

SISTEM DETEKSI KETERSEDIAAN KURSI BERBASIS DEEP LEARNING MENGUNAKAN ALGORITMA YOLO UNTUK MONITORING KETERSEDIAAN TEMPAT DUDUK (STUDI KASUS: JARAK COFFEE & EATERY)

DEEP LEARNING BASED SEAT AVAILABILITY DETECTION SYSTEM USING YOLO ALGORITHM FOR SEATING AVAILABILITY MONITORING (CASE STUDY: JARAK COFFEE & EATERY)

Afian Ibnu Yuwono¹, Veronica Lusiana²

Fakultas Teknologi Informasi Dan Industri, Program Studi Teknik Informatika, Universitas Stikubank
Semarang^{1,2}

Afianibnu0034@mhs.unisbank.ac.id¹

ABSTRACT

The development of Artificial Intelligence (AI) technology, especially in the field of computer vision, enables the development of systems to monitor seat availability automatically. This study aims to develop a deep learning-based seat availability detection system using the YOLOv11 algorithm to monitor seat availability at JARAK COFFEE & EATERY. The problem raised is the lack of a real-time monitoring system that can provide accurate information about empty seats, which can have an impact on decreasing service efficiency and customer satisfaction. The objectives of the study are to create a detection model that can recognize "chair" and "person" objects, implement Intersection over Union (IoU)-based spatial relationship analysis logic to determine seat occupancy status, and build a system prototype that processes CCTV video recording datasets. The system development method uses the Waterfall approach, starting from video data collection, dataset labeling, YOLOv11 model training, to implementation and testing. The test results show that the developed model has good performance with a mean Average Precision (mAP@0.5) value of 95.7%. The system successfully detects objects and calculates available seats in real-time by utilizing IoU logic, although performance can be affected by factors such as lighting and occlusion between objects.

ABSTRAK

Perkembangan teknologi *Artificial Intelligence* (AI) khususnya dalam bidang *computer vision* memungkinkan pengembangan sistem untuk memantau ketersediaan tempat duduk secara otomatis. Penelitian ini bertujuan mengembangkan sistem deteksi ketersediaan kursi berbasis *deep learning* menggunakan algoritma YOLOv11 untuk memonitor ketersediaan tempat duduk di JARAK COFFEE & EATERY. Permasalahan yang diangkat adalah belum adanya sistem pemantauan *real-time* yang dapat memberikan informasi akurat mengenai kursi kosong, yang dapat berdampak pada penurunan efisiensi pelayanan dan kepuasan pelanggan. Tujuan penelitian adalah membuat model deteksi yang dapat mengenali objek "kursi" dan "orang", mengimplementasikan logika analisis hubungan spasial berbasis *Intersection over Union* (IoU) untuk menentukan status keterisian kursi, serta membangun prototipe sistem yang memproses dataset video rekaman cctv. Metode pengembangan sistem menggunakan pendekatan *Waterfall*, dimulai dari pengumpulan data video, pelabelan dataset, pelatihan model YOLOv11, hingga implementasi dan pengujian. Hasil pengujian menunjukkan model yang dikembangkan memiliki kinerja yang baik dengan nilai *mean Average Precision* (mAP@0.5) sebesar 95,7%. Sistem berhasil mendeteksi objek dan menghitung kursi tersedia secara *real-time* dengan memanfaatkan logika IoU, meskipun performa dapat dipengaruhi oleh faktor seperti pencahayaan dan oklusi antar objek.

Kata Kunci: *algoritma YOLO, Ketersediaan Kursi, Object Detection, Intersection Over Union*

PENDAHULUAN

Perkembangan teknologi *Artificial Intelligence* (AI) khususnya dalam sub-bidang *computer vision*, sehingga memungkinkan mesin untuk menganalisis dan memahami data visual layaknya manusia. *Artificial Intelligence* (AI) adalah

cabang ilmu komputer yang bertujuan untuk menciptakan sistem yang dapat melakukan tugas – tugas yang memerlukan kecerdasan manusia. Salah satu cabang *Artificial Intelligence* (AI) yang relevan dengan penelitian ini adalah *deep learning*, *deep learning* adalah pendekatan yang

menggunakan jaringan syaraf yang dalam dan kompleks untuk memproses dan memahami teks (Sehan, 2025).

Optimasi pelayanan menjadi kunci untuk meningkatkan kepuasan pelanggan dan daya saing *coffee shop*. Salah satu tantangan yang sering dihadapi pemilik sebuah *coffee shop* adalah dalam memantau ketersediaan tempat duduk. Pada jam-jam sibuk, tempat ini kerap mengalami fluktuasi pengunjung yang signifikan. Jika tidak ada tempat, maka mereka harus kembali atau menunggu sampai menemukan tempat duduk. Hal ini menyita banyak waktu di tengah jadwal mereka yang sudah padat (Siddhi Pravin Lipare & Aryan Jain, 2022). Informasi mengenai ketersediaan tempat duduk bagi pelanggan memiliki peran penting dalam optimalisasi pelayanan, perencanaan kapasitas tempat duduk, serta peningkatan pengalaman pelanggan (Mulyana, 2025).

