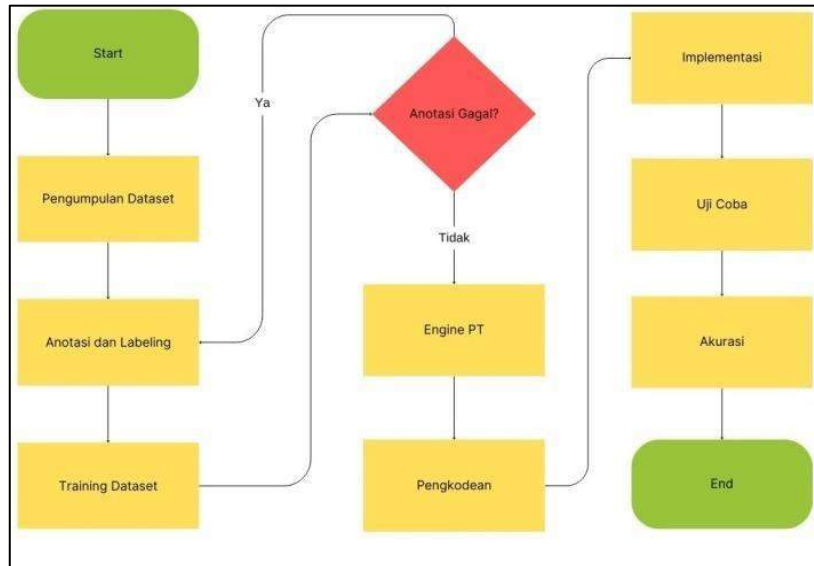


2. METODE PENELITIAN

2.1 Alur penelitian

Penelitian ini bertujuan untuk mengevaluasi presisi dan akurasi model YOLOv8 dengan 100 *epoch* dan 8 *batch*. Sehingga kita akan mendapatkan gambaran yang lebih jelas performa pada model YOLOv8 yang dilatih dengan parameter tersebut. Adapun alur penelitian diilustrasikan pada gambar 1.



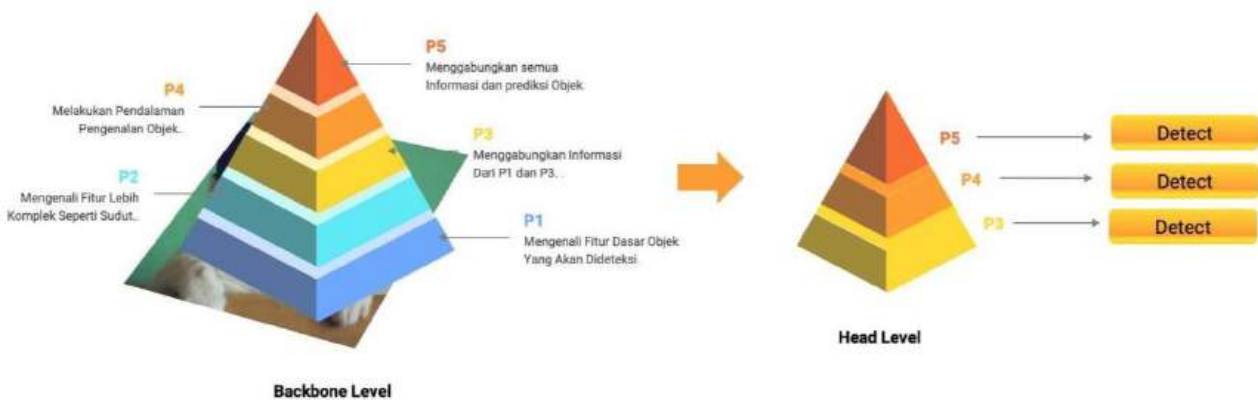
Gambar 1. Metode Pengembangan Penelitian

Pada gambar 1 dapat dijelaskan sebagai berikut:

1. Tahap pertama dalam penelitian ini adalah pengumpulan dataset jenis buah yang segar dan tidak segar. Dalam penelitian ini, kami hanya mengambil sampel Apel, Pisang, Mangga, Melon, Jeruk, Persik dan Pir.
2. Pada tahap kedua kami melakukan anotasi dan labelisasi gambar untuk memberikan objek koordinat terhadap jenis-jenis kesegaran buah dengan kelas segar, tidak segar dan semi segar.
3. Pada tahap ketiga kami melakukan pelatihan *dataset* dengan menggunakan *library ultralytics* untuk mendapatkan bobot *convolutional network* dalam format PT.
4. Selanjutnya di tahap kelima Bobot PT yang telah didapat akan dibaca dan kode python yang telah disusun, dan digunakan untuk melakukan deteksi kesegaran buah.
5. Pada tahap keenam, hasil dari deteksi akan kami evaluasi dengan *F1-Score* untuk melihat akurasi deteksi yang dihasilkan.

