Nama Penulis dan Tahun | Judul Penelitian | Hasil Analisis | Gap dengan Penelitian Sekarang |
|---|---|---|---|---|
| 1 | Ariansyah (2024) | Pendeteksi Kata dalam Bahasa Isyarat Menggunakan Algoritma YOLO Versi 8 | YOLOv8 berhasil digunakan untuk mendeteksi kata dalam bahasa isyarat dengan akurasi 93,8%. Meskipun hasilnya sangat baik, performa deteksi masih dipengaruhi oleh variasi pencahayaan, posisi tangan, dan jarak pengambilan gambar, yang dapat menurunkan konsistensi hasil pada kondisi tertentu. | Penelitian ini fokus pada deteksi kata, sementara penelitian sekarang menekankan pada pengembangan website edukasi interaktif BISINDO berbasis YOLOv11 sebagai media pembelajaran. |
| 2 | Kurniawan & Arie | *YOLOv8 Algorithm to Improve the Sign* | YOLOv8 mampu mendeteksi alfabet SIBI dengan akurasi | Penelitian ini fokus pada pengembangan model deteksi alfabet SIBI, |

| No | Nama Penulis dan Tahun | Judul Penelitian | Hasil Analisis | Gap dengan Penelitian Sekarang |
|---|---|---|---|---|
| | Wijaya (2025) | *Language Letter Detection System Model* | rata-rata 90–96%, *inference time* 4.6 ms, dan mAP50 mencapai 0.96. | sedangkan penelitian sekarang berfokus pada implementasi YOLOv11 dalam website edukasi BISINDO untuk pembelajaran interaktif. |
| 3 | Tri Ahmad Jabar, Rudi Heriansyah, dan Evi Purnamasari (2025) | Pengenalan Pola Huruf Bahasa Isyarat Menggunakan Framework You Only Look Once (YOLO) | Penelitian ini mengembangkan sistem pengenalan huruf Bahasa Isyarat Indonesia (SIBI) menggunakan model YOLOv11 dengan dataset yang dianotasi melalui Roboflow dan dilatih di Google Colab. Hasilnya menunjukkan performa tinggi dengan *Precision* 0,835, *Recall* 0,928, dan mAP@50 0,968. | Penelitian tersebut berfokus pada pengenalan huruf SIBI secara *real-time*, sedangkan penelitian ini mengembangkan website edukasi BISINDO berbasis YOLOv11 yang dilengkapi fitur latihan interaktif untuk pembelajaran gerakan tangan, bukan sekadar deteksi huruf. |
| 4 | Alaftekin, Pacal, & Cicek (2024) | *Real-time Sign Language Recognition Based on YOLO Algorithm* | Penelitian ini mengembangkan model YOLOv4-CSP dengan tambahan CSPNet, *Mish activation*, dan *Transformer block* untuk deteksi bahasa isyarat Turki secara *real-time*. Model mencapai akurasi tinggi dengan *Precision* 98.95%, *Recall* 98.15%, *F1-score* 98.55%, dan mAP 99.49%. | Penelitian ini berfokus pada optimisasi YOLOv4-CSP untuk bahasa isyarat Turki, sedangkan penelitian sekarang menerapkan YOLOv11 untuk BISINDO dan mengintegrasikannya ke dalam website edukasi interaktif sebagai media pembelajaran. |
| 5 | Ahmad Alfin Nur Hakim & Muhamad Prianda (2020) | Implementasi Pengenalan Bahasa Isyarat *Real-time* Menggunakan YOLOv11 untuk Komunikasi Inklusif | Penelitian ini mengembangkan sistem pengenalan bahasa isyarat *real-time* berbasis YOLOv11 untuk mendeteksi alfabet statis ASL (A–Z). Dengan dataset 520 gambar berukuran 640×640 piksel, model mencapai performa tinggi dengan mAP@0.5 sebesar 99,5%, *presisi* 97,8%, dan *Recall* 100%, menunjukkan kemampuan deteksi yang akurat dan stabil. | Penelitian ini berfokus pada pengenalan alfabet ASL (*American Sign Language*) menggunakan YOLOv11 dalam lingkungan komputasi terbatas. Sementara penelitian sekarang menerapkan YOLOv11 untuk BISINDO, serta mengintegrasikannya ke dalam website edukasi interaktif guna mendukung pembelajaran inklusif. |

