

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Penelitian ini akan mencakup tinjauan terhadap studi-studi sebelumnya yang berkaitan dengan topik deteksi objek dengan menerapkan metode 3C2S pada Tabel 2.1. Hal ini bertujuan agar penulis dapat memahami secara lebih mendalam mengenai deteksi objek dalam konteks bahasa isyarat dengan penerapan model YOLOv8. Penelitian ini merujuk pada penelitian-penelitian terdahulu yang relevan dengan topik ini.

Penelitian terdahulu yang pertama berjudul “Penerapan Algoritme *You Only Look Once Version 8* Untuk Identifikasi Abjad Bahasa Isyarat Indonesia”, yang dilakukan oleh Agung Ma’ruf dan Mardi Hardjianto pada tahun 2023. Penelitian ini bertujuan untuk meningkatkan aksesibilitas komunikasi bagi individu dengan gangguan pendengaran melalui penggunaan teknologi deteksi dan klasifikasi bahasa isyarat secara *real-time*. Data yang digunakan adalah dataset abjad BISINDO. Dataset terbentuk dari data primer 25 relawan dan data sekunder Kaggle, mencakup 11.469 citra setelah proses augmentasi. Hasil dari penelitian ini adalah sistem deteksi bahasa isyarat BISINDO dengan menggunakan model YOLOv8. Model yang dilatih dan dievaluasi mencapai tingkat akurasi tinggi, terutama pada intensitas cahaya di atas 50 lux dan jarak kurang dari 100 cm. Sistem ini mampu memberikan output teks dan suara secara *real-time* [22].

Penelitian selanjutnya berjudul “*American Sign Language Detection using YOLOv5 and YOLOv8*”, yang dilakukan oleh Shobhit Tyagi, Prashant Upadhyay, Hoor Fatima, Sachin Jain, dan Avinash Kumar Sharma pada tahun 2023. Penelitian ini bertujuan untuk membandingkan kinerja YOLOv5 dan YOLOv8 dalam mendeteksi bahasa isyarat Amerika (ASL), serta menyoroti tantangan pengenalan bahasa isyarat dan hasil dari kedua algoritma YOLO dalam presisi dan mAP. Data yang digunakan adalah “*American Sign Language Letters Dataset v1*” dan “*Two-handed ISL dataset for comparing*”

machine learning classification models" dengan akurasi 92%. Selain itu, dataset dari *Roboflow* juga digunakan untuk pengujian, pelatihan, dan validasi dalam rasio 1:21:2. Hasil dari penelitian ini adalah penggunaan algoritma YOLOv5 dan YOLOv8 untuk deteksi objek pengenalan Bahasa Isyarat Amerika. Secara keseluruhan, YOLOv8 lebih akurat dan cepat dalam mendeteksi objek, sedangkan YOLOv5 memiliki akurasi klasifikasi yang lebih baik [10].

Penelitian berikutnya berjudul "*Hand Symbol Classification for Human-Computer Interaction Using the Fifth Version of YOLO Object Detection*", yang dilakukan oleh Sugiarto Wibowo dan Indar Sugiarto pada tahun 2023. Penelitian ini bertujuan mengklasifikasikan simbol tangan secara *real-time* untuk implementasi HCI. Data yang digunakan adalah dataset simbol tangan, terdiri dari 500 gambar yang mencakup lima kelas, yaitu *ok*, *cancel*, *previous*, *next*, dan *confirm*. Hasil dari penelitian ini adalah pembuatan dan evaluasi model YOLOv5 dengan ukuran *medium* (YOLOv5m) untuk mengklasifikasikan simbol tangan secara *real time* dengan tingkat akurasi sebesar 80% berdasarkan rata-rata dari semua kelas [8].

Penelitian lain yang berjudul "*Hand Gesture Recognition Based on Various Deep Learning YOLO Models*", yang dilakukan oleh Soukaina Chraa Mesbahi, Mohamed Adnane Mahraz, Jamal Riffi, dan Hamid Tairi pada tahun 2023. Penelitian ini bertujuan mengatasi hambatan komunikasi pada individu tunarungu atau mereka dengan gangguan pendengaran, sehingga dapat meningkatkan efektivitas komunikasi mereka dengan orang lain. Data yang digunakan adalah dataset gestur tangan bahasa isyarat yang mencakup total 3600 gambar dengan klasifikasi pada tujuh label atau kelas, antara lain *Fist*, *I*, *Pointer*, *Ok*, *Palm*, *Thumb down*, dan *Thumb up*. Hasil dari penelitian ini adalah sebuah sistem terjemahan otomatis bahasa isyarat yang memungkinkan komputer menafsirkan dan mengubah bahasa isyarat menjadi teks dengan menggunakan model YOLOv3, YOLOv4, YOLOv4-tiny, dan YOLOv5. Dari ketiga versi YOLO yang digunakan, YOLOv5 mencapai

tingkat akurasi tertinggi pada IoU 0,5, yakni mencapai 98,47%, diikuti oleh YOLOv4 dan YOLOv3 dengan nilai akurasi berturut-turut sebesar 98,22% dan 97,15% [2].

Penelitian selanjutnya berjudul “*Indonesian Sign Language Recognition using YOLO Method*”, yang dilakukan oleh Steve Daniels, Nanik Suciati, dan Chastine Fathichah pada tahun 2021. Penelitian ini bertujuan mengembangkan sistem pengenalan bahasa isyarat dengan metode YOLO untuk mengenali Bahasa Isyarat Indonesia (BISINDO) dari data video secara *real-time*. Data yang digunakan adalah dataset bahasa isyarat yang terdiri dari 24 kelas, dan proses pelatihan memanfaatkan 80% dataset. Hasil dari penelitian ini adalah sebuah sistem pengenalan Bahasa Isyarat Indonesia (BISINDO) menggunakan metode YOLO. Dalam eksperimen pada data gambar, sistem mencapai akurasi 100%, sementara pada data video mencapai akurasi 72,97%, dengan kecepatan 8 fps [4].

Penelitian berikutnya berjudul “*Pengembangan Aplikasi Bahasa Isyarat Indonesia Berbasis Realtime Video Menggunakan Model Machine Learning*”, yang dilakukan oleh Al Muhtadi Ambarak dan Achmad Zakki Falani pada tahun 2023. Penelitian ini bertujuan mengembangkan aplikasi AI untuk memfasilitasi komunikasi komunitas tunarungu dengan fokus pada pengenalan bahasa isyarat dan mengurangi hambatan komunikasi, sehingga memudahkan interaksi mereka dengan orang di sekitar. Data yang digunakan adalah dataset BISINDO yang berjumlah 406 gambar yang dibagi ke dalam empat kelas, yakni Saya, Kamu, Mau, dan Kemana. Hasil dari penelitian ini adalah suatu aplikasi *android* dengan tingkat akurasi sebesar 77%, yang mampu mendeteksi empat kata dalam Bahasa Isyarat Indonesia (BISINDO) dengan menggunakan model EfficientDet-lite [7].

Penelitian lain yang berjudul “*Indonesian Sign Language Image Detection Using Convolutional Neural Network (CNN) Method*”, yang dilakukan oleh Andreas Nugroho Sihananto, Erista Maya Safitri, Yoga Maulana, Fikri Fakhruddin, dan Mochammad Ervinda Yudistira pada tahun 2023. Penelitian ini bertujuan untuk menerjemahkan bahasa isyarat dengan

akurasi yang baik menggunakan algoritma *Convolutional Neural Network* (CNN), sehingga dapat memperbaiki komunikasi antara individu tuli dan non-tuli di Indonesia. Data yang digunakan adalah dataset SIBI dan BISINDO. SIBI memiliki 468 file data uji dan 26 file data latih, sedangkan BISINDO memiliki 312 file data uji dan 26 file data latih. Hasil dari penelitian ini adalah deteksi alfabet bahasa isyarat dalam dataset SIBI dan BISINDO menggunakan model CNN, dimana akurasi pengenalan lebih tinggi pada dataset SIBI (93,29%) dibandingkan BISINDO (82,32%) [20].