Dalam upaya meningkatkan efisiensi pelayanan, akan sangat membantu jika mengetahui sebelumnya apakah ada meja atau kursi tersebut kosong atau tersedia. Sehingga, penggunaan algoritma YOLO dalam pengembangan sistem ini sangat dibutuhkan, karena algoritma YOLO dapat mendeteksi objek secara real-time dengan akurasi yang tinggi, dan proses kalkulasi yang sangat cepat dalam menghasilkan hasil deteksi (Dyah & Ricky, 2025). Dengan kemajuan algoritma deep learning untuk deteksi objek, dimungkinkan untuk membuat sistem yang tidak hanya dapat mendeteksi objek (manusia dan kursi) tetapi juga memahami konteks dan hubungan spasial antara mereka.

Penelitian yang dilakukan (Bima & Fetty, 2021) menunjukkan bahwa YOLO dapat diintegrasikan dengan kamera pengawas untuk mendeteksi jumlah orang yang berada dalam suatu area secara

akurat. Hal ini menunjukkan bahwa teknologi YOLO sangat potensial diterapkan dalam konteks *coffee shop* untuk mendeteksi kedatangan pelanggan dan ketersediaan tempat duduk.

Dengan adanya sistem ini, diharapkan pemilik JARAK COFFEE & EATERY memperoleh data yang lebih akurat mengenai ketersediaan tempat duduk pada setiap meja, sehingga dapat meningkatkan kualitas layanan dan efisiensi pengelolaan pelanggan.

TINJAUAN PUSTAKA

Sistem

Sistem menurut Andrianof dalam (Maydianto, 2021) adalah gabungan dari beberapa elemen, komponen atau variabel yang saling terintegrasi guna untuk membentuk sebuah satu kesatuan sehingga dapat tercapainya suatu tujuan dan sasaran. Sistem menurut Wijoyo, dkk (2021:4) adalah sekelompok elemen-elemen yang saling berinteraksi untuk mencapai tujuan tertentu. Menurut Ridwan dan Widiastiwi (2021:19) sistem adalah dasar dari segala aktivitas, dimana keberadaan sistem sangat penting dalam setiap bidang. Tanpa adanya sistem, kegiatan atau pekerjaan akan berjalan tanpa arah dan kontrol.

Dari beberapa pernyataan di atas mengenai pengertian sistem dapat disimpulkan bahwa sistem adalah gabungan dari kumpulan elemen, aktifitas, atau variabel yang membentuk suatu kesatuan guna mencapai suatu tujuan dan sasaran tertentu.

Deteksi Objek

Deteksi objek merupakan salah satu teknik untuk menemukan objek dalam gambar atau video. Deteksi objek adalah teknik Computer Vision untuk dapat menemukan contoh objek dalam gambar atau video. Salah satu metode untuk membuat deteksi objek adalah menggunakan TensorFlow Object Detection API yang menyediakan berbagai model pra terlatih. Model pra terlatih yang

dapat digunakan dalam pembuatan deteksi objek adalah SSD Mobilenet V2. Model tersebut dapat melakukan deteksi objek dengan menghasilkan akurasi dan area terdeteksi untuk keberadaan setiap kategori objek pada suatu gambar. Model ini menghasilkan besar akurasi untuk keberadaan setiap kategori objek (Aningtiyas. P. R, Sumin. A, 2020).

Proses pendeteksian objek dengan gambar ataupun video memiliki kesamaan dimana sebuah video terdiri dari beberapa susunan gambar atau frame. Proses identifikasi objek dalam video dilakukan menggunakan *Graphics Processing Unit* (GPU) dengan membagi video menjadi beberapa frame kemudian diproses dan dirangkai kembali menjadi video yang utuh.

Machine Learning

Menurut Arthur Samuel, Machine Learning adalah salah satu bidang ilmu komputer yang memberikan kemampuan pembelajaran kepada komputer untuk mengetahui sesuatu tanpa pemrograman yang jelas.

Didalam algoritma Machine Learning terbagi menjadi 3 kategori utama yaitu:

1. Supervised Learning

Supervised Learning merupakan pembelajaran menggunakan masukan data pembelajaran yang telah diberi label. Tujuannya adalah untuk membangun fungsi yang mampu mengeneralisir data yang belum dilihat sebelumnya.

2. Unsupervised Learning

Unsupervised Learning merupakan pembelajaran menggunakan masukan data pembelajaran yang tidak diberi label. Tujuan metode ini adalah untuk membangun fungsi mapping yang mengategorikan data menjadi beberapa

kelas berdasarkan ciri yang tersembunyi dalam data yang diproses.

3. Reinforcement Learning

Reinforcement Learning merupakan fase pembelajaran dan tes saling dicampur. Selama proses learning, algoritma akan secara acak mengeksplorasi pasangan keadaan-tindakan dalam beberapa lingkungan (untuk membangun tabel pasangan keadaan-tindakan).

Deep Learning

Pembelajaran mendalam (*Deep Learning*) merupakan cabang dari pembelajaran mesin (*machine learning*) yang berfokus pada penggunaan jaringan saraf tiruan (*artificial neural networks*) dengan lapisan yang mendalam (*deep layers*) untuk memproses, menganalisis, dan memahami data yang kompleks (Huang, Chai and Cho, 2020). Deep learning memiliki kemampuan luar biasa untuk menangani data yang tidak terstruktur, seperti gambar, suara, dan teks, dengan performa yang dapat melampaui metode pembelajaran tradisional (Muhammadiyah Mataram et al., 2024).