2.2 Algoritma YOLO

YOLOv8 adalah iterasi terbaru dalam seri keluarga detektor objek YOLO yang memiliki tingkat akurasi dan kecepatan diatas YOLO seri sebelumnya, sehingga menjadi pilihan yang paling baik untuk tugas deteksi kesegaran buah. Adapun ilustrasi arsitektur YOLOv8 dijabarkan pada gambar 1.

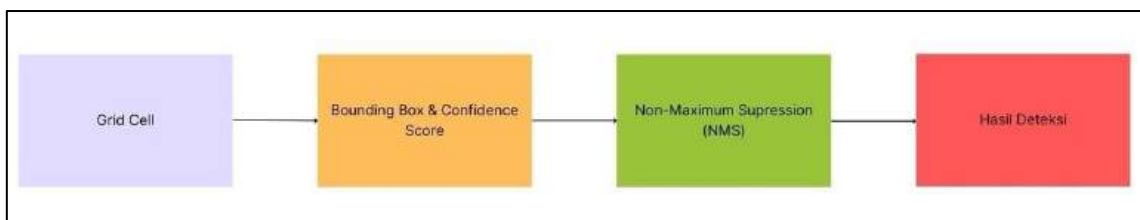


Gambar 2. Arsitektur YOLOv8

Pada gambar 2 dapat dijelaskan bahwa YOLOv8 memiliki 2 arsitektur jaringan yaitu *backbone* dan *head*. *Backbone* adalah bagian dari jaringan saraf konvolusi (CNN) yang bertanggung jawab untuk mengekstraksi fitur-fitur dari data masukan seperti gambar, video atau stream dari kamera. Pada Lapisan *backbone*, terdapat lima tingkatan yang merupakan tahapan metode pendeteksian.

1. Tingkatan P1 digunakan untuk mengenali fitur-fitur dasar dari objek, seperti garis, bentuk, warna dan elemen lainnya pada objek.
2. Tingkatan P2 digunakan untuk mengenali fitur yang lebih kompleks dari objek seperti sudut gambar.
3. Tingkatan P3 digunakan untuk menggabungkan informasi yang telah dikumpulkan oleh P1 dan P2 untuk diproses lebih lanjut.
4. Tingkatan P4 digunakan untuk melakukan pendalaman pengenalan objek yang sudah dikumpulkan level P1 dan P2.
5. Tingkatan P5 digunakan untuk menggabungkan semua informasi dan prediksi objek. Yang selanjutnya digunakan sebagai landasan untuk deteksi.

Selanjutnya pada tingkatan *head*, semua informasi yang telah diproses oleh *backbone* akan diproses untuk melakukan prediksi objek dengan membuat *bounding box*, *confidence score* dan mengeliminasi objek-objek lain dengan skor dibawah *threshold* yang telah diatur. Secara umum, YOLOv8 bekerja dengan metode yang ditunjukkan pada gambar 3, dimana ada 4 alir yang akan dilalui.



Gambar 3. Metode YOLOv8

1. Pada tahap awal, YOLO akan membagi gambar menjadi sel-sel (*grid cells*), dengan setiap sel bertugas untuk memprediksi objek yang terdapat di dalamnya. [16].
2. Selanjutnya, setiap sel akan membuat prediksi terhadap beberapa bounding box beserta skor kepercayaannya (*confidence score*) [17]. *Bounding box* ini menggambarkan koordinat objek

yang dideteksi dalam gambar, dengan menghubungkan koordinat objek yang sebenarnya dengan koordinat yang terdapat dalam dataset [18]. Untuk membuat bounding box, maka akan digunakan rumus (1)

$$(x, y) = \left(\frac{1}{1+e^{-tx}}, \frac{1}{1+e^{-ty}} \right) \quad (1)$$

(x, y) merupakan koordinat pusat dari bounding box objek. Sedangkan (tx, ty) merupakan koordinat pusat dari bounding box objek yang dihitung menggunakan fungsi sigmoid sebagaimana yang dijelaskan dalam rumus (1).

Kemudian, untuk menentukan ukuran bounding box, akan digunakan rumus (2).

$$(w, h) = (pw \cdot e^{tw}, ph \cdot e^{th}) \quad (2)$$

dimana pw dan ph merepresentasikan dimensi anchor box yang dipilih, sedangkan tw dan th adalah output dari jaringan saraf untuk dimensi kotak pembatas objek.

Lalu untuk menghitung *confidence score* maka akan menggunakan rumus (3).

$$P(\text{Objek}) = \sigma(t_{obj}) \quad (3)$$

Dimana t_{obj} adalah *output* dari jaringan saraf yang menunjukkan probabilitas keberadaan objek.