Penelitian lainnya yang berjudul “*Real-Time Flying Object Detection with YOLOv8*”, yang dilakukan oleh Dillon Reis, Jordan Kupec, Jacqueline Hong, dan Ahmad Daoudi pada tahun 2023. Penelitian ini bertujuan mengevaluasi kinerja model dalam hal *mean average precision* (mAP) dan kecepatan inferensi, serta untuk mengatasi tantangan dalam deteksi objek *real-time* dan memotivasi pengembangan model deteksi objek terbang. Data yang digunakan adalah dataset objek terbang, yang berisi 40 kelas dan kemudian dilakukan *transfer learning* pada dataset yang lebih representatif dari lingkungan dunia nyata. Hasil dari penelitian ini adalah sebuah model untuk deteksi objek terbang secara *real-time* menggunakan YOLOv8. Model ini mencapai *mean average precision* (mAP)₅₀₋₉₅ sebesar 0,685 dan kecepatan inferensi rata-rata sebesar 50 fps untuk video 1080p [11].

Penelitian yang lain dengan judul “*Parking Time Violation Tracking Using YOLOv8 and Tracking Algorithms*”, yang dilakukan oleh Nabin Sharma, Sushish Baral, May Phu Paing, dan Rathachai Chawuthai pada tahun 2023. Penelitian ini bertujuan untuk mengembangkan algoritma yang efektif dalam melacak pelanggaran waktu parkir menggunakan teknologi CCTV, model *Deep Learning*, dan algoritma pelacakan objek. Data yang digunakan adalah dataset rekaman video kamera CCTV di *King Mongkut's Institute of Technology, Ladkrabang* (KMITL). Kamera CCTV yang digunakan adalah seri Panasonic V dengan tingkat tangkapan 15-30 FPS dan resolusi 1080P. Data dikumpulkan menggunakan 4 kamera yang ditempatkan di lokasi berbeda di dalam kampus universitas, dengan resolusi 1080P dan *frame rate*

15 FPS. Pemilihan lokasi dataset mencakup beragam sudut pandang kamera dan kondisi cuaca. Video dalam dataset mencakup berbagai pengaturan pencahayaan dan cuaca, memberikan representasi komprehensif dari skenario tempat parkir di KMITL. Hasil dari penelitian ini adalah sebuah algoritma pelacakan pelanggaran waktu parkir menggunakan YOLOv8 untuk deteksi kendaraan dan DeepSORT/OC-SORT untuk pelacakan. Algoritma ini efektif dalam melacak pelanggaran waktu parkir dan menunjukkan akurasi yang tinggi [21].

Penelitian terakhir yang berjudul “*Comparison of YOLO (V3,V5) and MobileNet-SSD (V1,V2) for Person Identification Using Ear-Biometrics*”, yang dilakukan oleh Shahadat Hossain, Humaira Anzum, dan Shamim Akhter pada tahun 2023. Penelitian ini bertujuan untuk membandingkan kinerja algoritma *deep learning* YOLO (V3, V5) dan MobileNet-SSD (V1, V2) dalam mengidentifikasi individu berdasarkan biometrik telinga serta mengevaluasi akurasi dan efisiensinya. Data yang digunakan adalah EarVN1.0 Dataset, yang terdiri dari 27.592 gambar telinga dari 164 individu. Dari dataset tersebut, dipilih secara acak 10 individu dengan total 2057 gambar telinga. Sebanyak 85% dari gambar tersebut digunakan untuk pelatihan, 5% untuk validasi, dan 10% untuk pengujian. Hasil dari penelitian ini adalah membandingkan kinerja algoritma *deep learning* YOLO (V3, V5) dan MobileNet-SSD (V1, V2) untuk identifikasi orang menggunakan biometrik telinga, dengan YOLOV5 mencapai nilai akurasi tertinggi [14].

Berdasarkan Tabel 2.1, terdapat kemiripan dan perbedaan dari penelitian yang akan dilakukan penulis pada penelitian selanjutnya. Kemiripan antara penelitian sebelumnya dengan penelitian yang akan dilakukan selanjutnya terletak pada topik *machine learning* serta metode atau model yang digunakan untuk solusi dalam deteksi objek. Kemudian, untuk perbedaan antara penelitian terdahulu dengan penelitian yang akan dilakukan peneliti selanjutnya adalah objek penelitian, waktu penelitian, dan studi kasus penelitian.

Tabel 2.1 Penelitian Terdahulu

No	Judul	<i>Comparing</i>	<i>Contrasting</i>	<i>Criticize</i>	<i>Synthesize</i>	<i>Summarize</i>
1	Penerapan Algoritme <i>You Only Look Once Version 8</i> Untuk Identifikasi Abjad Bahasa Isyarat Indonesia [22]. (2023)	Penelitian ini menggunakan model YOLOv8 untuk deteksi bahasa isyarat BISINDO. Evaluasi dilakukan untuk mengukur tingkat presisi dan <i>recall</i> dari model deteksi bahasa isyarat BISINDO.	Evaluasi dilakukan untuk mengukur tingkat akurasi, <i>precision</i> , dan <i>recall</i> dari model deteksi bahasa isyarat BISINDO. Sedangkan penelitian yang sedang dilakukan mengukur nilai metrik <i>precision</i> , <i>recall</i> , <i>F1-Score</i> , dan <i>mAP⁵⁰</i> .	Penelitian ini dapat ditingkatkan dengan memperluas cakupan pengujian performa model deteksi, seperti pengujian pada berbagai kondisi cahaya dan jarak yang lebih jauh. Selain itu, penelitian ini juga dapat mempertimbangkan penggunaan metode validasi silang untuk memastikan keandalan dan generalisasi model deteksi bahasa isyarat BISINDO.	Penelitian ini bertujuan untuk meningkatkan aksesibilitas komunikasi bagi individu dengan gangguan pendengaran melalui penggunaan teknologi deteksi dan klasifikasi bahasa isyarat secara <i>real-time</i> .	Penelitian ini menghasilkan sistem deteksi bahasa isyarat BISINDO dengan menggunakan metode YOLOv8. Dataset terbentuk dari data primer 25 relawan dan data sekunder Kaggle, mencakup 11.469 citra setelah proses augmentasi. Model <i>machine learning</i> dilatih dan dievaluasi menggunakan YOLOv8, mencapai tingkat akurasi tinggi,

No	Judul	<i>Comparing</i>	<i>Contrasting</i>	<i>Criticize</i>	<i>Synthesize</i>	<i>Summarize</i>
						terutama pada intensitas cahaya di atas 50 lux dan jarak kurang dari 100 cm. Sistem ini mampu memberikan output teks dan suara secara <i>real-time</i> .
2	<i>American Sign Language Detection using YOLOv5 and YOLOv8</i> [10]. (2023)	Penelitian terdahulu ini menggunakan model <i>You Only Look Once</i> (YOLO).	Penelitian ini membandingkan model YOLOv5 dan YOLOv8 untuk mendeteksi bahasa isyarat Amerika (ASL). Sedangkan penelitian yang sedang dilakukan hanya menggunakan model YOLOv8 saja untuk mendeteksi Bahasa Isyarat	Kritik terhadap penelitian ini mencakup kurangnya analisis mendalam tentang faktor-faktor yang mempengaruhi kinerja kedua algoritma, seperti ukuran dataset, kompleksitas model, dan parameter pelatihan. Selain itu, penelitian ini juga dapat diperkaya dengan	Penelitian ini bertujuan untuk membandingkan kinerja YOLOv5 dan YOLOv8 dalam mendeteksi bahasa isyarat Amerika (ASL). Serta menyoroti tantangan pengenalan bahasa isyarat dan hasil dari kedua algoritma YOLO dalam presisi dan mAP.	Perbandingan kinerja kedua algoritma menunjukkan bahwa YOLOv8 memberikan hasil yang lebih baik dalam presisi dan mAP, sementara YOLOv5 memiliki nilai <i>recall</i> yang lebih tinggi.