Deep learning telah mengubah berbagai bidang teknologi dan memberikan solusi inovatif untuk beragam masalah. Salah satu aplikasi utamanya adalah dalam *computer vision* seperti pengenalan objek, deteksi wajah, dan segmentasi gambar (Rayed et al., 2024).

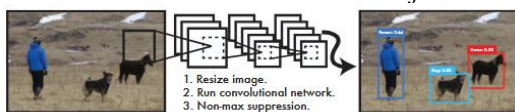
Algoritma

Algoritma didefinisikan sebagai usaha dengan urutan operasi yang disusun secara logis dan sistematis untuk menyelesaikan suatu masalah untuk menghasilkan suatu output tertentu (Kani, 2020).

You Only Look Once (YOLO)

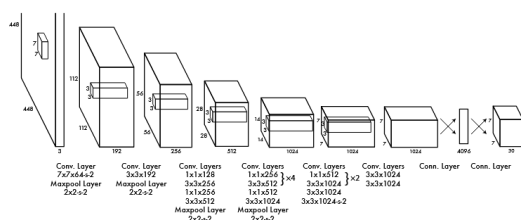
Berikut adalah cara yolo dalam memproses gambar:

1. Mengubah ukuran gambar input menjadi 448×448 .
2. Menjalankan jaringan konvolusional tunggal (*single convolutional network*) pada gambar.
3. Melakukan *threshold* pada hasil deteksi berdasarkan nilai *confidence*.



Gambar 1. Sistem deteksi YOLO
Sumber: Redmon et al

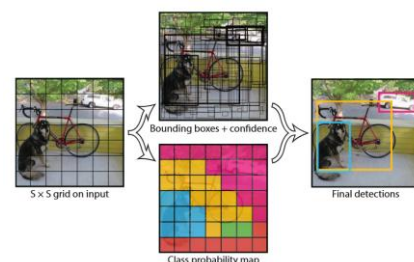
Jaringan deteksi YOLO memiliki 24 lapisan konvolusi (*convolutional layer*) yang diikuti oleh 2 lapisan yang terhubung penuh (*fully connected layer*). Beberapa lapisan konvolusi menggunakan lapisan reduksi 1x1 sebagai alternative dalam mengurangi kedalaman feature maps yang diikuti oleh 3x3 lapisan konvolusional (*convolutional layer*).



Gambar 2. Arsitektur YOLO
Sumber: Redmon et al

Setiap bounding box YOLO terdiri dari 5 prediksi: x , y , w , h dan *confidence*. Nilai dari koordinat(x,y) merepresentasikan titik pusat dari kotak terhadap batas grid cell. Lebar (*width*) dan tinggi (*height*) diprediksi berdasarkan data relatif dari keseluruhan gambar. Terakhir, nilai keyakinan (*confidence score*) merepresentasikan IoU antara kotak prediksi dan setiap *ground truth box*.

Jika pusat suatu objek jatuh ke dalam sel grid, maka sel grid bertanggung jawab untuk mendeteksi objek tersebut. Setiap sel grid memprediksi kotak pembatas (*bounding boxes*) B dan *confidence score* untuk kotak tersebut, serta probabilitas kelas kondisional C , $\Pr(Class_i|Object)$. YOLO hanya memprediksi satu set dari kelas probabilitas per grid cell, terlepas dari jumlah kotak.



Gambar 3. Proses Deteksi YOLO
Sumber: Redmon et al

YOLO membagi gambar input menjadi grid $S \times S$, dan setiap grid cell memprediksi B bounding boxes, confidence untuk setiap box, dan probability kelas C . Prediksi ini dikodekan sebagai $S \times S \times (B \times 5 + C)$ tensor.

Intersection Over Union (IoU)

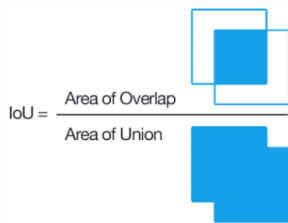
IoU (*Intersection over Union*) adalah metrik evaluasi yang fundamental dalam tugas object detection yang digunakan untuk mengukur seberapa tumpang tindih (*overlap*) antara dua area batas, yaitu kotak batas prediksi (*Predicted Bounding Box*)

dan kotak batas sebenarnya (*Ground Truth Bounding Box*).

$$IoU = \frac{\text{Area Irisan (Intersection)}}{\text{Area Gabungan (Union)}} \quad (1)$$

Keterangan:

- Nilai IoU selalu berkisar antara 0 hingga 1.
- $IoU = 1$, berarti bounding box terprediksi akurat
- $IoU = 0$, Tidak ada bounding box sama sekali.
- $IoU > 0.5$, Biasanya dianggap sebagai deteksi yang valid.