### 2.1.1. Penelitian oleh Ariansyah, 2024

Ariansyah (2024) meneliti pendeteksian kata dalam Bahasa Isyarat Indonesia menggunakan algoritma YOLOv8. Penelitian ini menunjukkan bahwa model mampu mengenali kata bahasa isyarat dengan tingkat akurasi tinggi dan waktu deteksi yang cepat. Dataset diperoleh dari citra gestur tangan pengguna, kemudian dilatih untuk mengenali kata berdasarkan kelas tertentu. Namun, performa model masih dipengaruhi oleh kondisi pencahayaan dan variasi posisi tangan pengguna. Persamaannya, penelitian ini dan penelitian sekarang sama-sama menggunakan algoritma YOLO dalam penerapan *computer vision* untuk deteksi gerakan tangan Bahasa Isyarat Indonesia. Gap-nya, penelitian Ariansyah berfokus pada deteksi kata saja, sedangkan penelitian sekarang mengembangkan website edukasi BISINDO berbasis YOLOv11 yang tidak hanya melakukan deteksi gerakan, tetapi juga menyediakan fitur latihan interaktif dan pembelajaran *real-time* untuk mendukung proses edukasi bahasa isyarat.

### 2.1.2. Penelitian oleh Kurniawan dan Arie Wijaya 2025

Kurniawan & Arie Wijaya (2025) melakukan penelitian terkait pengembangan sistem deteksi alfabet SIBI menggunakan algoritma YOLOv8. Hasil penelitian menunjukkan performa tinggi dengan akurasi rata-rata 90–96%, waktu inferensi cepat sebesar 4,6 ms, dan nilai mAP50 mencapai 0,96. Penelitian ini membuktikan bahwa YOLOv8 mampu melakukan deteksi bahasa isyarat secara *real-time* dengan tingkat akurasi yang baik. Persamaannya, penelitian ini dan penelitian sekarang sama-sama memanfaatkan algoritma YOLO sebagai pendekatan utama dalam pengenalan gerakan tangan berbasis *computer vision*. Perbedaannya, penelitian terdahulu berfokus pada pengembangan model deteksi alfabet SIBI, sedangkan penelitian sekarang menekankan pada implementasi YOLOv11 dalam sistem pembelajaran BISINDO interaktif berbasis website yang dilengkapi fitur latihan dan evaluasi pengguna secara *real-time*.

### 2.1.3. Penelitian oleh Jabar et al., 2025

Jabar et al. (2025) mengembangkan sistem pengenalan huruf Bahasa Isyarat Indonesia (SIBI) menggunakan algoritma YOLOv11. Dataset diperoleh dari citra gestur tangan yang dianotasi dengan Roboflow dan dilatih di Google Colab. Hasil penelitian menunjukkan performa tinggi dengan *Precision* 0,835, *Recall* 0,928, dan mAP@50 0,968. Persamaannya, penelitian ini dan penelitian sekarang sama-sama menggunakan algoritma YOLOv11 untuk mendeteksi gerakan tangan Bahasa Isyarat Indonesia secara *real-time*. Perbedaannya, penelitian ini berfokus pada pengenalan huruf SIBI, sedangkan penelitian sekarang mengembangkan website edukasi BISINDO interaktif yang dilengkapi dengan fitur latihan dan pembelajaran *real-time* untuk tujuan edukatif.

### 2.1.4. Penelitian oleh Alaftekin et al., 2024

Penelitian ini membahas sistem pengenalan bahasa isyarat Turki secara *real-time* menggunakan YOLOv4-CSP yang telah dioptimasi dengan penambahan CSPNet, *Mish activation function*, dan *Transformer block*. Hasil pengujian menunjukkan performa tinggi dengan mAP sebesar 99.49% dan F1-score sebesar 98.55%. Sistem ini mampu mengenali gerakan tangan tanpa memerlukan *preprocessing* atau latar belakang seragam. Relevansinya dengan penelitian sekarang terletak pada penggunaan YOLO sebagai basis deteksi bahasa isyarat, namun penelitian ini berfokus pada optimasi model untuk bahasa isyarat Turki, sementara penelitian sekarang menitikberatkan pada penerapan YOLOv11 dalam website edukasi interaktif BISINDO untuk pembelajaran inklusif.