No	Judul	<i>Comparing</i>	<i>Contrasting</i>	<i>Criticize</i>	<i>Synthesize</i>	<i>Summarize</i>
			Indonesia (BISINDO).	mempertimbangkan pendekatan lain dalam deteksi bahasa isyarat dan membandingkan dengan YOLOv5 dan YOLOv8.		
3	<i>Hand Symbol Classification for Human-Computer Interaction Using the Fifth Version of YOLO Object Detection</i> [8]. (2023)	Penelitian terdahulu ini mengadopsi model algoritma <i>You Only Look Once</i> (YOLO) untuk mengklasifikasikan simbol tangan secara <i>real time</i> .	Penelitian ini menggunakan model YOLOv5 dengan ukuran <i>medium</i> (YOLOv5m) untuk deteksi simbol tangan. Sedangkan penelitian yang sedang dilakukan menggunakan model YOLOv8 dengan ukuran <i>small</i> (YOLOv8s) dan <i>medium</i> (YOLOv8m).	Kritik terhadap penelitian ini mencakup terbatasnya keragaman data, dimana dataset simbol tangan hanya terdiri dari 500 gambar yang mencakup lima kelas, yaitu <i>ok</i> , <i>cancel</i> , <i>previous</i> , <i>next</i> , dan <i>confirm</i> . Selain itu, dataset hanya berisi karakteristik tangan para peneliti, sedangkan berbagai karakteristik tangan harus ditambahkan	Penelitian ini berfokus pada pembuatan dan evaluasi model pendeteksian objek untuk mengklasifikasikan simbol tangan secara <i>real time</i> dan dapat diimplementasikan untuk tujuan HCI.	Selama proses pelatihan dataset, sejumlah <i>hyperparameter</i> telah disesuaikan, termasuk ukuran gambar menjadi 416px, ukuran <i>batch</i> sebesar empat, dan jumlah <i>epochs</i> mencapai 300 iterasi. Proses pelatihan menyita waktu selama sekitar 4 jam 24 menit. Setelah proses pelatihan selesai, model dinilai, dan hasil

No	Judul	Comparing	Contrasting	Criticize	Synthesize	Summarize
				untuk membuat model yang lebih kuat dan akurat.		evaluasi menunjukkan tingkat akurasi sebesar 80% berdasarkan rata-rata dari semua kelas.
4	<i>Hand Gesture Recognition Based on Various Deep Learning YOLO Models</i> [2]. (2023)	Penelitian terdahulu ini melakukan deteksi objek dengan menggunakan model <i>deep learning</i> , yaitu beberapa model algoritma YOLO untuk deteksi objek dan pengenalan gerakan tangan.	Penelitian ini membandingkan model YOLOv3, YOLOv4, YOLOv4-tiny, dan YOLOv5 untuk mendeteksi gestur tangan.	Terkait dengan proses pelatihan, penelitian ini tidak memberikan informasi mengenai jumlah iterasi <i>epochs</i> yang digunakan.	Penelitian ini dilakukan dengan tujuan untuk mengatasi hambatan berkomunikasi yang dihadapi oleh individu tunarungu atau mereka yang mengalami gangguan pendengaran agar dapat memudahkan mereka untuk berkomunikasi dengan orang lain secara lebih efektif.	Hasil dari penelitian ini adalah sebuah sistem terjemahan otomatis bahasa isyarat yang memungkinkan komputer menafsirkan dan mengubah bahasa isyarat menjadi teks. Ukuran gambar pada tahap pelatihan adalah 416px dengan rasio 1:1. Dari ketiga versi YOLO yang digunakan,

No	Judul	Comparing	Contrasting	Criticize	Synthesize	Summarize
						YOLOv5 mencapai tingkat akurasi tertinggi pada IoU 0,5, yakni mencapai 98,47%, diikuti oleh YOLOv4 dan YOLOv3 dengan nilai akurasi berturut-turut sebesar 98,22% dan 97,15%.
5	<i>Indonesian Sign Language Recognition using YOLO Method</i> [4]. (2021)	Penelitian Pengenalan Bahasa Isyarat Indonesia (BISINDO) ini memanfaatkan metode <i>You Only Look Once</i> (YOLO) untuk deteksi objek secara <i>real-time</i> berbasis <i>Convolutional Neural Network</i> (CNN).	Penelitian ini menggunakan model YOLOv3 untuk deteksi objek. Sedangkan penelitian yang sedang dilakukan menggunakan model YOLOv8.	Meskipun hasil pada penelitian ini menunjukkan akurasi yang tinggi pada data gambar, akurasi pada data video masih perlu ditingkatkan. Kecepatan pemrosesan sistem juga belum mencapai <i>real-time</i> untuk input video. Pada penelitian ini	Penelitian ini bertujuan untuk mengembangkan sistem pengenalan bahasa isyarat dengan metode YOLO untuk mengenali Bahasa Isyarat Indonesia (BISINDO) dari data video secara <i>real-time</i> .	Hasil dari penelitian ini adalah sebuah sistem pengenalan Bahasa Isyarat Indonesia (BISINDO) dengan metode YOLO. Percobaan pada data gambar mencapai akurasi 100%, sedangkan

No	Judul	<i>Comparing</i>	<i>Contrasting</i>	<i>Criticize</i>	<i>Synthesize</i>	<i>Summarize</i>
				diperlukan suatu algoritma yang dapat memisahkan <i>frame</i> transisi dari <i>frame</i> isyarat sebagai upaya untuk meningkatkan tingkat akurasi.		data video mencapai akurasi 72,97%, dengan kecepatan 8 fps. Dataset terdiri dari 24 kelas bahasa isyarat, dan proses pelatihan memanfaatkan 80% dataset. Pelatihan dilakukan menggunakan <i>darknet53</i> yang telah dilatih sebelumnya di <i>ImageNet</i> , dengan <i>learning rate</i> yang bervariasi.
6	Pengembangan Aplikasi Bahasa Isyarat Indonesia Berbasis <i>Realtime</i> Video	Penelitian terdahulu ini mengembangkan sistem aplikasi Bahasa Isyarat Indonesia berbasis	Penelitian ini menggunakan model EfficientDet-lite dalam mengembangkan	Pada penelitian ini tidak diberikan metrik spesifik atau analisis komparatif dengan metode yang ada, yang	Penelitian ini bertujuan untuk mengembangkan aplikasi berbasis AI untuk memfasilitasi	Hasil dari penelitian ini adalah suatu aplikasi <i>android</i> dengan tingkat akurasi 77%,

No	Judul	Comparing	Contrasting	Criticize	Synthesize	Summarize
	Menggunakan Model <i>Machine Learning</i> [7]. (2023)	<i>realtime</i> video.	sistem aplikasi. Sedangkan penelitian yang sedang dilakukan menggunakan model YOLOv8.	dapat memperkuat validitas temuan.	komunikasi bagi komunitas tunarungu, dengan fokus pada pengenalan isyarat bahasa isyarat dan mengurangi hambatan komunikasi antara komunitas tunarungu serta membantu kelompok tersebut berkomunikasi dengan orang-orang di sekitarnya.	yang mampu mendeteksi empat kata dalam BISINDO berbasis <i>realtime</i> video. Proses pelatihan pada penelitian ini dilakukan dengan menggunakan arsitektur model <i>pre-trained EfficientDet-lite0</i> , dengan jumlah iterasi <i>epochs</i> mencapai 50.
7	<i>Indonesian Sign Language Image Detection Using Convolutional Neural Network (CNN) Method</i> [20]. (2023)	Penelitian terdahulu ini mendeteksi alfabet bahasa isyarat pada dataset SIBI dan BISINDO.	Penelitian ini menggunakan model <i>Convolutional Neural Network (CNN)</i> . Sedangkan penelitian yang sedang dilakukan	Penelitian ini dapat ditingkatkan dengan menerapkan metode <i>pre-processing</i> untuk mengurangi <i>noise</i> pada dataset BISINDO, serta mempertimbangkan	Penelitian ini bertujuan mengembangkan model menggunakan algoritma <i>Convolutional Neural Network (CNN)</i> untuk menerjemahkan	Penelitian ini menggunakan model CNN untuk mendeteksi alfabet bahasa isyarat dalam dataset SIBI dan BISINDO. Hasil penelitian ini menunjukkan

