

## BAB II

### LANDASAN TEORI

#### 2.1 Penelitian Terdahulu

Sebelum memasuki pembahasan lebih lanjut mengenai pengembangan penelitian, penting untuk melakukan analisis dari temuan-temuan yang telah tercatat dalam literatur sebelumnya. Berikut merupakan tabel perbandingan dari penelitian terdahulu yang membahas topik *Personal Protective Equipment (PPE) Object Detection*:

Tabel 2. 1 Tabel Penelitian Terdahulu

Nama Jurnal & Nama Artikel	Penulis	Vol/ No	Masalah	Simpulan
Machine Learning Knowledge Extraction 2023, 5, 1680-1716.  “A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS”	Terven J, Córdova-Esparza D-M, Romero-González J-A.	Vol 5	Menganalisis peningkatan model YOLO hingga YOLOv8, membahas spesifikasi setiap versi, menyoroti adaptasi dalam arsitektur yang memfasilitasi peningkatan ini, dan dampaknya terhadap kinerja di berbagai aplikasi.	Penelitian menunjukkan terjadi peningkatan akurasi kecepatan pemrosesan, dan meningkatkan efektifitas pada aplikasi secara real-time pada evolusi tiap versi yaitu YOLOv1 hingga YOLOv8.
Ultima Computing : Jurnal Sistem Komputer, 15(1), 8-13. 2023  “Performance Comparison of YOLOv5 and YOLOv8 Architectures in Human Detection using Aerial Images”	Sary, I., Andromeda, S., & Armin, E.	Vol 15 / No 1	Membuat perbandingan model YOLOv5 dan YOLOv8 dalam mendeteksi objek manusia dalam gambar yang diambil dari udara.	Hasil penelitian menunjukkan nilai performa model YOLOv8 lebih baik dibandingkan model YOLOv5 untuk presisi dan skor F1, selisih nilai masing-masing performa sebesar 2,82%, dan 0,98%.
Lecture Notes in Computer Science Mobile Web and Intelligent Information Systems 2023	Selcuk, Burcu & Serif, Tacha	LNCS, volume 13977	Membuat perbandingan model YOLOv5 dan YOLOv8 dalam elemen <i>Graphical User Interface (GUI)</i>	Penelitian menunjukkan bahwa model YOLOv8s dan YOLOv8n menghasilkan kinerja mAP lebih baik yaitu sebesar 3,32% dan 1,62%,

“A Comparison of YOLOv5 and YOLOv8 in the Context of Mobile UI Detection”			pada aplikasi berbasis mobile.	dibandingkan benchmark YOLOv5s.
<b>Nama Jurnal &amp; Nama Artikel</b>	<b>Penulis</b>	<b>Vol/ No</b>	<b>Masalah</b>	<b>Simpulan</b>
<i>Journal of Engineering and Technological Sciences, 2024</i>  “Comparison Study of Corn Leaf Disease Detection based on Deep Learning YOLO-v5 and YOLO-v8”	Chitraningrum, N., Banowati, L., Herdiana, D., Mulyati, B., Sakti, I., Fudholi, A., Saputra, H., Farishi, S., Muchtar, K., & Andria, A.	Vol 56/ No 1	Membuat perbandingan model YOLOv5 dan YOLOv8 dalam mendeteksi penyakit pada daun tanaman jagung.	Hasil penelitian menunjukkan bahwa YOLOv8 menunjukkan performa akurasi yang lebih baik dalam mendeteksi objek yaitu mAP50 sebesar 0,953 dibandingkan YOLOv5.
IEEE Access, 2023  “Deep Learning-Based YOLO Models for the Detection of People With Disabilities”	Alruwaili, Madallah & Atta, Muhammad & Siddiqi, Muhammad & Khan, Abdullah & Khan, Asfandyar & Alhwaiti, Yousef & Alanazi, Saad.	Vol 12	Membandingkan performa model YOLOv5, YOLOv7, dan YOLOv8 dalam mendeteksi penyandang disabilitas.	Hasil penelitian menunjukkan YOLOv8 memiliki performa terbaik dalam mendeteksi penyandang disabilitas dibandingkan YOLOv5 dan YOLOv7. Diperoleh hasil terbaik YOLOv8 dengan mAP50 sebesar 0,951; presisi 0,907, dan recall sebesar 0,887.
2021 International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON) Pune, India. Oct 29-30, 2021  “Personal Protective Equipment Kit Detection using Yolo v5 and TensorFlow”	Udatewar, Pallavi & Desai, Abhishek & Godghase, Gauri & Nair, Arunkumar & Kosamkar, Pranali..		Membuat perbandingan performa model <i>Personal Protective Equipment (PPE) object detection</i> dengan menggunakan Tensorflow model, dan YOLOv5	Hasil penelitian menunjukkan model yang memberikan performa terbaik dalam melakukan <i>PPE Object Detection</i> adalah YOLOv5 dengan memperoleh nilai <i>minimum classification</i> sebesar 3.719e-3 at 500 <i>epoch</i> . Model YOLOv5 menunjukkan hasil yang lebih baik dari model Tensorflow.
Sustainability 2023  “Real-Time Personal Protective Equipment Compliance Detection Based	Lo, Jye-Hwang & Lin, Lee-Kuo & Hung, Chu-Chun	Vol 15 /391	Mengembangkan model deteksi <i>Personal Protective Equipment (PPE)</i> berbasis YOLOv7 dan membandingkan kinerjanya dengan YOLOv3 dan YOLOv4.	Hasil penelitian menunjukkan efisiensi YOLOv7 yang menunjukkan performa lebih tinggi dari versi YOLO sebelumnya (v3 dan v4) yaitu akurasi mAP sebesar 97% dan efisiensi diatas 25 FPS.