Gambar 4. Intersection Over Union

Sumber: Redmon et al

Pada fase *training*, IoU digunakan untuk menentukan nilai *confidence score* dan menghitung fungsi kerugian (*loss function*). Untuk menentukan *confidence score* yang akan digunakan untuk menghitung:

$$\text{Confidance} = \text{Pr}(\text{Object}) \times IoU_{pred}^{truth} \quad (2)$$

Keterangan:

- $\text{Pr}(\text{Object})$ adalah probabilitas bahwa kotak batas mengandung objek.
- IoU_{pred}^{truth} adalah IoU antara kotak prediksi dan kotak ground truth.

Confusion Matrix

Confusion matrix atau *error matrix* adalah ringkasan hasil prediksi pada permasalahan klasifikasi. Jumlah klasifikasi yang benar dan yang salah dikumpulkan dengan nilai hitung kemudian dipecah oleh setiap kelas. Sehingga *confusion matrix*

tidak hanya memberikan informasi kesalahan yang dibuat classifier tetapi juga jenis kesalahannya (Umar et al., 2020).

Precision

Nilai presisi dapat dihitung dengan cara membagi total sampel positif yang diklasifikasikan dengan benar dengan total sampel positif yang diprediksi dengan presisi tinggi untuk menunjukkan contoh berlabel positif memang positif (FP) (Umar et al., 2020).

$$\text{Precision} = \frac{TP}{TP + FP} \quad (3)$$

Keterangan:

- True Positive (TP): aktual bernilai positif, dan diprediksi positif.
- False Positive (FP): aktual bernilai negatif, tetapi diprediksi negative.

Recall

Recall adalah rasio dari total keseluruhan sampel positif yang diklasifikasikan dengan benar kemudian dibagi dengan total sampel positif. High recall menunjukkan bahwa kelas dideteksi dengan benar (FN).

$$\text{Recall} = \frac{TP}{TP + FN} \quad (4)$$

Keterangan:

- High Recall, Low Precision: beberapa sampel positif yang dapat dikenali (FN) tetapi terdapat banyak positif palsu.
- Low Recall, High Precision: kehilangan banyak sampel positif (FN tinggi) namun yang diprediksi positif benar-benar positif (FP rendah).

Mean Average Precision

Mean Average Precision (mAP) merupakan metrik evaluasi utama untuk mengukur kinerja sistem deteksi objek secara keseluruhan. Nilai mAP dihitung dengan merata-ratakan nilai *Average*

Precision (AP) dari semua kelas objek. Sementara itu, setiap nilai AP diperoleh melalui perhitungan dan analisis kurva precision-recall, yang didasarkan pada formula perhitungan precision dan recall yang telah ditetapkan (Umar et al., 2020).

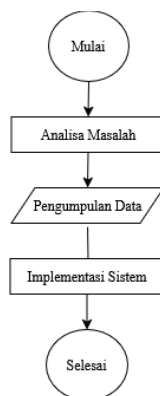
$$mAP = \frac{1}{n} \sum_{i=1}^n AP_i \quad (5)$$

Keterangan:

- a. n : Banyak Kelas.
- b. AP : *Average Precision*

METODE

Flowchart Penelitian



Gambar 5. Flowchart Penelitian

Sumber: Penelitian

Pada gambar 5. dapat dilihat serangkaian alur atau *flowchart* yang ada dalam penelitian ini.

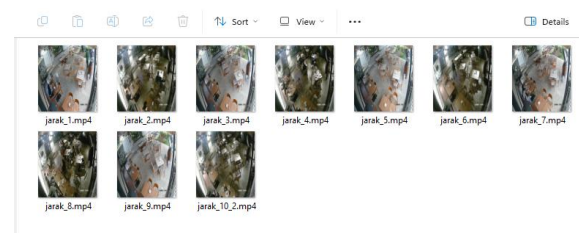
Analisis Masalah

Penelitian ini berfokus pada pengembangan sistem deteksi ketersediaan kursi di lingkungan JARAK COFFEE & EATERY menggunakan algoritma YOLOv11 berbasis deep learning. Masalah utama yang dihadapi adalah kurangnya sistem pemantauan real-time yang dapat memberikan informasi akurat mengenai ketersediaan tempat duduk kepada pengelola maupun pelanggan. Pada kondisi sibuk, pelanggan seringkali harus menunggu atau bahkan pergi karena tidak tersedianya kursi kosong, yang berpotensi

menurunkan kepuasan pelanggan dan efisiensi operasional.

Pengumpulan Data

Data diperoleh langsung melalui akses cctv yang didapat pada JARAK COFFEE & EATERY. Data ini secara spesifik berupa rekaman video berdurasi sekitar 1jam sampai 1 jam 30 menit yang diambil di JARAK COFFEE & EATERY, menangkap interaksi pelanggan dengan kursi pada waktu-waktu yang berbeda, termasuk saat sepi maupun ramai.

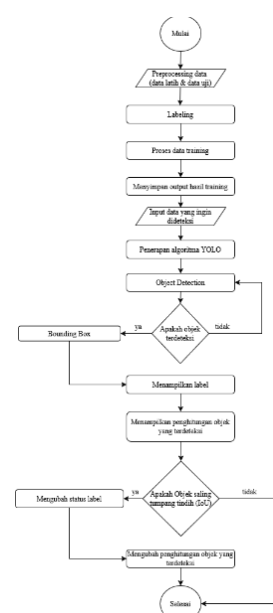


Gambar 6. Data Uji

Sumber: Dokumentasi Penelitian

Video-video ini akan menjadi dataset utama yang sangat krusial pada saat proses training model deep learning YOLOv11, serta sebagai data uji coba (*environment*) untuk mengevaluasi performa sistem yang dibangun.