No	Judul	<i>Comparing</i>	<i>Contrasting</i>	<i>Criticize</i>	<i>Synthesize</i>	<i>Summarize</i>
			menggunakan model YOLOv8.	penggunaan algoritma lain atau modifikasi model CNN untuk meningkatkan akurasi secara keseluruhan.	bahasa isyarat dengan akurasi yang baik, sehingga dapat memperbaiki komunikasi antara individu tuli dan non-tuli di Indonesia.	akurasi pengenalan lebih tinggi pada dataset SIBI (93,29%) dibandingkan BISINDO (82,32%).
8	<i>Real-Time Flying Object Detection with YOLOv8</i> [11]. (2023)	Penelitian terdahulu ini menggunakan model YOLOv8 untuk mendeteksi objek.	Penelitian ini mengembangkan sebuah model untuk mendeteksi objek terbang. Sedangkan penelitian yang sedang dilakukan mengembangkan sebuah model untuk mendeteksi bahasa isyarat Indonesia.	Penelitian ini dapat diperkaya dengan memperluas analisis terhadap kinerja model dalam situasi yang lebih kompleks dan variasi data yang lebih luas. Selain itu, penelitian ini dapat memberikan pemahaman yang lebih mendalam tentang implementasi praktis dari model ini dalam skenario dunia nyata.	Penelitian ini bertujuan untuk mengevaluasi kinerja model dalam hal <i>mean average precision</i> (mAP) dan kecepatan inferensi, serta untuk mengatasi tantangan dalam deteksi objek <i>real-time</i> dan memotivasi pengembangan model deteksi objek terbang.	Penelitian ini menghasilkan sebuah model untuk deteksi objek terbang secara <i>real-time</i> menggunakan YOLOv8. Model ini dilatih pada dataset yang berisi 40 kelas objek terbang dan kemudian dilakukan <i>transfer learning</i> pada dataset yang lebih representatif dari

No	Judul	<i>Comparing</i>	<i>Contrasting</i>	<i>Criticize</i>	<i>Synthesize</i>	<i>Summarize</i>
						lingkungan dunia nyata. Model ini mencapai <i>mean average precision</i> (mAP)50-95 sebesar 0.685 dan kecepatan inferensi rata-rata sebesar 50 fps untuk video 1080p.
9	<i>Parking Time Violation Tracking Using YOLOv8 and Tracking Algorithms</i> [21]. (2023)	Penelitian terdahulu ini menggunakan model <i>Deep Learning</i> , yaitu YOLOv8 untuk mendeteksi.	Penelitian ini menggunakan algoritma DeepSORT/OC-SORT, algoritma ini menggunakan pendekatan <i>frame-by-frame</i> untuk mendeteksi kendaraan, melacaknya, dan memantau pelanggaran waktu parkir	Penelitian ini dapat diperluas dengan mempertimbangkan faktor-faktor lain yang dapat memengaruhi kinerja algoritma, seperti kepadatan lalu lintas dan jenis kendaraan. Selain itu, penelitian ini dapat lebih memperjelas proses pelatihan model deteksi yang	Penelitian ini bertujuan untuk mengembangkan algoritma yang efektif dalam melacak pelanggaran waktu parkir menggunakan teknologi CCTV, model <i>Deep Learning</i> , dan algoritma pelacakan objek. Selain itu,	Penelitian ini menghasilkan sebuah algoritma pelacakan pelanggaran waktu parkir yang menggunakan YOLOv8 untuk deteksi kendaraan dan DeepSORT/OC-SORT untuk pelacakan. Algoritma ini

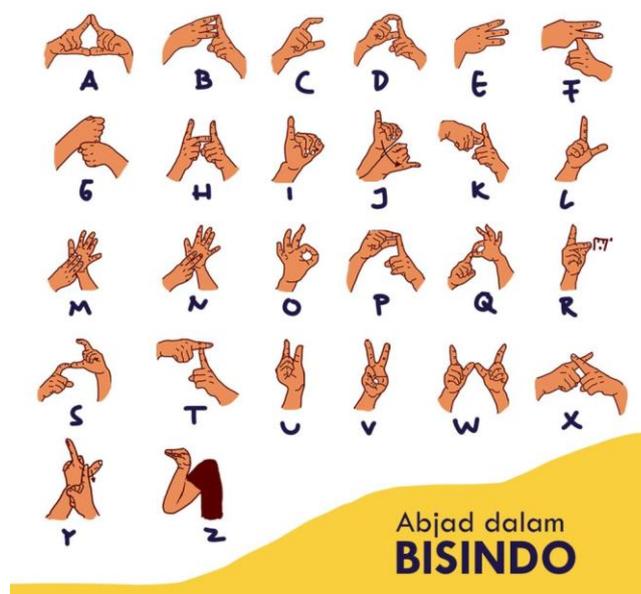
No	Judul	Comparing	Contrasting	Criticize	Synthesize	Summarize
			menggunakan kamera CCTV.	disesuaikan dan mempertimbangkan aspek privasi data yang digunakan.	penelitian ini mengevaluasi kinerja algoritma dalam berbagai kondisi alam untuk memberikan solusi yang akurat dan handal terhadap masalah pelanggaran waktu parkir.	efektif dalam melacak pelanggaran waktu parkir dan menunjukkan akurasi yang tinggi.
10	<i>Comparison of YOLO (V3,V5) and MobileNet-SSD (V1,V2) for Person Identification Using Ear-Biometrics [14]. (2023)</i>	Penelitian terdahulu ini menggunakan model <i>deep learning</i> , salah satunya adalah algoritma <i>You Only Look Once</i> (YOLO).	Penelitian ini membandingkan model YOLO (V3, V5) dan MobileNet-SSD (V1, V2) untuk mengidentifikasi orang menggunakan biometrik telinga. Sedangkan penelitian yang sedang dilakukan hanya menggunakan	Penelitian ini dapat diperkaya dengan mempertimbangkan faktor-faktor lain yang memengaruhi kinerja algoritma, seperti variasi pose dan pencahayaan pada gambar telinga. Selain itu, dapat memperluas analisisnya dengan membandingkan kinerja algoritma pada dataset yang lebih besar dan	Penelitian ini dilakukan dengan tujuan untuk membandingkan kinerja algoritma <i>deep learning</i> YOLO (V3, V5) dan MobileNet-SSD (V1, V2) dalam mengidentifikasi individu berdasarkan biometrik telinga serta mengevaluasi akurasi dan	Hasil dari penelitian ini adalah membandingkan kinerja algoritma <i>deep learning</i> YOLO (V3, V5) dan MobileNet-SSD (V1, V2) untuk identifikasi orang menggunakan biometrik telinga, dengan YOLOV5 mencapai nilai akurasi tertinggi.

No	Judul	<i>Comparing</i>	<i>Contrasting</i>	<i>Criticize</i>	<i>Synthesize</i>	<i>Summarize</i>
			model YOLOv8 saja untuk mendeteksi bahasa isyarat Indonesia.	beragam.	efisiensinya.	

2.2 Landasan Teori

2.2.1 Bahasa Isyarat Indonesia (BISINDO)

Bahasa Isyarat Indonesia atau BISINDO merupakan bentuk bahasa isyarat yang paling umum digunakan oleh komunitas difabel rungu di Indonesia. BISINDO termasuk salah satu dari banyak bahasa isyarat yang berkembang secara alami di seluruh dunia, dan oleh karena itu, mungkin terdapat sedikit perbedaan dalam pola penggunaannya di beberapa daerah atau wilayah [20]. Walaupun Pemerintah Indonesia telah menetapkan Sistem Bahasa Isyarat Indonesia (SIBI) sebagai bahasa isyarat resmi, BISINDO tetap menjadi pilihan utama dalam komunikasi bagi kalangan difabel rungu. BISINDO dihasilkan oleh kalangan difabel rungu sendiri untuk berinteraksi dengan masyarakat umum. Lebih lanjut, bentuk-bentuk dalam BISINDO cenderung lebih sederhana dan mudah dipahami jika dibandingkan dengan SIBI [23]. Terdapat 26 huruf abjad A-Z dan 10 kata isyarat pada Bahasa Isyarat Indonesia (BISINDO) yang digunakan dalam penelitian ini, yaitu antara lain: ayah, halo, kakak, minum, rumah, sama-sama, sehat, teman, terima kasih, dan tidur.



