**LAPORAN PROJEK UAS**

**PEMBELAJARAN MESIN A**

**“DETEKSI KUALITAS BUAH PIR”**

Dosen Pengampu:

**Drs. Hari Suparwito, S.J., M.App.IT**



Oleh:

**Gerardo Mayella Ardianta/ 235314003**

**Valentino Banyu Ampar Paraya/ 235314006**

PROGRAM STUDI INFORMATIKA

FAKULTAS SAINS DAN TEKNOLOGI

UNIVERSITAS SANATA DHARMA

2025

1. **Dokumen Teori**

1.1 Latar Belakang

Buah pir merupakan salah satu buah yang banyak diminati masyarakat karena rasanya yang manis dan teksturnya yang renyah. Di wilayah Yogyakarta, permintaan akan buah impor seperti pir cukup tinggi, terutama di pusat-pusat perbelanjaan, pasar modern, dan toko buah premium. Namun, seringkali kualitas buah pir yang dijual tidak seragam, sehingga konsumen menghadapi risiko membeli buah yang rusak, busuk, atau tidak layak konsumsi.

Deteksi kualitas buah pir secara manual oleh manusia memiliki berbagai keterbatasan, seperti subjektivitas penilaian, kelelahan, serta ketidakkonsistenan dalam proses seleksi. Hal ini menjadi tantangan terutama bagi pelaku UMKM, distributor buah, atau toko buah modern di Yogyakarta yang ingin menjaga reputasi dan kepuasan pelanggan.

Dengan kemajuan teknologi *computer vision* dan *deep learning*, khususnya melalui algoritma seperti YOLO (You Only Look Once), proses deteksi kualitas buah dapat dilakukan secara otomatis, cepat, dan konsisten. Sistem ini dapat mengenali buah pir berdasarkan kondisi permukaan, warna, dan tekstur, lalu mengklasifikasikan apakah buah tersebut dalam kondisi baik atau rusak.

Proyek ini bertujuan untuk membangun sistem deteksi kualitas buah pir yang sederhana menggunakan algoritma YOLO sebagai solusi secara cepat dalam penyaringan dan pengawasan kualitas buah. Namun untuk proyek ini hanya menggunakan

1.2 Desain dan Arsitektur YOLO – Literature Review

1. Apa itu YOLO dalam Konteks CNN Deep Learning

YOLO (You Only Look Once) adalah algoritma deteksi objek berbasis Convolutional Neural Network (CNN) yang mampu melakukan deteksi dalam satu langkah (single shot). Tidak seperti metode deteksi dua tahap seperti R-CNN, YOLO langsung memprediksi bounding box dan kelas objek dalam satu proses inferensi, menjadikannya sangat cepat dan efisien.

1. Arsitektur Dasar YOLO

YOLO membagi gambar input menjadi grid S X S. Setiap grid bertanggung jawab untuk mendeteksi objek yang pusatnya berada dalam grid tersebut. Setiap grid memprediksi beberapa bounding box, confidence score, serta kelas objek. Arsitektur YOLO terdiri dari backbone untuk ekstraksi fitur, kemudian diikuti oleh neck dan head untuk prediksi akhir.

1. Keuntungan dan Kelemahan YOLO

Keuntungan:

* **Sangat cepat** (pada GPU modern bisa > 150 FPS). Cocok untuk real-time (drone, CCTV, kendaraan otonom).
* **End-to-end sederhana**: satu jaringan, satu fungsi hilang; gampang dilatih & di-deploy pada edge-device.
* **Efisien memori**: lebih sedikit operasi dibanding model proposal, sehingga cocok untuk device terbatas.

Kelemahan:

* **Akurasi objek kecil lebih rendah** dibanding dua-tahap (Faster R-CNN, Mask R-CNN), karena resolusi fitur terbatas.
* **Tergantung ukuran anchor & grid**; sub-optimal jika distribusi ukuran objek baru sangat berbeda dari data latih.
* **Kesalahan lokalisasi** relatif lebih besar (bounding box bisa kurang presisi) terutama pada tumpang-tindih objek padat.

1. Anchor Boxes dan Grid Cells

YOLO menggunakan konsep grid cells untuk membagi gambar input. Setiap grid akan bertanggung jawab mendeteksi objek di dalamnya. Sementara itu, anchor boxes adalah bentuk awal (prior) dari bounding boxes yang digunakan untuk memprediksi bentuk objek sebenarnya. Anchor boxes memungkinkan model mengenali objek dengan bentuk dan rasio aspek berbeda pada satu grid cell.