on Deep Learning Algorithm”				
Nama Jurnal & Nama Artikel	Penulis	Vol/ No	Masalah	Simpulan
Seminar Nasional Inovasi Teknologi UN PGRI Kediri, 2022  “Deteksi Kelalaian Alat Pelindung Diri (APD) pada Pekerja Kontruksi Bangunan”	Nurfirmaryah, Agustin Dijaya, Rohman		Menerapkan metode YOLOv4 untuk mendeteksi pelanggaran APD pada pekerja secara <i>real-time</i> .	Berdasarkan hasil penelitian, objek pada citra APD berhasil dikenali dengan akurasi 63,03% sampai 99,4% menggunakan GPU yang kecepatan tinggi dan memiliki akurasi rata-rata 80% dengan validasi <i>mAP</i> sebesar 99,7% dan <i>avg loss</i> sebesar 1.305%.
Jurnal Infra, 2020  “Deteksi Alat Pelindung Diri Menggunakan Metode YOLO dan Faster R-CNN”	Adiwibowo, Jonathan Gunadi, Kartika Styati, Endang	Vol 8/ 2 2020	Menerapkan metode YOLO v3 dan metode <i>Faster R-CNN (FRCNN)</i> untuk mendeteksi alat keselamatan kerja untuk bagian kepala.	YOLOv3 memiliki hasil yang bagus yaitu 73.83% untuk mendeteksi kepala pekerja di dalam video berukuran 720x480. Akan tetapi YOLOv3 masih memiliki kelemahan yaitu susah mendeteksi saat 2 atau lebih kepala saling berhimpitan sehingga dianggap menjadi 1 region.
JURNAL TEKNIK ITS, 2023  “Sistem Pendeteksi Alat Pelindung Diri (APD) pada Pekerja Konstruksi Berbasis Convolutional Neural Network”	Nirvana, Muhammad N Rauchmadi, Reza R Purnama, I Ketut E	Vol 12, No. 3	Mendeteksi Alat Pelindung Diri (APD) dengan kamera menggunakan metode deteksi objek YOLOv7 berbasis CNN.	Berdasarkan seluruh pengujian, varian YOLOv7 menjadi pilihan yang optimal untuk deteksi APD yang kemudian digunakan untuk semua percobaan pada pengujian sistem. Hasil testing menggunakan testing set, pengujian berdasarkan perbedaan jarak, pengujian pada kondisi tidak ideal, dan pengujian pada kondisi keramaian secara berurutan memiliki <i>mAP</i> bernilai 0.877, 0.93, 0.784, dan 0,856.

Berdasarkan Tabel 2.1 adaptasi teknologi deteksi objek YOLO untuk Alat Pelindung Diri (APD) menunjukkan perkembangan signifikan dalam kemampuan mendeteksi pelanggaran keamanan di lokasi konstruksi.