Gambar 2.1 Abjad BISINDO [4]



Gambar 2.2 Sampel BISINDO: (a) Ayah; (b) Kakak; (c) Minum; (d) Rumah [24]

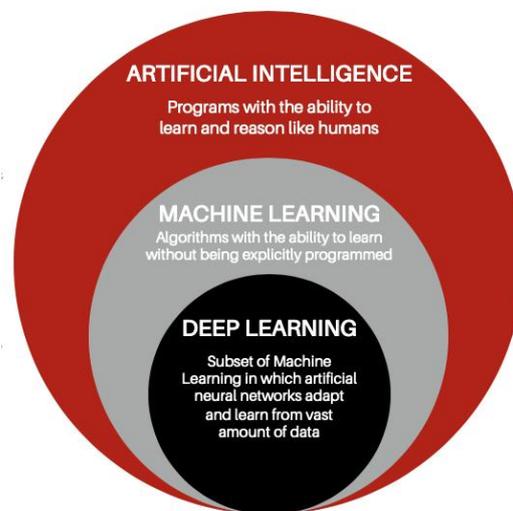
2.2.2 *Artificial Intelligence (AI)*

Kecerdasan buatan atau seringkali disebut dengan *Artificial Intelligence (AI)* menjadi salah satu topik yang trend khususnya dalam perkembangan teknologi di berbagai bidang ilmu, namun dengan ruang lingkup AI yang masih sangat luas sulit ditentukan kejelasan dan kepastian ruang lingkungannya karena berbeda dengan disiplin ilmu lain seperti matematika, fisika, kimia dan biologi yang memiliki batasan ruang lingkup yang jelas. Beberapa pandangan sejumlah ahli, kecerdasan buatan dapat didefinisikan sebagai berikut [25].

1. Buchanan dan Shortliffe (1985) menyatakan bahwa kecerdasan buatan merupakan proses manipulasi simbol untuk menyelesaikan masalah.
2. Staugaard dan Marvin Minsky berpendapat bahwa kecerdasan buatan adalah cabang ilmu pengetahuan yang bertujuan menciptakan mesin yang mampu melakukan tugas sebagaimana yang dilakukan manusia.
3. Schildt (1987) menyatakan bahwa kecerdasan buatan akan menampakkan perilaku yang mirip dengan manusia dalam mengatasi masalah-masalah yang dihadapi manusia.

Definisi lain dari *Artificial Intelligence* yaitu dapat dikatakan sebagai cabang ilmu komputer yang memusatkan perhatiannya pada pengembangan mesin cerdas yang mampu menjalankan tugas-tugas seperti layaknya manusia. Beberapa contoh implementasi AI dalam dunia teknologi melibatkan produk seperti Google Home, Siri, dan Alexa, yang saat ini lebih dikenal sebagai "asisten cerdas" manusia.

Pemakaian istilah "kecerdasan buatan" pertama kali muncul pada tahun 1956, yang diprakarsai oleh sejumlah peneliti, di antaranya Allen Newell dan Herbert A. Simon. Sejak saat itu, industri AI telah mengalami perkembangan yang signifikan [26].

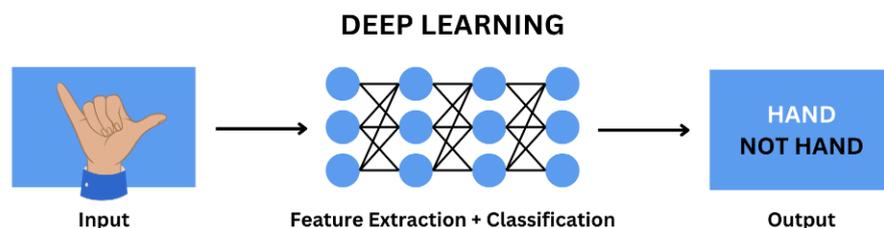


Gambar 2.3 Korelasi antara AI, *Machine Learning*, dan *Deep Learning* [27]

AI, *machine learning*, dan *deep learning* memiliki keterkaitan dan saling ketergantungan. *Machine learning* merupakan komponen dari AI, sedangkan *deep learning* merupakan bagian dari *machine learning*. *Machine learning* merupakan sistem yang dapat memperoleh pengetahuan secara otomatis tanpa perlu pemrograman berulang oleh manusia. Fokus utama *machine learning* adalah penggalian informasi baru dari data input untuk memudahkan pengambilan keputusan manusia. Algoritma *machine learning* mungkin tidak efisien dalam kasus kompleks. Oleh karena itu, *deep learning* memiliki peran khusus dalam mengelola volume data besar dan menangani masalah-masalah kompleks dengan meniru proses berpikir otak manusia. [12].

2.2.3 Deep Learning

Deep learning adalah suatu teknik dalam bidang *machine learning* yang melibatkan penggunaan sejumlah besar lapisan pemrosesan informasi nonlinier untuk melakukan ekstraksi fitur, pengenalan pola, dan klasifikasi [28]. Pendekatan ini menghasilkan arsitektur yang sangat kuat untuk *supervised learning*, di mana penambahan lapisan lebih lanjut memungkinkan model pembelajaran untuk secara lebih efektif merepresentasikan data citra yang berlabel [29]. *Deep learning* menerapkan konsep hirarki untuk menyelesaikan masalah dalam sistem pembelajaran komputer. Melalui penggabungan konsep-konsep yang lebih sederhana, komputer dapat memahami konsep yang kompleks [30].



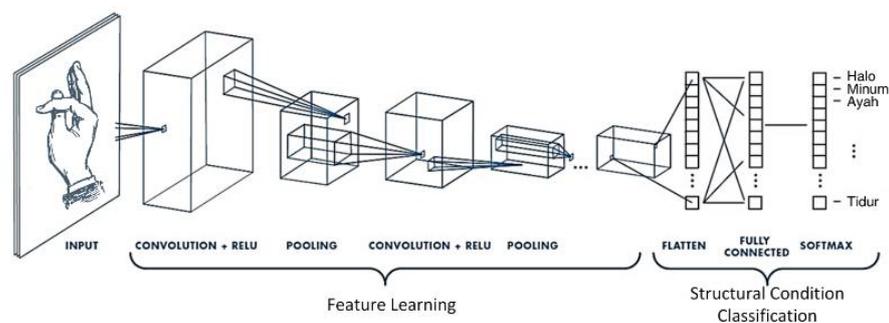
Gambar 2.4 Konsep Kerja *Deep Learning*

Pada Gambar 2.4, terdapat representasi ilustratif mengenai konsep kerja dari *deep learning*. Sistem ini menerima input berupa citra yang selanjutnya akan mengalami proses *feature extraction* dan *classification*. Tahap *feature extraction* mengubah citra menjadi representasi angka, disebut *encoding*, dan melibatkan dua lapisan, yaitu *Convolutional Layer* dan *Pooling Layer* [31].

2.2.4 Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) adalah metode dalam *deep learning* yang sering digunakan untuk mengolah gambar. CNN terdiri dari dua komponen utama, yakni *Feature Extraction Layer* (termasuk *Convolutional Layer* dan *Pooling Layer*) dan *Fully Connected Layer* [4]. *Convolutional Layer* dalam CNN menggunakan

kernel dan *filter* untuk mengekstraksi fitur-fitur penting pada gambar masukan, menghasilkan output yang dikenal sebagai *feature map* [20] [4]. *Pooling Layer*, yang berfungsi untuk *downsampling*, mengurangi dimensi *feature map* dari *Convolutional Layer* [4]. *Fully Connected Layer*, sebagai lapisan terakhir, bertugas melakukan kategorisasi berdasarkan fitur-fitur penting yang diidentifikasi pada *feature map* secara keseluruhan [20]. Ilustrasi arsitektur CNN dapat ditemukan dalam Gambar 2.5.