**2. Implementasi**

2.1. Implementasi YOLO

Dalam proyek ini, kami menggunakan algoritma YOLO (You Only Look Once) untuk melakukan deteksi kualitas buah pir berdasarkan citra gambar. Implementasi dilakukan melalui tiga tahap utama, yaitu preprocessing, training, dan inference. Kami menggunakan Roboflow untuk anotasi dan manajemen dataset.

1. Preprocessing

Langkah awal dalam proses implementasi adalah menyiapkan dataset gambar buah pir. Gambar-gambar tersebut terdiri dari dua kategori: Good Pear dan Bad Pear.

Kami melakukan labeling (anotasi bounding box) menggunakan platform Roboflow, yang menyediakan antarmuka grafis untuk menandai objek secara manual dan otomatis. Setelah semua gambar dilabeli, kami menggunakan fitur Export Dataset dari Roboflow untuk mengekspor dataset dalam format YOLO v8.

Langkah-langkah preprocessing yang dilakukan:

* Semua gambar diubah ukurannya menjadi 640x640 piksel secara otomatis oleh Roboflow
* Melakukan augmentation data:

Horizon Flip

Rotation

Noise injection

* Dataset lalu dibagi untuk kebutuhan training (92%), validation (5%) dan testing (3%)

1. Training Model

Kami menggunakan model YOLOv8n (versi ringan dari YOLOv8 oleh Ultralytics), karena ukurannya kecil dan cepat dilatih, cocok untuk eksperimen awal dan dataset yang sedikit.

1. Evaluasi

Setelah proses pelatihan selesai, kami melakukan evaluasi terhadap model menggunakan fungsi model.val() dari pustaka Ultralytics YOLO. Evaluasi ini dilakukan pada dataset validasi, yang sebelumnya telah dipisahkan oleh Roboflow secara otomatis selama proses ekspor dataset.

Tujuan evaluasi ini adalah untuk mengukur seberapa baik performa model dalam mengenali dan mengklasifikasikan buah pir, serta untuk mengetahui kekuatan dan kelemahan model berdasarkan metrik yang diperoleh.

2.2. Dataset

Siapkan dataset yang sesuai dengan kasus penggunaan yang dipilih. Jika dataset tidak tersedia secara publik, deskripsikan proses pengumpulan data. Lakukan eksplorasi dan visualisasi dataset. Dataset yang unik dan baru akan sangat dihargai

Kami ingin membuat program kami dapat mendeteksi kualitas buah pir. Kami menggunakan gambar buah pir yang kami foto sendiri :

link dataset google drive:

<https://drive.google.com/file/d/1U9-dM8Y_7clyv4-i9qDyz5XxwXSYRrrW/view?usp=drive_link>