Penelitian yang ditulis oleh Udatewar, et al menunjukkan bahwa algoritma YOLOv5 memberikan hasil performa yang lebih baik dibandingkan model Tensorflow dalam mendeteksi objek yaitu Alat Pelindung Diri [12]. Pada penelitian yang ditulis oleh Lo dengan judul "*Real-Time Personal Protective Equipment Compliance Detection Based on Deep Learning Algorithm*" [13], menunjukkan bahwa penggunaan model YOLOv7 dapat meningkatkan akurasi mAP sebesar 97% dibandingkan performa versi sebelumnya yaitu YOLOv3, dan YoLOv4. Selain itu, penelitian yang dilakukan oleh Nurfirmansyah dan Dijaya menggunakan YOLOv4 menunjukkan akurasi yang cukup tinggi yaitu mencapai 99,4% [14]. Studi lain oleh Adiwibowo et al. membandingkan YOLOv3 dan Faster R-CNN dalam mendeteksi alat keselamatan pekerja, menemukan bahwa YOLOv3 efektif [15]. Dapat disimpulkan bahwa metode algoritma *You Only Look Once* (YOLO) terbukti memberikan pengaruh performa akurasi yang baik dibandingkan model deteksi objek lainnya.

Penelitian yang ditulis oleh Terven dan Romero yang membahas perkembangan YOLOv1 hingga YOLOv8 menunjukkan bahwa evolusi versi YOLO didukung oleh terjadinya peningkatan kecepatan proses, akurasi, dan efektivitas secara real-time [11]. Penelitian yang berjudul "*Performance Comparison of YOLOv5 and YOLOv8 Architectures in Human Detection using Aerial Images*" [16], "*A Comparison of YOLOv5 and YOLOv8 in the Context of Mobile UI Detection*" [17], dan "*Comparison Study of Corn Leaf Disease Detection based on Deep Learning YOLO-v5 and YOLO-v8*" [18] melakukan perbandingan terhadap performa model YOLOv5 dan YOLOv8 terhadap objek penelitiannya masing-masing. Ketika penelitian tersebut menunjukkan YOLOv8 memiliki hasil yang lebih baik dibandingkan model YOLOv5. Selain itu penelitian yang berjudul "*Deep Learning-Based YOLO Models for the Detection of People With Disabilities*" juga melakukan perbandingan model YOLOv5, YOLOv7, dan YOLOv8 dan memperoleh hasil performa terbaik oleh YOLOv8 yaitu akurasi mAP50 sebesar 0,951 [19]. Melalui penelitian-penelitian tersebut, dapat

disimpulkan YOLOv8 merupakan hasil pengembangan yang menunjukkan hasil performa akurasi deteksi objek yang lebih baik dari versi-versi YOLO sebelumnya.

Berdasarkan analisa tersebut, penelitian dengan judul "Implementasi Algoritma YOLOv8 untuk Mendeteksi Alat Pelindung Diri (APD) pada Pekerja Konstruksi di Unit K3L UMN" melakukan implementasi dari versi terbaru YOLO yaitu YOLOv8 yang unggul pada penelitian sebelumnya [11][16][17][18][19], namun terhadap perbedaan yang dilakukan dalam konteks deteksi Alat Pelindung Diri / *Personal Protective Equipment Detection* yaitu penggunaan versi YOLO terbaru yaitu YOLOv8 karena penelitian sebelumnya baru mengimplementasikan YOLO versi sebelumnya pada deteksi objek Alat Pelindung Diri (APD) yaitu YOLOv5 [12], YOLOv3-YOLOv4-YOLOv7 [13], YOLOv4 [14], YOLOv3[15], dan YOLOv7 [20]. Oleh karena itu, novelty penelitian ini terletak pada penggunaan YOLOv8 dengan harapan akan memberikan peningkatan hasil performa akurasi deteksi objek APD. Selain itu, model YOLOv8 yang akan dilatih akan diimplementasikan secara langsung untuk digunakan pada Unit K3L Universitas Multimedia Nusantara yang bertanggung jawab terhadap Kesehatan dan Keselamatan Kerja (K3) untuk seluruh sivitas di wilayah universitas. Hal ini dilakukan dengan tujuan untuk mengurangi terjadinya kelalaian APD yang dapat menyebabkan terjadinya resiko kecelakaan pada pekerja konstruksi.

## 2.2 Teori yang digunakan

### 2.2.1 *Object Detection*

*Object Detection* adalah suatu tugas dalam bidang visi komputer yang bertujuan untuk mendeteksi dan mengidentifikasi objek-objek tertentu dalam gambar atau video [21]. Tugas ini melibatkan dua aspek utama: lokalisasi objek (menemukan di mana objek tersebut berada dalam gambar) dan klasifikasi objek (mengidentifikasi jenis atau kategori objek tersebut).

Tujuan akhir dari *Object Detection* adalah memberikan informasi tentang keberadaan dan atribut objek secara simultan [22].