Gambar 2.5 Arsitektur CNN [31]

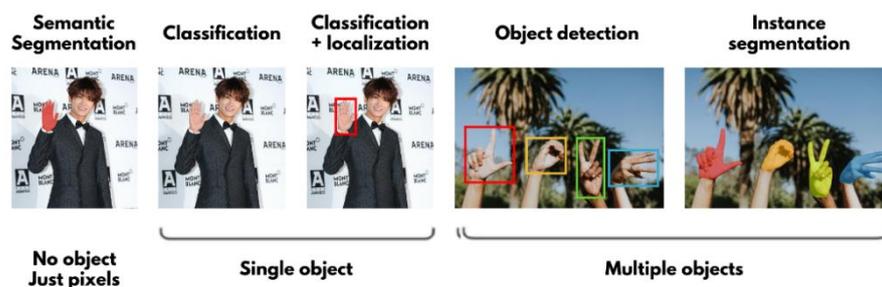
Gambar 2.5 merupakan arsitektur CNN yang melibatkan input berupa citra atau gambar. Tahapan ini mencakup proses konvolusi dan fungsi aktivasi *ReLU*, diikuti oleh proses *pooling*. Seluruh rangkaian ini diulang beberapa kali, dan akhirnya menghasilkan output berupa proses klasifikasi.

2.2.5 Computer Vision

Computer vision merupakan salah satu domain AI bersama dengan *Data Science* dan *Natural Language Processing*. Fokusnya adalah pada kemampuan mesin untuk meniru kemampuan manusia dalam pengolahan visual dari berbagai media digital, seperti foto dan video, untuk memperoleh informasi krusial. Mesin ini dilatih untuk melakukan tindakan atau memberikan rekomendasi berdasarkan informasi yang diperoleh. Untuk mencapai hasil model yang baik, *computer vision* memerlukan sejumlah besar data. Cara kerjanya

adalah dengan menganalisis data secara berulang hingga menemukan perbedaan yang dapat mengidentifikasi objek dalam gambar [32].

Computer vision memanfaatkan kemajuan dalam teknologi *deep learning* dan *Convolutional Neural Network* (CNN). Teknologi *deep learning* memungkinkan komputer untuk secara mandiri mampu menemukan perbedaan dalam foto atau video. CNN membantu dalam proses visualisasi dengan membagi gambar menjadi piksel yang diberi label dan melakukan konvolusi untuk menghasilkan prediksi dari apa yang terlihat oleh komputer. *Computer vision* memiliki kemampuan untuk melaksanakan beberapa tugas yaitu klasifikasi gambar (*image classification*), deteksi objek (*object detection*), pelacakan objek (*object tracking*), dan pengambilan gambar berbasis konten (*content-based image retrieval*) [32]. Beberapa tugas ini dapat dilihat pada Gambar 2.6.

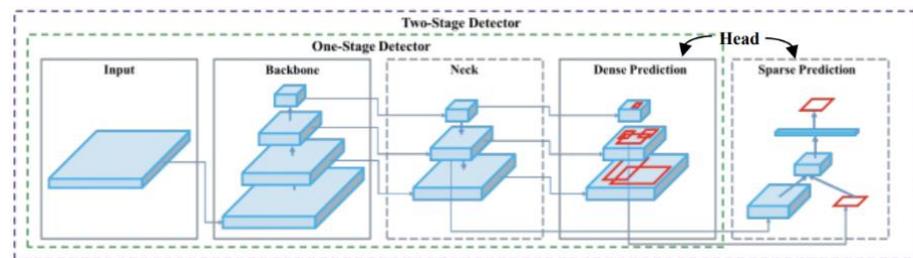


Gambar 2.6 Tugas dalam *Computer Vision*

2.2.6 *Object Detection*

Detektor objek CNN dapat dikelompokkan menjadi beberapa jenis, termasuk *one-stage detector* (YOLOX, FCOS, DETR, Scaled-YOLOv4, dan EfficientDet), *two-stage detector* (VFNet dan CenterNet2), *anchor-based detector* (Scaled-YOLOv4 dan YOLOv5), dan *anchor-free detector* (CenterNet, YOLOX, dan RepPoints). Secara umum, mereka terdiri dari dua komponen utama, yaitu *backbone* (VGG, ResNet, dll.) untuk ekstraksi fitur dan *head* untuk prediksi kelas dan *bounding box*. Tren pengembangan mencakup penambahan lapisan "*neck*" antara *backbone* dan *head* [33]. Meskipun

backbone umumnya menggunakan arsitektur umum seperti VGG, ResNet, DenseNet, MobileNet, EfficientNet, CSPDarknet53, Swin Transformer, dan sebagainya, penelitian terus mengoptimalkan mereka. *Neck* dirancang untuk memproses dan menggunakan fitur yang diekstraksi oleh *backbone*. *Head*, yang berfungsi mendeteksi lokasi dan kategori objek, dapat dibagi menjadi *one-stage detector* dan *two-stage detector*, dengan *two-stage detector* seperti keluarga RCNN dan *one-stage detector* seperti YOLO, SSD, dan RetinaNet [33]. Ilustrasi dua konsep arsitektur *object detection* dapat dilihat pada gambar berikut.



Gambar 2.7 Arsitektur Konsep *Object Detection* [34]

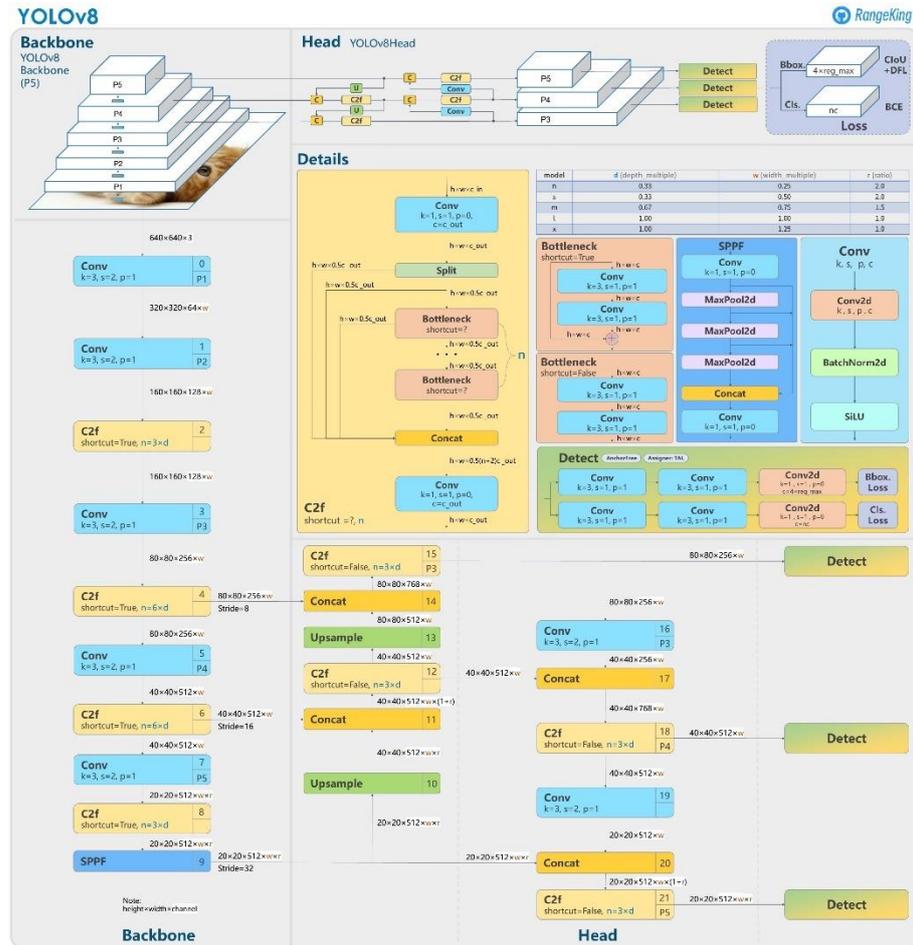
2.2.7 Preprocessing

Proses *preprocessing* gambar merupakan tahap yang sangat krusial dalam upaya untuk memperoleh fitur optimal dan meningkatkan akurasi klasifikasi. Langkah penting dalam proses ini adalah menyesuaikan ukuran gambar (*resizing*), diikuti dengan normalisasi data untuk mencapai konsistensi dalam rentang nilai variabel, umumnya skala antara 0 hingga 1 [36]. Setelahnya, dilakukan augmentasi data untuk memperkenalkan variasi bentuk tambahan pada gambar yang akan dilatih. Augmentasi data adalah teknik untuk memperluas ukuran dataset pelatihan dengan melakukan transformasi pada data, seperti perubahan warna, *flip*, *rotate*, *crop*, *shear*, *exposure*, *blur*, *noise*, tanpa mengubah labelnya. Hal ini bertujuan untuk mengatasi masalah *overfitting* dan *underfitting* dengan keyakinan bahwa informasi tambahan dapat diekstraksi dari dataset asli melalui augmentasi, sehingga memperkaya dataset