link dataset roboflow:  
<https://app.roboflow.com/test-gswap>



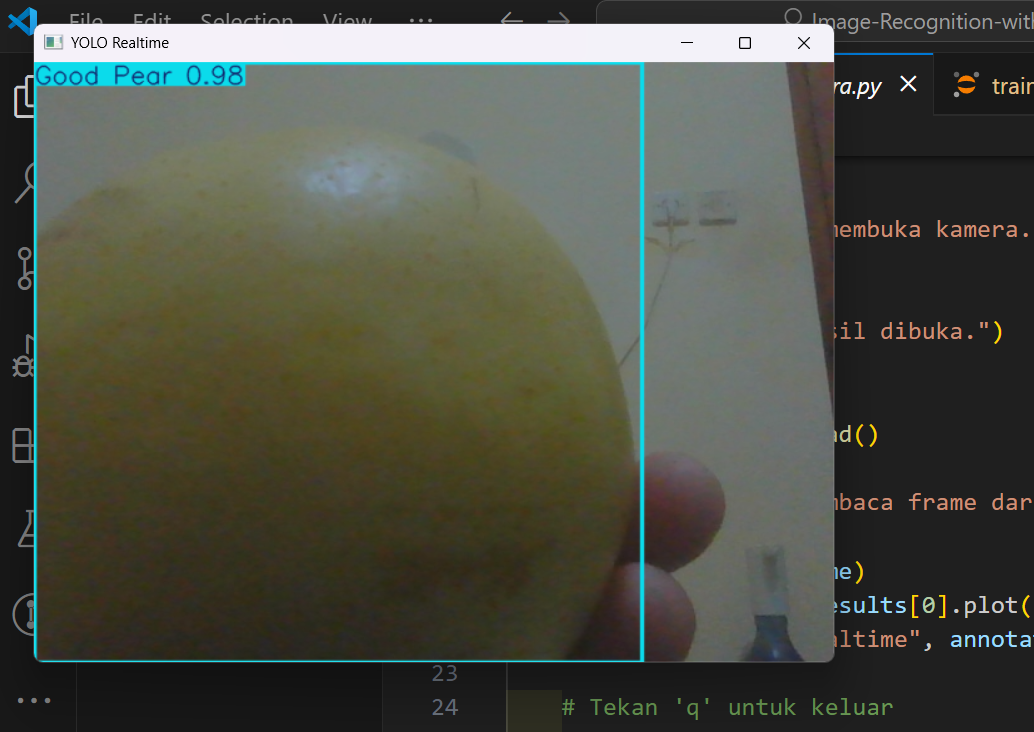
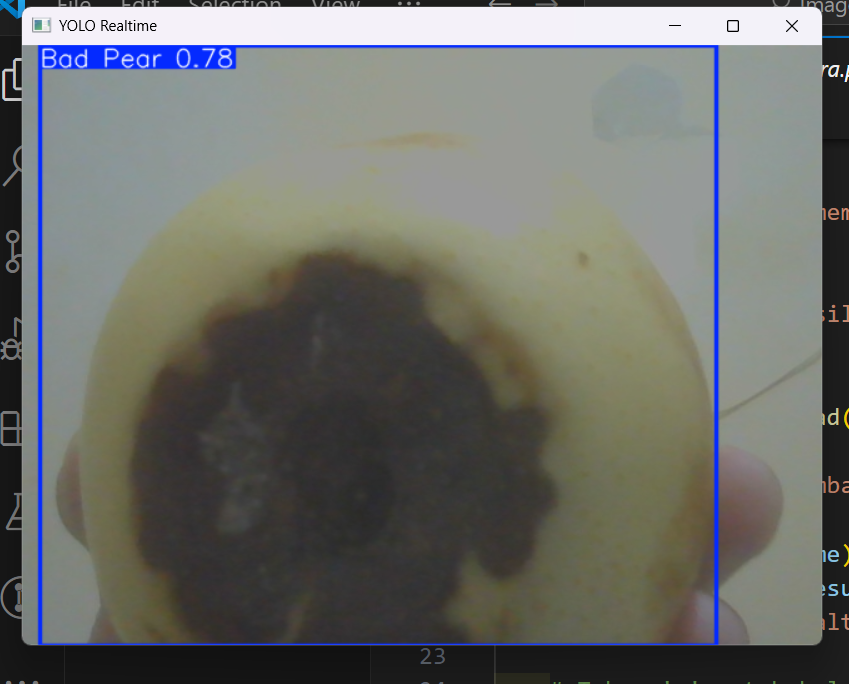
gambar pir seperti ini untuk mendeteksi pir bagus (good pear)



gambar pir seperti ini untuk mendeteksi pir busuk (bad pear)

Kami mengambil gambar dengan total 708, kemudian kita augmentasi data untuk memberikan data yang lebih banyak menjadi 1.838 data. Jadi total data yang kami pakai untuk mendeteksi buah pir adalah 1.838.

2.3. Inference dan Visualisasi



Berikut adalah inference dan visualisasinya. Ada 2 gambar, yang dimana program dapat mendeteksi buah pir busuk dan bagus. Dengan confidence dari bad pear 0.78 (78%) dan confidence good pear 0.98 (98%). Dalam pendeteksian, terkadang kita harus menempatkan posisi yang tempat agar buah dapat dideteksi dengan baik. Seperti mendekatkan pada kamera ataupun memberikan cahaya.

2.4. Evaluasi dan Analisis

Setelah proses pelatihan selesai, kami melakukan evaluasi terhadap model menggunakan dataset validasi yang telah disiapkan melalui Roboflow. Dataset validasi terdiri dari 90 gambar dengan label "Good Pear" (Pir Bagus) tanpa file yang rusak atau kosong.

Data Evaluasi dapat dilihat dengan perintah:



Ringkasannnya dapat dilihat sebagai berikut:  
 Jumlah Layer: 72

Parameter: 3.006.038

Dataset Validasi: 90 gambar

Metrik Evaluasi:

| Metrik | Nilai |
| --- | --- |
| Precision | 0.999 |
| Recall | 1.000 |
| mAP@0.5 | 0.995 |
| mAP@0.5:0.95 | 0.968 |

Dapat dikatakan bahwa deteksi sangat akurat dan lengkap dengan precision dan recall mendekati sempurna.