### **2.2.2 *Image Processing***

*Image processing* adalah suatu bidang dalam ilmu komputer yang berkaitan dengan pemrosesan dan analisis citra atau gambar. Tujuan utama dari *image processing* adalah untuk meningkatkan kualitas citra, mengekstrak informasi yang berguna, dan mengambil keputusan berdasarkan data visual [23]. *Image processing* dapat digunakan dalam berbagai aplikasi, termasuk pengolahan gambar medis, pengenalan pola, pengolahan gambar satelit, pengolahan gambar digital, dan banyak lagi.

### **2.2.3 *Machine Learning***

*Machine learning* merupakan sebuah ilmu yang menggunakan teknologi pembelajaran mesin secara matematis yang banyak digunakan untuk menyelesaikan suatu permasalahan dan memudahkan pengerjaan suatu kegiatan [24]. Penggunaan *machine learning* banyak digunakan untuk mendapatkan suatu informasi atau wawasan dari sekumpulan data yang dimiliki agar informasi tersebut dapat digunakan secara lebih efisien atau efektif, misalnya untuk melihat suatu pola, memprediksi suatu kejadian, dan membantu manusia dalam menentukan keputusan.

### **2.2.4 *Deep Learning***

*Deep learning* merupakan cabang dari pembelajaran mesin yang fokus pada pengembangan dan penerapan algoritma yang dapat belajar dan membuat prediksi berdasarkan data dalam jumlah besar. Ini menggunakan jaringan saraf tiruan dengan banyak lapisan (*deep neural networks*) untuk memodelkan proses belajar yang kompleks[25]. *Deep learning* telah

terbukti sangat efektif dalam berbagai aplikasi termasuk pengenalan suara, pengolahan bahasa alami, dan pengenalan objek dalam gambar dan video.

Salah satu keunggulan utama dari *deep learning* adalah kemampuannya untuk mengekstraksi fitur tingkat tinggi dari data mentah secara otomatis, yang memungkinkan model untuk belajar representasi data yang kompleks tanpa kebutuhan intervensi atau pemberian fitur yang ditentukan secara manual. Ini dicapai melalui proses pelatihan yang intensif menggunakan *backpropagation*, sebuah metode untuk memperbarui bobot jaringan berdasarkan gradien dari fungsi kerugian [26].

### 2.2.5 *Confusion Matrix*

*Confusion Matrix* memiliki dua dimensi, dengan indeks pertama menunjukkan kelas objek aktual dan indeks kedua menunjukkan kelas perkiraan. *Matrix* ini dapat membantu menyempurnakan klasifikasi atau perkiraan yang diturunkan dari klasifikasi tersebut. Dengan menggunakan data yang ada pada matriks tersebut, kinerja model klasifikasi yang dibuat dinilai. Berikut merupakan gambaran dari sebuah *Confusion Matrix*

Tabel 2. 2 *Confusion Matrix*

	<i>Actually Positive (1)</i>	<i>Actually Negative (0)</i>
<i>Predicted Positive (1)</i>	<i>True Positive (TP)</i>	<i>False Positive (FP)</i>
<i>Predicted Negative (0)</i>	<i>False Negative (FN)</i>	<i>True Negative (TN)</i>

*True Negative (TN)* adalah nilai yang benar-benar menunjukkan bahwa sebuah objek bernilai negatif, *False Negative (FN)* adalah nilai yang menunjukkan bahwa sebuah objek bernilai negatif, *False Positive (FP)* adalah nilai yang menunjukkan bahwa sebuah objek bernilai positif, dan *True Positive (TP)* adalah nilai yang menunjukkan bahwa sebuah objek bernilai positif [27]. Nilai dari TN, FN, FP, TP, kita dapat melakukan kalkulasi untuk *metrics*

seperti *precision*, *recall*, *specificity*, *accuracy*, dan *F1 score* dapat dihitung.

### 2.2.6 *Precision*

*Precision* merupakan suatu metrik yang digunakan untuk mengukur kualitas dan performa model prediksi dalam konteks *machine learning* dan statistika. Nilai *precision* menunjukkan rasio hasil *True Positive* (TP) pada hasil prediksi terhadap total keseluruhan hasil positif yang diperoleh. *Precision* berperan penting dalam mengukur performa implementasi model yang perlu menghindari terjadinya *False Positive* [28]. Berikut merupakan rumus perhitungan dari metrik *Precision*:

$$Precision = \frac{TP}{TP + FP}$$

Rumus 2. 1 Perhitungan nilai *Precision*

Keterangan:

- TP (*True Positive*) = Jumlah kasus positif yang terdeteksi oleh model sebagai positif.
- FP (*False Positive*) = Jumlah kasus negatif yang terdeteksi model sebagai positif.