pelatihan [37]. *Preprocessing* data ini dilakukan untuk mengubah data ke format yang diinginkan, menghasilkan dataset yang lebih besar dan beragam, sehingga pada akhirnya dapat meningkatkan kinerja model yang digunakan dalam analisis gambar [38].

2.2.8 YOLOv8

YOLOv8 merupakan versi terbaru dari model *Computer Vision You Only Look Once*, diperkenalkan oleh Ultralytics pada Januari 2023 [11]. Meskipun mengadopsi arsitektur *backbone* CSPDarknet53 dari YOLOv5, YOLOv8 menambahkan *C2f module* untuk meningkatkan akurasi dengan pemrosesan fitur-fitur dengan informasi kontekstual [21]. Berbeda dengan versi sebelumnya, YOLOv8 tidak menggunakan *anchor boxes* dalam melakukan prediksi label atau kelas pada suatu gambar, sehingga dapat mengurangi beban komputasi [11]. Perhitungan nilai *loss* pada YOLOv8 terdiri dari dua aspek utama, yaitu *classification loss* menggunakan *Binary Cross Entropy* (BCE), dan *bounding box loss* dengan dua fungsi perhitungan *loss*, yakni *Complete Intersection over Union* (CIoU) dan *Distribution Focal Loss* (DFL). Meskipun fungsi aktivasi utama yang digunakan adalah SiLU, YOLOv8 juga menyajikan pilihan fungsi aktivasi lainnya, seperti LeakyReLU, ReLU, Sigmoid, dan sejenisnya [35]. Evaluasi menggunakan dataset MS COCO 2017 menunjukkan bahwa YOLOv8 berhasil mencapai tingkat akurasi sebesar 53,9% pada IoU 0,5 [11].



Gambar 2.8 Arsitektur YOLOv8 [35]

Ilustrasi bentuk arsitektur YOLOv8 seperti pada Gambar 2.8, yang melibatkan komponen, lapisan, atau modul, seperti C2f, Bottleneck, SPPF (*Spatial Pyramid Pooling Fast*), Conv, dan Detect. Pada C2f, komponen Conv dipanggil untuk menjalankan proses konvolusi, operasi Split digunakan untuk memecah *feature map* menjadi beberapa bagian, Bottleneck berfungsi untuk mengurangi beban komputasi melalui proses konvolusi lanjut, dan operasi Concat digunakan untuk menggabungkan kembali beberapa *feature map* yang sebelumnya telah dipisahkan pada operasi Split. SPPF melibatkan komponen Conv, MaxPool2d untuk melaksanakan proses *pooling* pada gambar dua dimensi, dan operasi Concat. Conv menjalankan proses konvolusi dua dimensi yang diikuti oleh normalisasi dan menggunakan fungsi aktivasi SiLU. Selanjutnya, pada lapisan Detect,

dilakukan proses konvolusi yang diikuti oleh perhitungan dua nilai *loss*, yaitu *classification loss* dan *bounding box loss*.

Proses ekstraksi fitur pada YOLOv8 mengadopsi konsep Piramida dengan beberapa tingkatan (P1 hingga P5), masing-masing merepresentasikan tingkat resolusi dan cakupan detail berbeda pada gambar. Tingkatan ini dimulai dari P1 yang memiliki resolusi tertinggi dan menangkap detail paling luas, hingga P5 yang memiliki resolusi terendah dan fokus pada detail yang lebih sempit. Proses konvolusi pada gambar yang dimasukkan ke dalam arsitektur YOLOv8 diulang beberapa kali, dimulai dengan resolusi tertentu yang terus berkurang melalui setiap iterasi konvolusi.

Setiap lapisan konvolusi memiliki parameter-parameter seperti jumlah *kernel*, jumlah *stride*, dan jumlah *padding*. Selanjutnya, terdapat C2f, varian dari C2 dengan proses lebih cepat, yang berfungsi menggabungkan fitur-fitur dengan informasi kontekstual untuk meningkatkan akurasi deteksi. C2f memiliki parameter *shortcut*, yang dapat diatur sebagai *Boolean*, memungkinkan proses ekstraksi berurutan atau tidak pada setiap iterasi.

Di ujung *backbone*, dilakukan proses *pooling* menggunakan modul SPFF. Selanjutnya, menuju ke *head*, dilakukan proses *upsampling* terlebih dahulu jika *pooling* telah dilakukan pada *backbone* untuk menyamakan ukuran resolusi dengan hasil konvolusi sebelumnya. Operasi *Concat* kemudian digunakan untuk menggabungkan hasil konvolusi yang sebelumnya dipisahkan pada C2f. Puncak dari *head* adalah *layer Detect*, di mana dilakukan perhitungan nilai *loss* dari operasi deteksi objek pada sampel gambar pelatihan.

2.2.9 Confusion Matrix

Confusion matrix merupakan sebuah instrumen evaluasi yang digunakan untuk menilai kinerja suatu model dengan

mempertimbangkan *precision*, *recall*, *F1 Score* dan akurasi (mAP^{50}). Terdapat empat parameter utama yang terlibat, yaitu *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). Secara prinsip, *confusion matrix* digunakan untuk membandingkan hasil klasifikasi model dengan hasil klasifikasi yang diharapkan. Dengan menganalisis kombinasi nilai dalam tabel *confusion matrix*, berbagai metrik evaluasi model dapat dihitung, seperti *precision*, *recall*, *F1 Score* dan akurasi (mAP^{50}). Berikut perhitungan *confusion matrix* terdapat pada Tabel 2.2.

Tabel 2.2 *Confusion Matrix*

		Prediksi	
		Positif	Negatif
Aktual	Positif	TP	FN
	Negatif	FP	TN

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

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

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

$$mAP = \frac{1}{N} \sum_{i=1}^n AP_i \quad (2.4)$$

$$AP = \frac{1}{11} \sum_{r=0.0}^{1.0} p(r) \quad (2.5)$$

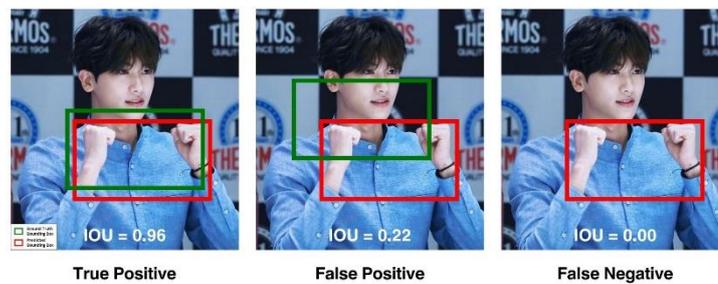
Nilai *precision* merupakan kemampuan dari model dalam mengidentifikasi objek-objek yang relevan pada sebuah gambar yang dihitung dengan Persamaan (2.1). Nilai *recall* merupakan kemampuan dari model dalam mencari keseluruhan kemungkinan *bounding boxes* yang ada pada sebuah gambar yang dihitung dengan Persamaan (2.2). Nilai *F1 Score* merupakan rata-rata *harmonic* dari nilai *precision* dan *recall* untuk mengindikasikan keseimbangan antara kedua metrik tersebut yang dihitung dengan Persamaan (2.3). *True Positive* (TP)

adalah jumlah keseluruhan gambar yang diprediksi benar (positif) dan sesuai dengan kelas bahasa isyarat (positif). *True Negative (TN)* adalah jumlah keseluruhan gambar yang diprediksi salah (negatif) dan tidak sesuai dengan kelas bahasa isyarat (negatif). *False Positive (FP)* adalah jumlah keseluruhan gambar yang diprediksi benar (positif) namun tidak sesuai dengan kelas bahasa isyarat (negatif). Dan *False Negative (FN)* adalah jumlah keseluruhan gambar yang diprediksi salah (negatif) namun data aktual bersifat sesuai dengan kelas bahasa isyarat (positif).