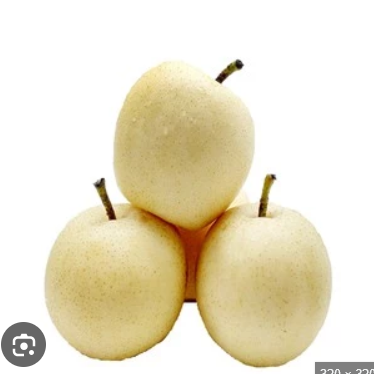
Performa Kecepatan:

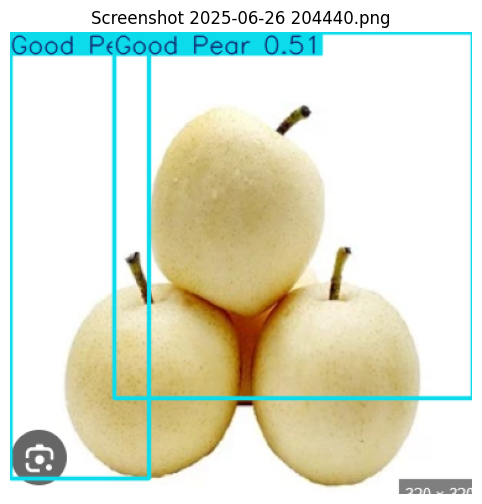
Preprocess: 8.3 ms/gambar

Inference: 4.9 ms/gambar

postprocess: 5.6 ms/gambar

Hasil deteksi dengan menggunakan sampel gambar buah pir dari google:



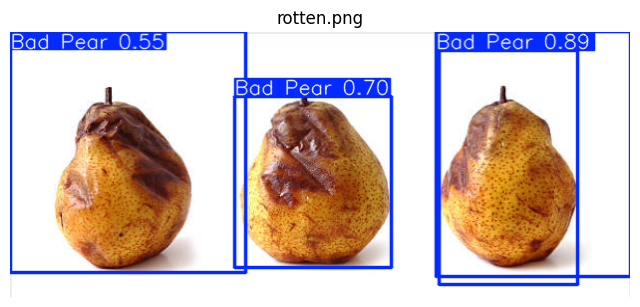


Pada Gambar Pir Bagus diatas Model berhasil mendeteksi 2 buah pir dengan label “Good Pear” dengan Confidence score: 0.51, Bagi kami hasil ini kurang baik karena seharusnya model mendeteksi 3 pir bagus bukan hanya 2 saja.

Analisis:

* Model cenderung under-detecting (hanya mendeteksi sebagian)
* Confidence rendah menandakan bahwa model belum terlalu yakin terhadap deteksi tersebut
* Kemungkinan disebabkan oleh variasi posisi buah pir, pencahayaan terlalu terang, atau kemiripan bentuk antar objek yang membuat model bingung





Pada gambar pir busuk, model berhasil mendeteksi ketiga buah pir dengan label “Bad Pear”, masing-masing dengan confidence cukup tinggi kecuali buah pir paling kiri, yaitu: 0.55, 0.70, 0.89. Bounding box terlihat berada di sekitar objek target.

Analisis:

* Model sangat kuat mendeteksi buah busuk bahkan ketika permukaan buah memiliki kerusakan besar
* Warna dan tekstur busuk yang kontras dengan latar putih kemungkinan memudahkan model mengenali kelasnya
* Confidence yang tinggi menunjukkan kepercayaan tinggi dari model terhadap klasifikasi

Kekuatan Model:

1. Deteksi terhadap buah pir rusak sangat akurat, dengan bounding box tepat dan confidence tinggi
2. Model tetap mampu mengenali pir yang bentuknya bengkok atau teksturnya rusak, menunjukkan robustness terhadap deformasi objek
3. Model sudah mampu membedakan dua kelas (*Good* dan *Bad Pear*) tanpa kebingungan klasifikasi silang.

Kelemahan Model:

1. Hanya satu dari tiga pir baik yang berhasil terdeteksi, dan dengan confidence rendah (0.51). Ini mengindikasikan masalah recall pada kelas “Good Pear”
2. Dataset yang kami ambil untuk “Good Pear” memiliki banyak noise dan pengambilan gambar yang kurang bagus sehingga model tidak belajar dengan baik
3. Gambar pir baik yang terang dan objek yang saling tumpang tindih bisa menurunkan akurasi karena model sulit membedakan batas antar buah