3. Selanjutnya, pada langkah terakhir, YOLO menggunakan teknik *Non-Maximum Suppression* (NMS) untuk menghilangkan *bounding box* yang memiliki skor kepercayaan rendah, sehingga hanya *bounding box* dengan skor kepercayaan tinggi yang dipertahankan. [19].
4. Terakhir, model menghasilkan output berupa *bounding box* yang mencakup prediksi objek beserta skornya. Setiap *bounding box* juga mencantumkan label kelas objek dan tingkat kepercayaan model terhadap prediksi tersebut. [20].

2.3 Parameter Dataset dan Pelatihan

Keterbatasan *hardware* menjadi perhatian dalam pemilihan parameter dalam melatih model YOLOv8 untuk mendeteksi kesegaran buah. Pada penelitian ini kami menggunakan *google colab pro* dengan GPU Nvidia L2 dengan 24Gb Vram untuk melatih 5215 gambar dataset yang bersumber dari Jawaharlal Nehru University [21]. *Dataset* tersebut dibagi kedalam 83% untuk training data, 11% untuk validasi data, dan 6% untuk testing data dengan rincian seperti yang dijabarkan pada tabel 1.

Tabel 1. Penjabaran Dataset

No	Jenis Buah	Class	Jumlah
1	Apel	<i>Apple</i>	221
		<i>Apple Rotten</i>	183
		<i>Apple Semifresh</i>	63
		<i>Apple Semirotten</i>	158
2	Pisang	<i>Banana</i>	57
		<i>Banana Rotten</i>	173
		<i>Banana Semifresh</i>	533

No	Jenis Buah	Class	Jumlah
		<i>Banana Semirotten</i>	83
3	Mangga	<i>Mango</i>	37
		<i>Mango Rotten</i>	127
		<i>Mango Semifresh</i>	64
		<i>Mango Semirotten</i>	112
4	Melon	<i>Melon</i>	47
		<i>Melon Rotten</i>	12
		<i>Melon Semifresh</i>	21
		<i>Melon Semirotten</i>	18
5	Orange	<i>Orange</i>	81
		<i>Orange Rotten</i>	103
		<i>Orange Semifresh</i>	62
		<i>Orange Semirotten</i>	47
6	Pir	<i>Pear</i>	31
		<i>Pear Rotten</i>	95
		<i>Pear Semifresh</i>	46
		<i>Pear Semirotten</i>	21
7	<i>Peach</i>	<i>Peach</i>	83
		<i>Peach Rotten</i>	72
		<i>Peach Semifresh</i>	152
		<i>Peach Semirotten</i>	47
<i>Preprocessing</i>		<i>Orient: Applied</i> <i>Resize: Stretch to 640x640</i>	
<i>Augmentations</i>		<i>Outputs per training example: 3</i> <i>Flip: Horizontal, Vertical</i>	

Pada tabel 1. *Orient* merujuk pada fitur untuk menyesuaikan orientasi gambar dalam dataset, sehingga semua gambar dalam dataset memiliki orientasi yang konsisten dan sesuai dengan kebutuhan model yang akan dilatih. Selain itu ukuran gambar dataset juga dibuat dengan ukuran 640x640 px. Selanjutnya pada augmentasi, kami menambahkan variasi gambar dengan merotasinya secara horizontal dan vertikal, masing-masing gambar akan di augmentasi 3x. Selanjutnya kami menggunakan 100 *epoch* pelatihan dengan 8 *batch*. *Epoch* mengacu pada jumlah siklus melalui seluruh dataset pelatihan yang akan dilakukan selama proses pelatihan [22]. Sedangkan *batch* mengacu pada jumlah sampel data yang dimasukkan ke dalam model pada satu waktu untuk setiap iterasi pelatihan [23].

2.4 Presisi, Recall dan F1-Score

Untuk menghitung presisi, *recall* dan *F1-score* pada model YOLOv8 yang dikembangkan untuk mendeteksi kesegaran buah. Maka kami menggunakan rumus (4) untuk menghitung presisi, rumus (5) untuk menghitung *recall*, dan rumus (6) hasil *F1-score*.

1. Rumus Presisi (*precision*) untuk mengukur seberapa banyak dari prediksi positif yang benar [24] adalah sebagai berikut:

$$presisi = \frac{true\ positive}{true\ positive + false\ positive} \quad (4)$$

-
2. Rumus *recall* untuk mengukur seberapa banyak dari keseluruhan data positif yang berhasil ditemukan oleh model [25] adalah sebagai berikut:

$$recall = \frac{true\ positive}{true\ positive + false\ negative} \quad (5)$$

3. Terakhir untuk mendapatkan *f1-score* kami menggunakan formula (6) sebagai berikut:

$$F1\ Score = 2 \frac{presisi * recall}{presisi + recall} \quad (6)$$