### 2.1.5. Penelitian oleh Hakim dan Prianda (2020)

Penelitian oleh Ahmad Alfin Nur Hakim dan Muhamad Prianda (2020) mengusulkan sistem deteksi bahasa isyarat berbasis YOLOv11 untuk mengenali alfabet statis *American Sign Language* (ASL). Model dilatih menggunakan dataset berisi 520 gambar dengan ukuran 640×640 piksel dan menunjukkan hasil sangat baik dengan nilai mAP@0.5 sebesar 99,5%, presisi 97,8%, *Recall* 100%, dan F1-score 98,9%. Penelitian ini membuktikan efektivitas YOLOv11 dalam mendeteksi gestur tangan secara waktu nyata dan berpotensi besar dalam pengembangan teknologi komunikasi asistif. Persamaan penelitan ini dengan penelitan sekarang terletak pada penggunaan YOLOv11. Perbedaannya dengan penelitian sekarang

terletak pada fokus bahasa dan konteks penerapan, di mana penelitian ini berorientasi pada ASL, sementara penelitian sekarang menerapkan YOLOv11 untuk Bahasa Isyarat Indonesia (BISINDO) dalam platform website edukasi interaktif yang mendukung pembelajaran inklusif.

## 2.2. Landasan Teori

Dalam bab ini akan dibahas teori-teori yang menjadi dasar dalam pelaksanaan penelitian ini. Landasan teori disusun secara sistematis, dimulai dari konsep umum mengenai Bahasa Isyarat Indonesia (BISINDO), kemudian dilanjutkan dengan teori mengenai teknologi *computer vision* hingga teori khusus terkait implementasi algoritma YOLO serta pengembangan website edukasi interaktif.

Teori-teori yang dibahas ini akan menjadi pijakan konseptual dalam merancang dan membangun sistem, mulai dari pemahaman pentingnya BISINDO sebagai sarana komunikasi inklusif, dasar kerja pengolahan citra digital, prinsip kerja deteksi objek menggunakan YOLO, hingga kerangka implementasi sistem berbasis web. Dengan adanya landasan teori ini, penelitian dapat memiliki arah yang jelas dalam mewujudkan website edukasi BISINDO dengan fitur latihan interaktif *real-time* dan umpan balik langsung melalui kamera.

### 2.2.1. Bahasa Isyarat Indonesia (BISINDO)

Bahasa Isyarat Indonesia (BISINDO) merupakan bentuk bahasa visual yang digunakan oleh komunitas Tuli di Indonesia untuk berinteraksi menggunakan gerakan tangan, ekspresi wajah, serta posisi tubuh. Tidak seperti Sistem Isyarat Bahasa Indonesia (SIBI) yang mengikuti kaidah tata bahasa Indonesia, BISINDO berkembang secara alami di kalangan komunitas Tuli dan memiliki struktur linguistik yang khas (Sri Nugraheni et al., 2021).

Pemerintah telah menetapkan BISINDO sebagai salah satu sarana komunikasi resmi bagi penyandang disabilitas melalui Undang-Undang Nomor 8 Tahun 2016 tentang Penyandang Disabilitas. Dalam ranah teknologi, penerapan BISINDO melalui sistem berbasis pengolahan citra tangan dapat berperan sebagai jembatan komunikasi antara individu Tuli dan masyarakat umum, sekaligus mendukung terwujudnya pendidikan yang inklusif.

### 2.2.2. Sekolah Galuh Handayani

Sekolah Inklusif Galuh Handayani Surabaya merupakan lembaga pendidikan yang telah berdiri sejak tahun 1995 dan berkomitmen untuk memberikan layanan pendidikan bagi anak berkebutuhan khusus di lingkungan yang inklusif. Sekolah ini dikenal sebagai salah satu sekolah yang memiliki fasilitas lengkap dalam mendukung perkembangan siswa, termasuk terapi wicara, terapi okupasi, terapi perilaku, serta pusat sensori integrasi. Galuh Handayani melayani pendidikan dari tingkat TK, SD, SMP, hingga SMA.