Implementasi Sistem



Gambar 7. Flowchart Implementasi Sistem

Sumber: Dokumentasi Penelitian

Keterangan:

1. Preprocessing Data

Preprocessing Data (Data Latih & Data Uji) merupakan tahapan dalam pengolahan data video sebelum data tersebut digunakan untuk analisis lebih lanjut seperti object detection. Adapun beberapa langkah dalam preprocessing data adalah dimulai dari pemisahan frame yaitu memisah video menjadi frame-frame individual sehingga setiap frame dapat diproses secara terpisah. Selanjutnya mengubah resolusi frame sehingga sesuai dengan kebutuhan analisis dan konsisten dengan model yang digunakan. Setiap gambar atau frame video diproses secara berurutan menggunakan model YOLO. Pada tahap ini akan dilakukan ekstraksi dari video diubah menjadi bentuk gambar per frame secara berulang untuk mendeteksi objek. Setiap frame video akan dianalisis menggunakan model YOLO untuk mendeteksi objek yang ada pada frame tersebut.

2. Labeling

Setelah dilakukan pengumpulan gambar dan video, maka proses selanjutnya adalah melakukan labeling pada video. Video yang telah disimpan dalam folder akan dimasukkan satu persatu ke dalam aplikasi Roboflow yang akan melakukan penyeleksian pada objek yang akan dideteksi nantinya. Setiap video diberikan label berupa bounding box dan kelas objek yang ingin dideteksi.

3. Data Trining

Proses *training* adalah proses melatih jaringan syaraf tiruan untuk mengnali atau mempelajari pola sehingga dapat mengnali deteksi objek sesuai dengan yang diharapkan dengan tingkat akurasi yang tinggi. Setelah proses labeling pada video selesai dilakukan, setiap objek yang telah dipilih akan diberikan label dan format deteksinya dikonversi ke standar YOLOv11. Selanjutnya, melalui proses pelatihan model, akan dihasilkan berkas berformat best.pt yang merupakan keluaran dari model YOLOv11.

4. Menyimpan Data Trining

Data yang telah dilabeli kemudian berubah kedalaam format file .pt, dimana informasi video tersebut diubah menjadi kode-kode terstruktur yang dapat diproses oleh model YOLOv11. Kode-kode tersebut kemudian dimasukkan ke dalam program pelatihan YOLOv11 untuk dijadikan sebagai basis atau acuan dalam sistem pendeteksian yang akan dibangun.

5. Object Detection

Setiap frame dari video diproses menggunakan model YOLO yang telah dilatih sebelumnya. YOLO akan menganalisis setiap frame untuk mendeteksi objek-objek yang sebelumnya telah ditentukan pada proses labelling.

6. Bounding Box

Bounding box merupakan kotak imajiner yang membatasi area di sekitar objek yang teridentifikasi. Bounding box sendiri berbentuk kotak yang dimana besarnya sama seperti besar objek yang teridentifikasi tersebut. Pada penelitian ini dilakukan pengaturan bounding box dengan *thickness* = 2. Parameter *thickness* = 2 mengacu pada ketebalan garis tepi (outline) dari objek pada gambar.

7. Penghitungan Objek

Setelah deteksi objek dilakukan pada setiap frame dalam video, objek yang terdeteksi dihitung berdasarkan kriteria yang sudah ditetapkan pada proses labeling. Misalnya setiap kursi terdeteksi maka sistem akan mengaplikasikan penghitungan untuk kategori kursi tersebut.

$$\text{Rasio Deteksi} = \frac{\text{Jumlah Obek Terdeteksi}}{\text{Ob jek Deteksi}} \times 100 \quad (6)$$

Keterangan:

- Jumlah Objek Terdeteksi: jumlah objek terdeteksi oleh model dengan kepercayaan di atas *threshold* yang telah ditentukan.
- Total Deteksi: jumlah total deteksi yang dilakukan oleh model, termasuk deteksi yang benar dan salah.

8. Mengubah Hasil Penghitungan Objek Berdasarkan Logika IoU

Penghitungan objek dalam sistem ini dapat berubah secara dinamis berdasarkan hasil analisis *Intersection over Union* (IoU) antara bounding box objek. Setelah model YOLO melakukan deteksi awal dan menampilkan jumlah objek yang teridentifikasi sistem kemudian menganalisis tumpang tindih spasial antara kedua kelas objek tersebut. Perubahan status ini secara langsung memengaruhi penghitungan akhir objek dengan jumlah kursi yang tersedia dikurangi sesuai dengan banyaknya kursi yang terisi, sementara jumlah orang tetap sama.