### 2.2.7 *Recall*

*Recall* merupakan suatu metrik evaluasi yang digunakan untuk mengukur kemampuan model dalam melakukan identifikasi terhadap nilai positif dari keseluruhan nilai yang ada. Nilai *recall* menggambarkan rasio dari jumlah *True Positive* terhadap total nilai positif sebenarnya dalam keseluruhan positive yang ada [28]. Nilai *Recall* berperan penting dalam kasus yang tidak bisa mentolerir kesalahan deteksi pada kasus *positive* (*false negative*), contohnya deteksi suatu penyakit. Berikut merupakan rumus perhitungan dari metrik *Recall*:



$$Recall = \frac{TP}{TP + FN}$$

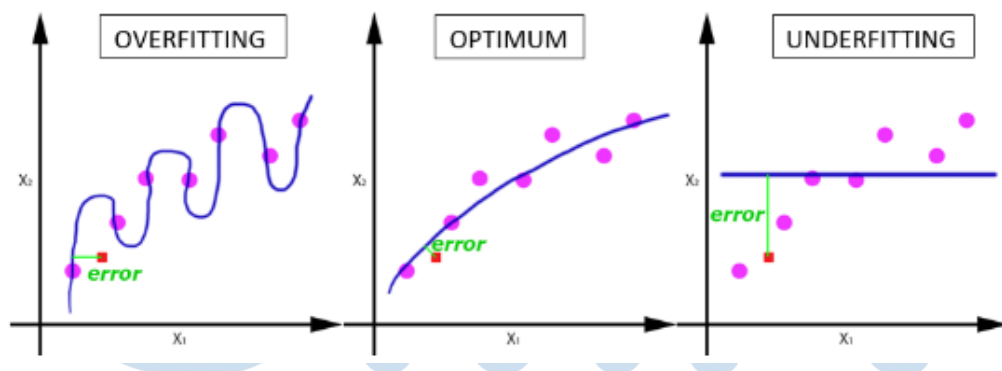
Rumus 2. 2 Perhitungan nilai *Recall*

Keterangan:

- TP (*True Positive*) = Jumlah kasus positif yang terdeteksi oleh model sebagai positif.
- FN (*False Negative*) = Jumlah kasus positif yang terdeteksi model sebagai negatif.

### 2.2.8 *Epoch*

*Epoch* merupakan istilah dalam machine learning yang mengacu pada satu siklus yang berhasil melalui keseluruhan data pada dataset pada proses *training* model [29]. Setiap satu siklus *dataset* / satu *epoch* selesai, maka akan tersedia data yang dapat dianalisa untuk mengetahui pola yang muncul ketika proses *training* berjalan. *Epoch* merupakan unit yang mengatur berapa kali model akan memperbarui bobotnya untuk keseluruhan *dataset*.



Gambar 2. 1 Pola grafik *Epoch* saat *training* [30]

Gambar 2.1 menunjukkan pola yang dapat dianalisa ketika melakukan *training*. Nilai *epoch* yang terlalu sedikit dapat menyebabkan *underfitting* yaitu performa model kurang baik karena perlu data yang lebih banyak. Sementara, nilai *epoch* yang terlalu banyak dapat membuat *overfitting*, yaitu model belajar terlalu mendalam yang menyebabkan