**Gambar 2. 1** Sekolah Galuh Handayani Surabaya

### *2.2.3. Computer Vision*

*Computer vision* merupakan bidang dalam kecerdasan buatan yang berfokus pada kemampuan komputer untuk mengenali, menganalisis, dan menafsirkan data visual dari gambar maupun video layaknya cara manusia memahami visual di sekitarnya. Teknologi ini melibatkan kombinasi antara pengolahan citra digital, pembelajaran mesin, dan algoritma deteksi objek untuk mengenali pola, bentuk, maupun gerakan (Gupta et al., 2020).

Dalam penelitian ini, *computer vision* berperan penting dalam mendeteksi dan mengenali gestur tangan pengguna secara *real-time*. Dengan memanfaatkan kamera dan model deteksi berbasis YOLO, sistem dapat mengidentifikasi posisi serta

bentuk tangan untuk menerjemahkannya menjadi simbol Bahasa Isyarat Indonesia (BISINDO).

### 2.2.4. Website Edukasi Interaktif

Website edukasi interaktif merupakan media pembelajaran berbasis web yang dirancang untuk mendukung proses belajar melalui interaksi langsung antara pengguna dan sistem. Berbeda dari media pembelajaran konvensional, website interaktif menyediakan elemen dinamis seperti animasi, video, kuis, serta umpan balik *real-time* yang dapat meningkatkan keterlibatan dan pemahaman pengguna (Anggara & Sujatmiko, 2024).

Dalam penelitian ini, website edukasi interaktif digunakan sebagai sarana pembelajaran Bahasa Isyarat Indonesia (BISINDO). Melalui integrasi teknologi YOLOv11, pengguna dapat berlatih gerakan isyarat secara langsung menggunakan kamera dan mendapatkan hasil deteksi secara *real-time*, sehingga proses belajar menjadi lebih menarik, praktis, dan inklusif.

### 2.2.5. Framework Flask

Flask adalah *framework web* berbasis *Python* yang bersifat *microframework*, artinya menyediakan komponen inti untuk pengembangan aplikasi web tanpa memaksakan struktur tertentu. *Framework* ini ringan, fleksibel, dan mudah dikembangkan sesuai kebutuhan proyek (Ningtyas & Setiyawati, 2021).

Dalam penelitian ini, Flask digunakan untuk mengintegrasikan model deteksi bahasa isyarat dengan antarmuka website edukasi. Melalui Flask, sistem dapat menghubungkan proses inferensi dari model YOLOv11 dengan tampilan web secara *real-time*, sehingga hasil deteksi gestur tangan dapat langsung ditampilkan kepada pengguna.

### 2.2.6. Python

Python adalah bahasa pemrograman tingkat tinggi yang banyak digunakan karena memiliki sintaks yang ringkas, fleksibel, serta mudah dipelajari dan dipahami oleh pengguna. Bahasa ini mendukung berbagai paradigma pemrograman seperti *object-oriented*, *procedural*, dan *functional programming*.

Python juga memiliki ekosistem pustaka yang sangat luas sehingga memudahkan pengembang dalam membangun berbagai jenis aplikasi. Python banyak digunakan dalam bidang *machine learning* dan *computer vision* karena memiliki pustaka pendukung yang kuat, seperti OpenCV untuk pengolahan citra, TensorFlow dan PyTorch untuk pembelajaran mendalam, serta Flask untuk pengembangan web (Singh, 2025).

Dalam penelitian ini, Python berperan sebagai bahasa utama dalam pembangunan sistem deteksi BISINDO berbasis YOLOv11 dan integrasi dengan website edukasi interaktif.

### *2.2.7. Waterfall*

Metode *Waterfall* merupakan model pengembangan sistem yang bersifat klasik dan banyak diterapkan dalam bidang rekayasa perangkat lunak. Model ini menerapkan metode yang sistematis dan bertahap, di mana setiap proses harus diselesaikan sepenuhnya sebelum melanjutkan ke proses berikutnya. Pendekatan *Waterfall* dianggap ideal untuk proyek dengan kebutuhan sistem yang sudah jelas, stabil, serta memerlukan dokumentasi yang lengkap pada setiap fase pengembangannya.