Sedangkan *mAP (Mean Average Precision)* merupakan nilai rata-rata dari *Average Precision* pada setiap *class* yang dihitung dengan Persamaan (2.4). Nilai *Average Precision* dapat dihitung dengan Persamaan (2.5). Untuk perhitungan nilai *mAP* sangat bergantung terhadap nilai dari *Intersection Over Union (IoU)*. *IoU* mengukur seberapa besar nilai *overlapping* pada *bounding boxes* yang dirumuskan pada Gambar 2.9. Semakin tinggi nilai dari *IoU* maka akan semakin akurat juga letak dari *bounding boxes* pada sebuah gambar seperti yang ditunjukkan pada Gambar 2.10. Penelitian-penelitian yang berkaitan dengan citra deteksi objek, biasanya perhitungan nilai akurasi mengacu pada *mAP*. Pada penelitian ini, perhitungan nilai akurasi akan menggunakan *mAP⁵⁰* yang merupakan rata-rata keseluruhan dari nilai *AP* dengan *IoU* 0,5. Namun perlu digarisbawahi bahwa nilai *mAP* yang terlalu tinggi belum tentu baik di saat melakukan proses deteksi apabila diberikan data yang baru (*overfitting*), begitu juga jika nilai *mAP* yang terlalu rendah maka model yang dilatih tidak akan dapat mendeteksi dengan baik (*underfitting*).

$$IOU = \frac{\text{area of overlap}}{\text{area of union}} = \frac{\text{[Diagram showing two overlapping rectangles, one green and one red, with their intersection shaded blue]}{\text{[Diagram showing the union of the two overlapping rectangles, shaded blue]}}$$

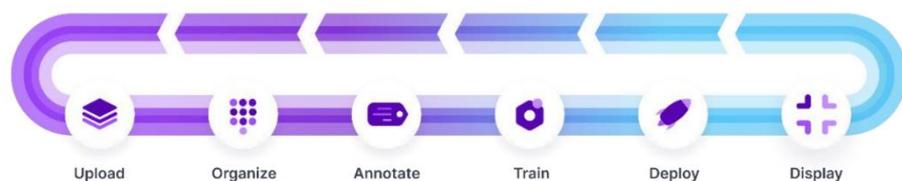
Gambar 2.9 Rumus Perhitungan Nilai IoU [41]



Gambar 2.10 Contoh Penerapan Perbedaan Nilai IoU

2.2.10 Roboflow

Roboflow merupakan sebuah *platform* yang berperan khusus untuk memberikan bantuan kepada para *AI engineer* dalam memproses dataset yang mengaplikasikan proyek *computer vision*.

Gambar 2.11 *Roboflow* [42]Gambar 2.12 Alur Kerja *Computer Vision* pada *Roboflow* [42]

Roboflow menyediakan layanan yang mendukung pengembang *computer vision* dalam menjalankan proyek sebagaimana dijelaskan dalam alur kerja yang tergambar pada Gambar 2.12. Langkah awal melibatkan proses unggah dataset ke *platform Roboflow*, di mana dataset tersebut disederhanakan untuk mempermudah anotasi dan pemberian *bounding box*. Data yang telah disiapkan kemudian dapat digunakan sebagai input untuk melatih model, dan selanjutnya dilakukan *deployment* serta hasil deteksi dari model yang telah dilatih dapat dievaluasi [42].

Roboflow menyediakan akses ke 20 library model yang beragam, yaitu YOLOX, Vision Transformer, YOLOv5, YOLOR, OpenAI Clip, Scaled-YOLOv4, Resnet32, EfficientDet-D0-D7, YOLOv4-tiny, YOLOv4 Darknet, EfficientNet, Detectron2, EfficientDet, Faster R-CNN, YOLO v3 PyTorch, YOLO v3 Keras, MobileNetSSDv2, YOLOv4, PyTorch, MobileNetV2 Classification, dan ResNet-32 [43].

2.2.11 Python

Python adalah bahasa pemrograman yang berkembang sebagai turunan dari bahasa pemrograman ABC. Guido van Rossum, pada tahun 1990-an, memimpin pengembangan Python di CWI, Amsterdam. Python dikenal sebagai bahasa pemrograman dinamis yang mendukung paradigma berbasis objek. Keunggulan Python meliputi lisensi *Open Source* yang tidak bertentangan dengan *General Public License* (GPL), memungkinkan penggunaan bebas termasuk untuk tujuan komersial. Python juga terkenal karena struktur data tingkat tinggi yang efisien. Fleksibilitas dan kemampuannya dalam menangani berbagai jenis tugas pemrograman membuat Python sangat serbaguna dan populer di komunitas pengembang perangkat lunak. Beberapa kelebihan Python termasuk [44]:

- a. Pembelajaran mudah berkat struktur yang sederhana.

- b. Kode mudah dibaca dengan penggunaan spasi yang 'menjorok' untuk menjaga kebersihan kode.
- c. Kode Python umumnya lebih singkat daripada bahasa lain untuk menyelesaikan masalah serupa.
- d. Kompatibel dengan berbagai sistem operasi, termasuk Windows, UNIX, Linux, dan Mac OS.
- e. Mendukung pemrograman prosedural.
- f. Memiliki beragam library dan paket yang dapat digunakan untuk keperluan khusus.



Gambar 2.13 Python

2.2.12 *Google Colab*

Google Colab merupakan *Integrated Development Environment* (IDE) yang telah didesain khusus untuk pemrograman dalam bahasa Python, dimana proses komputasi dan pemrosesan data dilaksanakan di server Google yang dilengkapi dengan perangkat keras unggulan. Dari perspektif perangkat lunak, *Google Colab* telah menyediakan sebagian besar pustaka (*library*) yang diperlukan, seperti Keras, TensorFlow, NumPy, Pandas, dan pustaka-pustaka pendukung lainnya, termasuk yang dibutuhkan untuk visualisasi Matplotlib. Seluruh versi dari pustaka-pustaka tersebut dapat diakses, termasuk TensorFlow dalam versi 1.x dan 2.x, serta versi Python dari 2.x hingga 3.x. Secara terintegrasi dengan Google Drive, *Google Colab* menyediakan layanan penyimpanan data dan menyajikan berbagai jenis prosesor seperti CPU, GPU, dan TPU, dengan kapasitas RAM yang dapat disesuaikan. Dengan ketersediaan server yang andal, hampir semua proses pemrosesan dapat dijalankan dengan lancar di *Google Colab*, selama terdapat koneksi internet yang stabil [45].



Gambar 2.14 *Google Colab*

2.2.13 Streamlit

Streamlit adalah suatu *framework* sumber terbuka yang menggunakan bahasa pemrograman Python, dirancang untuk mempermudah pengembangan aplikasi web, terutama yang berkaitan dengan *data science* dan *machine learning*. *Framework* ini menyediakan berbagai fitur yang mendukung pengembangan model *machine learning*. Salah satu keunggulan Streamlit adalah kemampuannya dalam menyajikan visualisasi data melalui antarmuka interaktif, memungkinkan komunikasi informasi yang relevan kepada pengguna. Proses penyebaran (*deploy*) aplikasi melalui Streamlit juga terbilang mudah, dapat dilakukan melalui *platform* berbagi *cloud* Streamlit, dan aplikasi yang dihasilkan dapat diakses melalui berbagai *platform* [19].