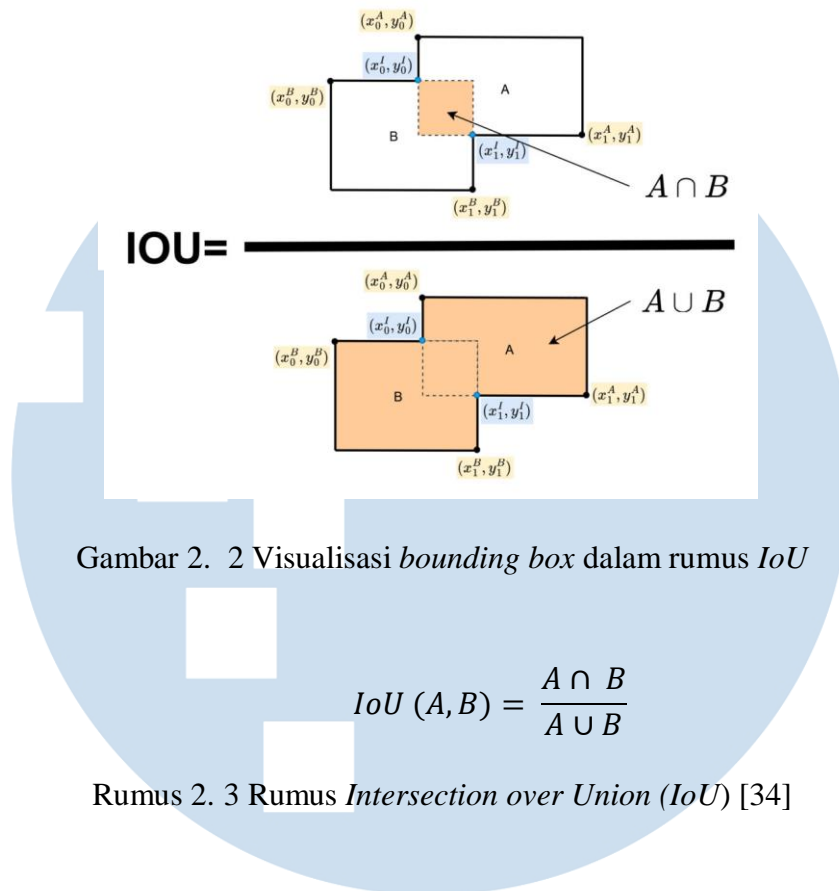
performa model memberikan hasil yang tidak umum ketika validasi/*testing* karena model hanya menghafal dan bukan mempelajari pola ketika melakukan *training* [31].

### 2.2.9 *mAP50* dan *mAP50-95*

*mAP* atau yang merupakan singkatan dari *mean Average Precision at Intersection over Union (IoU) threshold* merupakan suatu metrik evaluasi yang digunakan secara umum untuk mengetahui kemampuan model dalam melakukan deteksi dan klasifikasi objek secara akurat pada *threshold IoU*. *mAP* mengukur nilai rata-rata dari *Average Precision (AP)* untuk tiap *class* dalam *dataset* [32]. Nilai hasil *mAP* dapat menggambarkan baiknya performa model dalam melakukan deteksi objek dengan memperhatikan nilai *precision* dan *recall*. Nilai *mAP50* menggambarkan toleransi *IoU* yang digunakan untuk mengukur *mAP* setidaknya 50%, sementara *mAP50-95* menggambarkan toleransi *IoU* untuk mengukur *mAP* setidaknya 50-95%

*Intersection over Union (IoU)* merupakan suatu metrik untuk mengetahui ukuran tingkat *overlap* antara *bounding box* sebenarnya / *ground truth*, dengan *bounding box* hasil prediksi [33]. Perhitungan *IoU* dilakukan dengan membagi area irisan antar kedua *bounding box* yang terjadi *overlap*, dengan area gabungan/*overlap* antar kedua *bounding box*. Nilai *IoU* menunjukkan persentase terjadinya *overlap* antar hasil *bounding box* prediksi dan *groundtruth*. Berikut merupakan rumus untuk menghitung *IoU*:

U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A



Gambar 2. 2 Visualisasi *bounding box* dalam rumus *IoU*

$$IoU(A, B) = \frac{A \cap B}{A \cup B}$$

Rumus 2. 3 Rumus *Intersection over Union (IoU)* [34]

### 2.2.10 *Box Loss*

*Box Loss* merupakan metrik yang digunakan untuk mengetahui seberapa baik *bounding box* hasil prediksi model dengan *ground truth*. Nilai *box loss* menggambarkan kesalahan antara hasil koordinat prediksi *bounding box* dengan *ground truth* [35].

$$Box Loss = \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{obj} \left[ (x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 + (\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2 \right]$$

Rumus 2. 4 Rumus *Box Loss* [36]

Keterangan:

- $\lambda_{coord}$  adalah parameter bobot konstan yang mengukur pentingnya *box loss*
- $S$  adalah jumlah *grid cell* dalam gambar

- B jumlah bounding box terprediksi dalam tiap grid cell.
- $1_{ij}^{obj}$  adalah fungsi indicator yang sama dengan 1 bila bounding box ke-j dalam cell-i berisi objek, dan bernilai 0 jika sebaliknya
- $x_i, y_i$  merupakan koordinat tengah dari ground truth bounding box.
- $\hat{x}_i, \hat{y}_i$  merupakan koordinat tengah dari predicted bounding box.
- $w_i, h_i$  merupakan panjang dan lebar dari ground truth bounding box.
- $\hat{w}_i, \hat{h}_i$  merupakan panjang dan lebar dari predicted bounding box.