Saputra (2024) menjelaskan bahwa metode *Waterfall* terdiri dari beberapa tahapan utama, yaitu analisis kebutuhan sistem, desain sistem, implementasi, dan pengujian. Tahapan-tahapan tersebut dilakukan secara bertahap, di mana hasil dari setiap tahap menjadi masukan bagi tahap berikutnya. Model ini memudahkan peneliti dalam melakukan kontrol proses pengembangan karena setiap langkah terdokumentasi dengan baik.

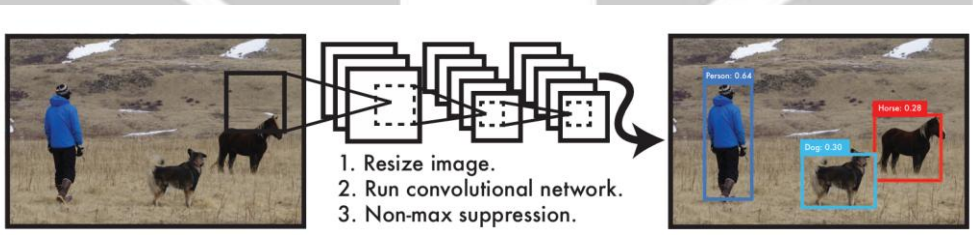
### 2.2.8. YOLO

YOLO (*You Only Look Once*) adalah algoritma deteksi objek berbasis *deep learning* yang dikembangkan untuk mengenali dan menemukan objek dengan cepat serta presisi tinggi hanya dalam satu kali proses inferensi. Berbeda dengan metode konvensional yang memerlukan tahap pemindaian berulang, YOLO memproses seluruh citra sekaligus, sehingga mampu melakukan deteksi secara *real-time* (Rismayanti & Rahmadewi, 2025).

Menurut (Malik & Mangellak, 2024) algoritma YOLO memiliki keunggulan dalam kecepatan inferensi *real-time* dibandingkan detektor dua-tahap seperti Faster R-CNN yang walaupun memiliki akurasi lebih tinggi, waktu pemrosesannya jauh lebih lama. YOLO mencapai *Precision* dan *Recall* masing-masing ~0,98 dan accuracy ~0,96, sedangkan Faster R-CNN sedikit di bawahnya (~0,95 accuracy) dalam tugas deteksi

YOLO bekerja dengan membagi citra ke dalam grid dan memprediksi *bounding box* beserta kelas objek pada setiap sel grid. Keunggulan utama YOLO terletak pada kecepatannya dan kemampuannya mengenali berbagai objek secara bersamaan.

Dalam penelitian ini, YOLO digunakan sebagai dasar pengenalan gerakan tangan untuk mendeteksi alfabet Bahasa Isyarat Indonesia (BISINDO) secara waktu nyata melalui kamera.




**Gambar 2. 2** Alur Kerja YOLO

### 2.2.9. YOLOv11

YOLOv11 merupakan versi terbaru dari algoritma *You Only Look Once* yang dikembangkan oleh Ultralytics pada tahun 2024 sebagai penyempurnaan dari YOLOv8. Versi ini mengadopsi arsitektur *anchor-free* dengan peningkatan pada efisiensi deteksi dan akurasi tinggi dalam pengenalan objek secara *real-time*. YOLOv11 menggunakan kombinasi *C2f-ELAN backbone* dan *SPPF+* untuk mempercepat proses ekstraksi fitur, serta mendukung *multi-task learning* seperti deteksi, segmentasi, dan klasifikasi dalam satu model. Selain itu, integrasinya dengan TensorRT dan ONNX Runtime membuat proses inferensi lebih cepat dan ringan (Ultralytics, 2024).

Dalam penelitian ini, YOLOv11 digunakan untuk mendeteksi gerakan tangan pengguna sebagai representasi huruf BISINDO secara langsung, yang kemudian ditampilkan pada website edukasi interaktif.