Proses Implementasi Algoritma YOLOv11

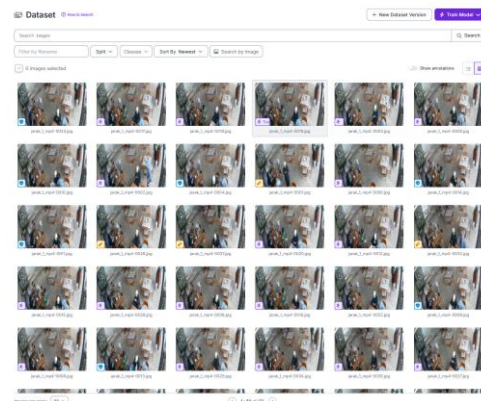
YOLOv11 merupakan versi lanjutan dari algoritma YOLO (*You Only Look Once*), dirancang untuk mendeteksi objek dalam gambar secara semi real-time dengan tingkat akurasi dan kecepatan yang tinggi. Perbedaan signifikan pada YOLOv11 terletak pada optimalisasi struktur jaringan dan mekanisme pelatihan yang lebih efisien, sehingga memungkinkan deteksi yang lebih cepat tanpa mengorbankan akurasi. Dibandingkan dengan YOLOv8, YOLOv11 menawarkan keunggulan dalam hal kecepatan inferensi, efisiensi memori, serta peningkatan akurasi pada objek dengan ukuran kecil dan kompleksitas tinggi. Selain itu, model ini lebih stabil dalam proses pelatihan dan memiliki kemampuan generalisasi yang lebih baik terhadap data yang bervariasi (Nofita Sary et al., 2025).

HASIL DAN PEMBAHASAN Implementasi

Pada tahap implementasi merupakan tahap penerapan sistem atau aplikasi yang telah dibuat. Pada penelitian ini implementasi diterapkan pada sistem monitoring pelanggan untuk menentukan ketersediaan tempat duduk menggunakan algoritma YOLOv11 pada JARAK COFFEE & EATERY.

Labeling Objek

Labeling Objek adalah proses memberi label pada data mentah (gambar, teks, video, dll.) agar model machine learning dapat mengenali dan mengklasifikasi objek tersebut. Tahap pelabelan objek dilakukan ketika dataset yang awalnya berupa rekaman video CCTV dari JARAK COFFEE & EATERY telah diolah menjadi gambar.



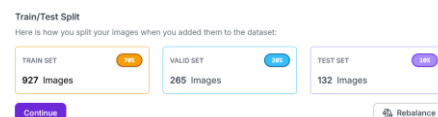
Gambar 8. Dataset Gambar

Sumber: Dokumentasi Penelitian

Pada proses pelabelan terdapat 1324 gambar untuk dijadikan dataset seperti pada gambar 8. yang nantinya akan digunakan untuk tahap pelabelan. Pada tahap pelabelan peneliti melakukan proses pelabelan pada frame video setiap 30 detik.

Pembagian Dataset

Dataset yang telah terkumpul akan dibagi menjadi 70% dataset *training*, 20% dataset *validating*, dan 10% dataset *testing* seperti pada gambar 9.



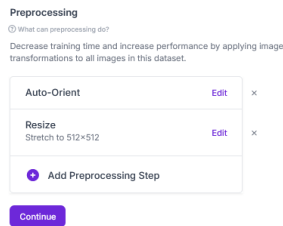
Gambar 9. Pembagian dataset

Sumber: Dokumentasi Penelitian

Resize

Resize adalah proses mengubah gambar awal menjadi ukuran yang dapat diterima oleh model, ukuran citra diubah menjadi 512x512 pixel dengan ketentuan *Auto-Orient* seperti pada gambar 10. untuk

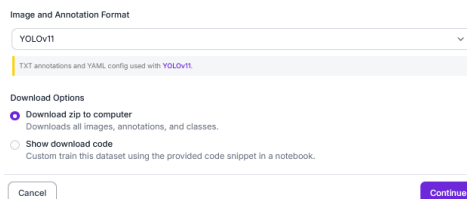
memastikan konsistensi pada dimensi gambar.



Gambar 10. Resize Dataset
Sumber: Dokumentasi Penelitian

Export Dataset

Dataset yang telah dilakukan proses resize akan di-export ke dalam format model YOLOv11 seperti pada gambar 11.

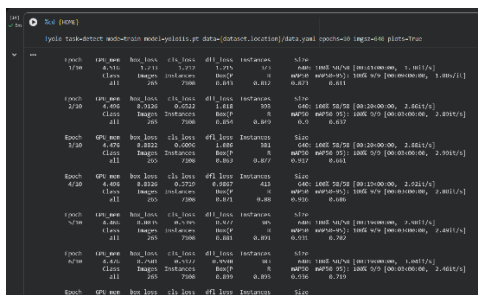


Gambar 11. Export Dataset
Sumber: Dokumentasi Penelitian

Trining Data

Proses ini bertujuan untuk menginisiasi pelatihan pada sistem komputasi melalui pemrosesan data citra dan label yang telah dikumpulkan. Melalui proses ini, sistem diarahkan untuk mengekstraksi dan membentuk pola atau fitur karakteristik unik dari setiap kelas.

Karena pada penelitian ini proses komputasi yang berat saat melakukan proses training, maka peneliti menggunakan *Google Colaboraty* yang merupakan sebuah platform yang disediakan oleh *Google*.