### 2.2.11 Class Loss

*Class Loss* merupakan metrik yang digunakan untuk mengetahui besarnya kesalahan dalam melakukan klasifikasi objek/ ketidakakuratan deteksi objek pada tiap prediksi [35].

$$Class\ Loss = \sum_{i=0}^{S^2} \mathbb{1}_i^{obj} \sum_{c \in classes} (\mathcal{P}_i(c) - \hat{\mathcal{P}}_i(c))^2$$

Rumus 2. 5 Rumus Class Loss [37]

Keterangan:

- $S^2$  adalah *grid* yang digunakan oleh YOLO untuk membagi *input image*.
- $\mathcal{P}_i(c)$  adalah jumlah *ground truth* atau *true probability* dari class  $c$  pada grid cell  $i$ .
- $\hat{\mathcal{P}}_i(c)$  adalah jumlah *predicted probability* dari class  $c$  pada grid cell  $i$  yang merupakan hasil dari output model.

### 2.2.12 Distribution Focal Loss

*Distribution Focal Loss* atau DFL merupakan metrik yang digunakan untuk mengatasi ketidakseimbangan untuk tiap kelas yang ada pada dataset dengan tujuan untuk meningkatkan performa model [38].

$$DFL(S_i, S_{i+1}) = -((\psi_{+1} - \psi) \log(S_i) + (\psi - \psi_i) \log(S_{i+1}))$$

## Rumus 2. 6 Rumus Distribution Focal Loss [39]

Keterangan:

- $S_i, S_{i+1}$  adalah parameter yang mewakili probabilitas prediksi dari *discrete points*.
- $y$  adalah jumlah *ground truth label* atau *actual value* yang ingin di prediksi oleh model.

### 2.3 Framework/ Algoritma yang digunakan

#### 2.3.1 YOLO

YOLO (*You Only Look Once*) adalah sebuah *framework* atau kerangka kerja untuk deteksi objek dalam pengolahan citra dan video. YOLO dikembangkan untuk memberikan deteksi objek secara cepat dan efisien dengan mengidentifikasi objek dalam satu proses pengolahan gambar, tanpa memerlukan langkah-langkah yang kompleks atau proses yang berulang [40].

Fitur utama dari YOLO adalah kemampuannya untuk mendeteksi objek secara *real-time* dengan tingkat akurasi yang tinggi. YOLO menggunakan jaringan saraf tiruan (*neural network*) untuk melakukan deteksi objek [41]. Versi-versi yang berbeda dari YOLO telah dikembangkan seiring waktu, seperti YOLOv1, YOLOv2 (atau YOLO9000), YOLOv3, dan seterusnya.

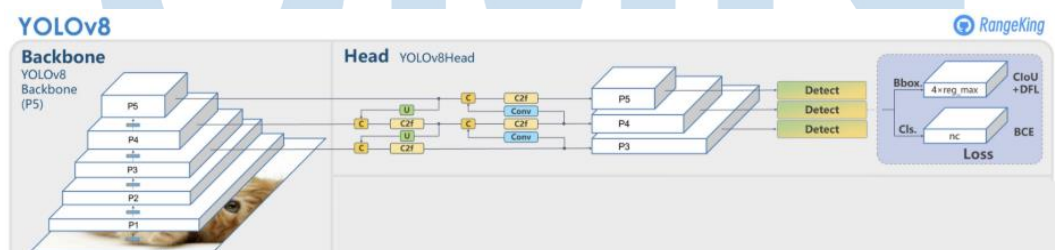
Berikut adalah beberapa karakteristik utama YOLO[42] :

- Deteksi Objek Secara *Real-time*: YOLO dirancang untuk memberikan deteksi objek dalam waktu nyata, menjadikannya cocok untuk aplikasi yang memerlukan respons cepat, seperti deteksi objek dalam video atau di lingkungan waktu nyata.
- *Single Forward Pass*: YOLO melakukan prediksi objek hanya dalam satu kali proses maju (*forward pass*) melalui jaringan saraf, yang membuatnya efisien dan cepat.

- *Grid-based Detection*: Citra dibagi menjadi grid, dan setiap grid digunakan untuk memprediksi objek dalam sel tersebut. Ini memungkinkan YOLO untuk mendeteksi *multiple object* dalam satu citra.
- Skala Multi-resolusi: YOLO mendukung deteksi objek pada berbagai skala resolusi, memungkinkan penggunaan pada gambar dengan resolusi yang berbeda.
- Dukungan untuk banyak kelas: YOLO mampu mendeteksi objek dari berbagai kelas dalam satu citra.
- YOLO telah menjadi populer dalam berbagai aplikasi termasuk pemantauan video, kendaraan otonom, pengawasan keamanan, dan lainnya karena kemampuannya untuk memberikan deteksi objek secara cepat dan efisien.