### 2.2.10. Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) merupakan jenis jaringan saraf tiruan yang dirancang khusus untuk mengenali pola visual pada citra. CNN bekerja dengan mengekstraksi fitur-fitur penting melalui proses *convolution*, *pooling*, dan *fully connected layer* untuk menghasilkan klasifikasi atau deteksi objek. Keunggulan CNN terletak pada kemampuannya mendeteksi fitur secara otomatis tanpa memerlukan ekstraksi manual, sehingga banyak digunakan pada bidang *computer vision* seperti pengenalan wajah, deteksi objek, dan bahasa isyarat (Maya Kinanti Putri & Fauzan Rozi, 2024).

Dalam penelitian ini, CNN menjadi dasar arsitektur algoritma YOLOv11 yang digunakan untuk mendeteksi gestur tangan BISINDO secara akurat dan *real-time*.

### 2.2.11. Google Collabolatory

Google Colaboratory, atau biasa disebut Google Colab, adalah layanan berbasis komputasi awan (*cloud computing*) dari Google yang memungkinkan pengguna menulis serta menjalankan kode Python langsung melalui peramban web. Platform ini dikembangkan berdasarkan Jupyter Notebook, sehingga memfasilitasi penggabungan antara penulisan teks menggunakan markdown, visualisasi data, dan eksekusi kode dalam satu tampilan yang terintegrasi.

Google Colab merupakan platform yang sangat bermanfaat dalam penelitian dan pengembangan yang melibatkan *machine learning*, *data science*, dan *natural language processing*, karena menyediakan akses ke sumber daya komputasi tinggi seperti GPU (*Graphics Processing Unit*) dan TPU (*Tensor Processing Unit*) secara gratis. Hal ini menjadikan Google Colab sebagai alternatif praktis bagi peneliti dan mahasiswa yang tidak memiliki perangkat keras dengan spesifikasi tinggi (Gujjar J & Kumar V, 2021).

Kemampuan berbagi notebook secara daring juga mendukung kolaborasi tim dalam eksperimen dan pengembangan model secara efisien

### 2.2.12. Blackbox Testing

*Black box testing* adalah metode pengujian perangkat lunak yang menitikberatkan pada pengujian fungsi sistem tanpa memperhatikan struktur

internal atau kode sumbernya. Tujuan dari pengujian ini adalah untuk memverifikasi bahwa setiap fitur dalam sistem beroperasi sesuai dengan spesifikasi yang ditetapkan serta menghasilkan output yang tepat berdasarkan input yang dimasukkan. Dengan demikian, metode ini berorientasi pada hasil keluaran (output) dari sistem, bukan pada cara sistem tersebut memproses data di dalamnya (Nur Ichsanudin et al., 2022).

Menurut Widiastiwi et al., (2020), pengujian *black box* menilai perangkat lunak dari perspektif pengguna, di mana penguji hanya mengetahui input dan output tanpa memahami bagaimana proses internal terjadi. Pendekatan ini digunakan untuk memverifikasi apakah sistem telah memenuhi kebutuhan fungsional yang ditentukan berdasarkan spesifikasi tanpa mempertimbangkan struktur kode atau algoritma yang digunakan di dalamnya.

### *2.2.13. Confusion Matrix*

*Confusion Matrix* merupakan metode evaluasi yang digunakan untuk menganalisis kinerja model klasifikasi dengan membandingkan hasil prediksi terhadap label aktual pada data uji. Matriks ini disajikan dalam bentuk tabel yang menunjukkan hubungan antara kelas sebenarnya dan kelas hasil prediksi, sehingga dapat digunakan untuk melihat tingkat ketepatan model serta pola kesalahan klasifikasi yang terjadi.

Dalam penerapan deteksi objek multi-kelas, seperti pada algoritma YOLO, *Confusion Matrix* digunakan untuk menganalisis kemampuan model dalam mengenali setiap kelas objek serta kesalahan prediksi antar kelas. Nilai pada diagonal utama matriks menunjukkan prediksi yang benar, sedangkan nilai di luar diagonal menunjukkan kesalahan klasifikasi, baik berupa objek yang terdeteksi sebagai kelas lain maupun objek yang tidak terdeteksi.