Gambar 12. Trining Model YOLOv11
Sumber: Dokumentasi Penelitian

Pada gambar 12. informasi yang dapat diperoleh selama proses training adalah jumlah *epoch* yang sudah ditetapkan yaitu berjumlah 10 *epoch* yang artinya learning model bekerja dalam keseluruhan dataset tersebut sebanyak 10 *batch* (10 *cycles*) dan setiap 1 *batch epoch* memerlukan waktu 20 detik untuk melakukan proses *training*. Selain itu, pada gambar tersebut memperlihatkan nilai *precision*, *recall*, *mAP50* yang meningkat sering bertambahnya *epoch*. Setelah proses *trining* custom dataset dilakukan akan dihasilkan sebuah file *weight* bernama *best.pt* yang merupakan file untuk pendeteksian objek.

Deteksi Objek Menggunakan algoritma YOLOv11

Proses deteksi objek menggunakan algoritma YOLOv11 diawali dengan persiapan model *best.pt* dan dataset dalam satu folder yang akan digunakan untuk aplikasi deteksi. Tahap pertama melibatkan import berbagai library penting seperti *os*, *sys*, *argparse*, *cv2*, *numpy*, dan *ultralytics* YOLO untuk menangani operasi sistem, parsing argumen command line, pemrosesan gambar/video, serta eksekusi model deteksi.

Untuk konsistensi deteksi objek, sistem menerapkan mekanisme matching menggunakan perhitungan *Intersection over Union* (IoU) dengan *threshold* 0.3 untuk mendefinisikan deteksi baru dengan track yang sudah ada. Fitur persistensi objek digunakan untuk mempertahankan kursi yang sementara tidak terdeteksi selama beberapa frame tertentu.

Pada gambar 13. ditunjukkan kode program untuk perhitungan jumlah kursi kosong dengan membandingkan tumpang tindih antara *bounding box* orang dan kursi menggunakan IoU *threshold* 0.1. Hasil deteksi ditampilkan secara visual dengan *bounding box* berwarna dan teks yang menunjukkan jumlah total kursi serta kursi yang tersedia.

[illegible]

Gambar 13. Kode Program Pengaplikasian YOLOv11

Sumber: Dokumentasi Penelitian

Pengujian Performa Model YOLOv11

Pengujian ini bertujuan untuk mengukur tingkat akurasi model weight YOLOv11 yang telah di training dan disesuaikan dengan YOLOv11. Dataset terdiri dari 1324 gambar, dengan pembagian 927 gambar untuk dataset *training* dan 265 gambar untuk dataset *testing*, dan 132 gambar untuk dataset *validating*. Kelas objek yang dikenali dalam sistem ini adalah kursi dan orang.

Hasil dari *training* data dapat dilihat seperti pada tabel 2. pada kelas kursi menunjukkan nilai AP 98% dengan jumlah *True Positive* sebesar 4967 dimana jauh lebih besar dibandingkan dengan jumlah *True False* sebesar 0. Pada kelas orang menunjukkan nilai AP 95% dengan jumlah *True Positive* sebesar 1921 dimana jauh lebih besar dibandingkan dengan jumlah *True False* sebesar 0. Waktu yang digunakan untuk pemrosesan model sekitar 20 detik. Dengan nilai *recall* model sebesar 99%, *precision* model sebesar 100%, *f1_score* model sebesar 92%, dan mAP@5 model sebesar 95,7% yang berarti model dapat mengenali objek dengan sangat akurat.

Tabel 1. Performa Model YOLOv11

No	Model	YOLOv11
1	Kursi	AP 98%
		TP 4967
		FP 0
2	Orang	AP 95%

		TP	1921
		FP	0
3	Waktu Pemrosesan	20 detik	
4	<i>Recall</i>	99%	
5	<i>Precision</i>	100%	
6	<i>F1_Score</i>	92%	
7	<i>mAP@0.5</i>	95,7%	

Pengujian Model Hasil Deteksi








Pada tahap ini pengujian menggunakan weights yaitu best.pt untuk mengetahui keakuraatan model dari kelas yang telah dilatih. Program kemudian dijalankan dengan memasukkan beberapa video rekaman CCTV dari JARAK COFFEE & EATERY pada waktu dan hari yang berbeda-beda. Variasi waktu ini penting untuk menguji keandalan (*robustness*) model dalam berbagai kondisi cahaya dan kepadatan pengunjung. Hasil dari deteksi objek menggunakan weight YOLOv11 dapat dilihat seperti pada tabel 2.

Tabel 2. Rata-rata Nilai Confidane

Pengujian ke-	Tanggal	Waktu	Kursi	Orang	Hasil Deteksi
1	10/29/25	Siang	87%	76%	21/19
2	10/25/25	Malam	93%	78%	16/13
3	11/01/25	Siang	92%	70%	23/22
		Malam	87%	79%	13/6
4	11/02/25	Siang	62%	80%	21/17
		Malam	90%	79%	21/19
5	11/03/25	Siang	80%	82%	22/21
		Malam	96%	92%	16/12
6	11/04/25	Siang	70%	83%	21/19
		Malam	80%	79%	18/12

Dapat dilihat pada tabel 3. saat menjalankan deteksi dan penghitungan, hasil dari input video menghasilkan nilai confidence yang berbeda-beda pada setiap frame. Hal ini disebabkan oleh objek yang berpindah tempat, sebagian besar objek tertutup oleh objek lain, objek jauh dari kamera, objek berukuran kecil, kursi memiliki bentuk yang berbeda-beda, pencahayaan kurang, maupun warna objek mirip dengan objek lain.