### 2.3.2 YOLOv8

YOLOv8 adalah *framework* YOLO terbaru yang dikembangkan oleh Ultralytics yang dapat digunakan untuk deteksi objek, klasifikasi gambar, dan tugas segmentasi instans. YOLOv8 diluncurkan pada 10 Januari 2023, dan terus mengalami pembaruan dan pengembangan aktif oleh *developer*. YOLOv8 mencakup banyak perubahan dan peningkatan pengalaman arsitektur oleh *developer* yang sama dengan *developer* yang menciptakan versi YOLOv5.



Gambar 2. 3 Arsitektur Algoritma YOLOv8 [43]

YOLOv8 merupakan perbaikan dari versi YOLO sebelumnya, pada versi ini digunakan teknologi arsitektur *Neural Network* yang terbaru yaitu

*Feature Pyramid Network (FPN)*, *Path Aggregation Network (PAN)*, serta teknologi untuk menyederhanakan proses anotasi dan penambahan label dalam deteksi objek [43]. Dengan implementasi teknologi *FPN* dan *PAN* pada arsitektur YOLOv8, maka model mampu melakukan penangkapan objek yang lebih baik pada skala dan resolusi yang berbeda, sehingga akurasi deteksi objek meningkat meskipun deteksi dilakukan pada ukuran dan bentuk gambar yang berbeda[11].

Performa YOLOv8 teruji lebih cepat dan akurat melalui *API* yang ramah pengguna (*Command + Python*). *YOLOv8* memiliki lima model di setiap kategori deteksi, segmentasi, dan klasifikasi. Berikut beberapa kelebihan dari algoritma YOLOv8 [44] [45]:

- YOLOv8 memiliki performa akurasi yang tinggi, dan lebih baik dibandingkan versi sebelumnya.
- YOLOv8 memiliki *python package* yang lengkap dan terstruktur sehingga memudahkan *developer* dalam tahap pengembangan dan penggunaan CLI yang sederhana.
- Model YOLOv8 berukuran kecil sehingga mudah untuk diimplementasikan dan lebih ringan untuk dijalankan dibandingkan versi YOLO sebelumnya.
- YOLOv8 mampu mendeteksi banyak objek pada gambar dengan akurasi yang baik dan waktu pemrosesan lebih cepat dibandingkan versi YOLO sebelumnya.
- Komunitas YOLOv8 yang luas dan terus berkembang.

## 2.4 Tools yang digunakan

### 2.4.1 Python

Python adalah sebuah bahasa pemrograman yang menjadikan objek sebagai orientasinya dan cocok untuk digunakan dalam berbagai pengembangan pendekatan perangkat lunak maupun *machine learning*[46].

#### 2.4.2 Google Colaboratory

Jupyter Notebook digunakan sebagai alat untuk menampilkan dan menganalisis data yang sudah diolah oleh Python melalui Google Colaboratory (Google Colab). Colab memungkinkan *user* menggunakan dan membagikan jupyter notebook dengan pengguna lain tanpa perlu mendownload, menginstal, atau menjalankan apa pun[47].

#### 2.4.3 Roboflow

Roboflow merupakan sebuah *platform* yang dapat digunakan oleh *developer* untuk membangun model *machine learning* yang bergerak dalam bidang *Computer Vision*. Roboflow menyediakan berbagai *tool* yang mendukung berbagai tahapan/ proses saat membangun model *computer vision*. Melalui Roboflow, dapat dilakukan manajemen dataset, *labeling data*, *data pre-processing* untuk gambar, *data augmentation*, bahkan *training* dan *deployment* [48]. Selain itu, Roboflow menyediakan fitur bernama *Roboflow Universe* yang menyediakan berbagai dataset gambar yang berkualitas dan mendukung untuk proyek *computer vision*.

#### 2.4.4 Flask

Flask merupakan sebuah *micro-framework* yang bersifat ringan, modular, dan sederhana untuk melakukan pengembangan aplikasi *website* berbasis bahasa pemrograman Python. *Framework* Flask menyediakan fungsi dasar untuk membangun aplikasi *web* dan menyediakan berbagai *library* tambahan yang dapat digunakan sesuai dengan kebutuhan pengembangan aplikasi [49]. *Framework* flask umumnya digunakan karena sifatnya yang sederhana dan mudah untuk digunakan, fungsionalitas yang fleksibel, dan mendukung untuk pengembangan aplikasi *web* dalam skala kecil hingga menengah.