Penggunaan *Confusion Matrix* dalam penelitian ini bertujuan untuk memberikan gambaran visual mengenai performa model dalam mengenali setiap kelas Bahasa Isyarat Indonesia (BISINDO) serta membantu peneliti memahami jenis kesalahan prediksi yang masih terjadi. *Confusion Matrix* digunakan sebagai alat analisis pendukung dan dikombinasikan dengan metrik evaluasi lain untuk memperoleh penilaian performa model yang lebih komprehensif (Utomo et al., 2024).

Dalam *Confusion Matrix*, evaluasi kinerja model dilakukan berdasarkan empat komponen utama, yaitu *True Positive* (TP), *False Positive* (FP), *False Negative* (FN), dan *True Negative* (TN). TP menunjukkan jumlah prediksi yang benar untuk suatu kelas, FP menunjukkan kesalahan prediksi ketika model mendeteksi kelas yang tidak sesuai, sedangkan FN menunjukkan kondisi ketika objek seharusnya terdeteksi namun gagal dikenali oleh model. Sementara itu, TN menggambarkan prediksi benar terhadap kelas negatif, namun pada sistem deteksi objek multi-kelas seperti YOLO, nilai TN tidak digunakan karena area latar belakang yang sangat besar dan tidak terdefinisi secara spesifik sebagai kelas negatif, sehingga tidak relevan untuk evaluasi performa (Kusuma et al., 2023).

### 2.2.14. Metode Evaluasi *USE Questionnaire*

USE (*Usefulness, Satisfaction, Ease of use, and Ease of learning*) Questionnaire merupakan metode evaluasi yang digunakan untuk mengukur tingkat *usability* atau kegunaan suatu sistem dari perspektif pengguna. Metode ini dikembangkan oleh Arnold M. Lund dan telah banyak digunakan dalam penelitian untuk menilai sejauh mana sistem informasi memenuhi kebutuhan pengguna dalam hal kemanfaatan, kemudahan penggunaan, kemudahan dipelajari, dan kepuasan.

Dalam penerapannya, kuesioner USE biasanya terdiri dari beberapa pernyataan yang diukur menggunakan skala Likert 1–5, mulai dari "Sangat Tidak Setuju" hingga "Sangat Setuju." Hasil dari kuesioner tersebut diolah untuk memperoleh skor *usability* berdasarkan persamaan berikut, sehingga dapat diketahui tingkat kegunaan, kemudahan, dan kepuasan pengguna terhadap sistem yang diuji. Dengan demikian, metode ini membantu peneliti menilai kualitas pengalaman pengguna secara objektif dan terukur.

$$Score\ Total = \frac{Skor\ Aktual}{Skor\ Ideal} \times 100\% \quad (1)$$

Keterangan :

Skor Total : Hasil Perhitungan

Skor Aktual : Total Skor Yang Didapatkan

Skor Ideal : Total Skor Yang Diharapkan

Skor aktual diperoleh melalui hasil perhitungan seluruh pendapat responden sesuai dengan klasifikasi bobot yang diberikan. Sementara skor ideal didapatkan dengan mengalikan jumlah responden dengan bobot tertinggi.

Nilai persentase usability yang diperoleh kemudian diinterpretasikan berdasarkan kriteria tertentu. Persentase sebesar 0% hingga 20% diinterpretasikan sebagai kategori tidak baik, persentase 21% hingga 40% termasuk dalam kategori kurang baik, persentase 41% hingga 60% dikategorikan cukup baik, persentase 61% hingga 80% menunjukkan kategori baik, sedangkan persentase 81% hingga 100% termasuk dalam kategori sangat baik. Kriteria interpretasi ini digunakan sebagai acuan dalam menilai kualitas pengalaman pengguna terhadap sistem yang diuji (Timotius et al., 2024).

Dengan demikian, metode *USE Questionnaire* memberikan gambaran yang menyeluruh mengenai pengalaman pengguna, sehingga dapat menjadi dasar dalam melakukan perbaikan dan pengembangan sistem agar lebih sesuai dengan kebutuhan pengguna.