Tabel 3. Hasil Deteksi CCTV

Pengujian ke-	Hari	Tanggal	Waktu	Hasil Deteksi
1	Rabu	10/29/25	10:31	
2	Sabtu	10/25/25	18:29	
3	Sabtu	11/01/25	11:30	
4	Sabtu	11/01/25	20:14	
5	Minggu	11/02/25	12:52	
6	Minggu	11/02/25	19:11	
7	Senin	11/03/25	16:02	

SIMPULAN

Sistem deteksi ketersediaan kursi menggunakan YOLOv11 telah berhasil dibuat. Sistem ini bisa mengenali objek "kursi" dan "orang," serta melihat

bagaimana keduanya berhubungan dengan menggunakan perhitungan *Intersection over Union* (IoU). Model yang telah dilatih menunjukkan kinerja yang sangat baik, dengan nilai *mean Average Precision* (mAP@0.5) sebesar 95,7% baik saat siang maupun malam, dan didukung oleh nilai *precision* dan *recall* yang cukup baik untuk digunakan di JARAK COFFEE & EATERY. Secara umum, model ini memiliki kemampuan deteksi yang sangat baik dengan rata-rata *confidence* untuk kursi antara 62-96% dan untuk orang 70-92%. Hasil terbaik tercatat pada input tanggal 11-03-2025 malam dengan *confidence* 96% untuk kursi dan 92% untuk orang. Sistem ini juga dapat menghitung ketersediaan kursi secara langsung dengan menggunakan logika IoU, di mana kursi dianggap "terisi" ketika nilai IoU lebih dari 0,1.

DAFTAR PUSTAKA

- Siddhi Pravin Lipare and Aryan Jain. 2022. Empty Seat Detection in Campus Restaurants for IISER - B Community Using Computer Vision. *3rd year, Data Science and Engineering Major, IISER Bhopal*.
- Abdillah, Fajrian Fatmahan. 2023. Implementasi You Only Look Once (YOLO) V3 untuk deteksi objek pada aktivitas manusia di ruangan kelas. Sarjana thesis, UIN Sunan Gunung Djati Bandung.
- Bima Putra Gusti Pamungkas, Budi Nugroho, Fetty Anggraeny. 2021. DETEKSI DAN MENGHITUNG MANUSIA MENGGUNAKAN YOLO-CNN. *Informatika, Fakultas Ilmu Komputer, 2(1), 67-75. Jurnal Informatika dan Sistem Informasi (JIFoSI)*.
- Saptadi, Kristiawan, Rahayu. 2025. *Deep Learning: Teori, Algoritma, dan Aplikasi*. Sada Kurnia Pustaka.
- Naf'an, Emil and Islami, Fajrul and Gushelmi, Gushelmi. 2022. *DASAR-DASAR DEEP LEARNING dan Contoh Aplikasinya*. In: *DASAR-DASAR DEEP*

- Alnando Rizki Kurniawan, Zuly Burdiarso, Eka Ardhianto. 2025. IMPEMENTASI ALGORITHMMA YOLO UNTUK DETEKSI JUMLAH PENGUNJUNG DIDALAM GEDUNG. *Jurnal Mahasiswa Teknik Informatika*, 9(2), 2944-2948.
- Budi Hartono. 2021. *Cara Mudah dan Cepat Belajar Pengembangan Sistem Informasi*. Semarang: Yayasan Prima Agus Teknik.
- Bayu Ismail, Eri Zuliarso. 2023. Implementasi Metode YOLO pada Deteksi Pakaian Keselamatan yang Lengkap di Proyek Kontruksi. *Journal Of Multidisciplinary Research and Development*, 6(1), 56-63.
<https://doi.org/10.38035/rtrj.v6i1>
- Mohamad Sepriyanto Utomo. 2023. DETEKSI JUMLAH MANUSIA MENGGUNAKAN METODE YOLOv4 DALAM SUATU GEDUNG (STUDI KASUS: LAB ELEKTRO UNISSULA). Universitas Islam Sultan Agung Semarang.
- Rancang Bangun Sistem Deteksi Manusia dengan YOLO pada video CCTV. 2024. *Prosding SEMNAS INOTEK: Seminar Nasional Inovasi Teknologi*, 8(2), 1149-1154.
- Fahri Muhamad Zulfikarnaen. 2024. SISTEM DETEKSI OBJEK MANUSIA MENGGUNAKAN ALGORITMA YOLOV8 BERBASIS KAMERA DEPTH SENSOR (STUDI KASUS: CV. ATERI GLOBAL TEKNOLOGI). Universitas Sangga Buana YPKP
- Aldhiyatika Amwin. 2024. DETEKSI DAN KLASIFIKASI KENDARAAN BERBASIS ALGORITMA YOU ONLY LOOK ONCE (YOLO).