### 2.2.15. Roboflow

Roboflow merupakan sebuah platform yang menyediakan layanan komprehensif untuk pengelolaan, anotasi, pelatihan, dan distribusi dataset yang digunakan dalam pengembangan model *computer vision*. Platform ini banyak digunakan oleh peneliti dan pengembang untuk mempercepat proses *machine learning pipeline*, khususnya dalam bidang *object detection*, *image classification*, dan *segmentation*.

Roboflow memungkinkan pengguna untuk mengunggah dataset citra, melakukan proses *labeling* atau pelabelan objek secara manual maupun semi-otomatis, serta mengonversi dataset tersebut ke berbagai format yang kompatibel dengan model-model *deep learning* populer seperti YOLO, TensorFlow, dan PyTorch. Selain itu, Roboflow juga menyediakan fitur *data augmentation* yang berfungsi untuk memperbanyak variasi data melalui transformasi citra (misalnya rotasi, pemotongan, perubahan pencahayaan, dan flipping), sehingga model dapat dilatih dengan data yang lebih beragam dan memiliki kemampuan generalisasi yang lebih baik (Kumawat et al., 2025).

Dalam penelitian ini, Roboflow digunakan sebagai sumber sekaligus pengelola dataset citra tangan yang merepresentasikan berbagai gerakan Bahasa Isyarat Indonesia (BISINDO).

### 2.2.16. *Precision*, *Recall*, dan mAP50

Dalam sistem deteksi objek berbasis deep learning menggunakan algoritma YOLO, evaluasi kinerja model dilakukan dengan menggunakan metrik *Precision*, *Recall*, dan Mean Average *Precision* (mAP). *Precision* mengukur tingkat ketepatan model dalam menghasilkan prediksi yang benar, yaitu perbandingan antara jumlah prediksi benar terhadap seluruh prediksi yang dihasilkan oleh model. Secara matematis, *Precision* dirumuskan sebagai berikut:

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

Keterangam :

TP (True Positive) : jumlah objek yang berhasil dideteksi dengan benar oleh model.

FP (False Positive) : jumlah objek yang terdeteksi oleh model, tetapi tidak sesuai dengan objek sebenarnya.

Sementara itu, *Recall* menunjukkan kemampuan model dalam mendeteksi seluruh objek yang seharusnya dikenali, yaitu perbandingan antara jumlah prediksi benar terhadap seluruh objek yang benar-benar ada pada data uji. Rumus *Recall* dinyatakan sebagai berikut:

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

Keterangan :

TP (True Positive) : jumlah objek yang berhasil dideteksi dengan benar oleh model.

FN (False Negative) : jumlah objek yang terdeteksi oleh model, tetapi tidak sesuai dengan objek sebenarnya.

Mean Average *Precision* (mAP) merupakan metrik utama dalam evaluasi deteksi objek yang mengombinasikan *Precision* dan *Recall* untuk seluruh kelas objek. Pada penelitian ini digunakan mAP@50, yaitu nilai mAP dengan ambang Intersection over Union (IoU) sebesar 0,5. Suatu deteksi dianggap benar apabila

nilai IoU antara bounding box hasil prediksi dan ground truth memenuhi atau melebihi ambang tersebut. Secara umum, Average *Precision* (AP) diperoleh dari luas area di bawah kurva *Precision–Recall* untuk setiap kelas, sedangkan mAP dihitung sebagai rata-rata nilai AP dari seluruh kelas objek, yang dirumuskan sebagai berikut:

$$\text{mAP@50} = \frac{1}{N}\sum_{i=1}^{N} A\,P_i \tag{4}$$

Keterangan :

mAP@50 : nilai rata-rata *Average Precision* seluruh kelas objek dengan ambang *Intersection over Union* (IoU) sebesar 0,5.

$AP_i$ : nilai *Average Precision* pada kelas ke-$i$, yang diperoleh dari luas area di bawah kurva *Precision–Recall*.

**N** : jumlah total kelas objek yang digunakan dalam pengujian.

di mana $AP_i$ merupakan nilai Average *Precision* pada kelas ke-$i$, dan $N$adalah jumlah total kelas objek yang digunakan. Nilai *Precision*, *Recall*, dan mAP@50 berada pada rentang 0 hingga 1, dengan nilai yang semakin mendekati 1 menunjukkan performa deteksi model yang semakin baik (Padilla et al., 2021).