



SKRIPSI

SISTEM ESTIMASI MASSA TANDAN BUAH SEGAR KELAPA SAWIT MENGGUNAKAN CITRA *UNMANNED AERIAL VEHICLE* DENGAN MODEL *RANDOM FOREST REGRESSION*

Disusun dan diajukan oleh:

MUHAMMAD IJLAL NURHADI
D121 19 1068



PROGRAM STUDI SARJANA TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS HASANUDDIN
GOWA
2024



LEMBAR PENGESAHAN SKRIPSI

SISTEM ESTIMASI MASSA TANDAN BUAH SEGAR KELAPA SAWIT MENGGUNAKAN CITRA *UNMANNED AERIAL VEHICLE* DENGAN MODEL *RANDOM FOREST REGRESSION*

Disusun dan diajukan oleh

Muhammad Ijfal Nurhadi
D121 19 1068

Telah dipertahankan di hadapan Panitia Ujian yang dibentuk dalam rangka Penyelesaian Studi Program Sarjana Program Studi Teknik Informatika Fakultas Teknik Universitas Hasanuddin Pada tanggal 31 JULI 2024 dan dinyatakan telah memenuhi syarat kelulusan

Menyetujui,

Pembimbing Utama,

Prof. Dr. Ir. Indrabayu, S.T., M.T., M.Bus.Sys., IPM, ASEAN, Eng.
NIP 197507162002121004

Pembimbing Pendamping,

Elly Warni, S.T., M.T.
NIP 198202162008122001

Ketua Program Studi,



Prof. Dr. Ir. Indrabayu, S.T., M.T., M.Bus.Sys., IPM, ASEAN, Eng.
NIP 197507162002121004



PERNYATAAN KEASLIAN

Yang bertanda tangan dibawah ini;

Nama : Muhammad Ijlal Nurhadi
 NIM : D121191068
 Program Studi : Teknik Informatika
 Jenjang : S1

Menyatakan dengan ini bahwa karya tulisan saya berjudul

**SISTEM ESTIMASI MASSA TANDAN BUAH SEGAR KELAPA SAWIT
 MENGGUNAKAN CITRA *UNMANNED AERIAL VEHICLE* DENGAN
 MODEL *RANDOM FOREST REGRESSION***

Adalah karya tulisan saya sendiri dan bukan merupakan pengambilan alihan tulisan orang lain dan bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri.

Semua informasi yang ditulis dalam skripsi yang berasal dari penulis lain telah diberi penghargaan, yakni dengan mengutip sumber dan tahun penerbitannya. Oleh karena itu semua tulisan dalam skripsi ini sepenuhnya menjadi tanggung jawab penulis. Apabila ada pihak manapun yang merasa ada kesamaan judul dan atau hasil temuan dalam skripsi ini, maka penulis siap untuk diklarifikasi dan mempertanggungjawabkan segala resiko.

Segala data dan informasi yang diperoleh selama proses pembuatan skripsi, yang akan dipublikasi oleh Penulis di masa depan harus mendapat persetujuan dari Dosen Pembimbing.

Apabila dikemudian hari terbukti atau dapat dibuktikan bahwa sebagian atau keseluruhan isi skripsi ini hasil karya orang lain, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Gowa, 31 Juli 2024

Yang Menyatakan

Muhammad Ijlal Nurhadi



ABSTRAK

MUHAMMAD IJLAL NURHADI. *Sistem Estimasi Massa Tandan Buah Tandan Buah Segar Kelapa Sawit Menggunakan Citra Unmanned Aerial Vehicle dengan Model Random Forest Regression* (dibimbing oleh Indrabayu dan Elly Warni)

Salah satu aspek penting dalam pengelolaan perkebunan kelapa sawit adalah perencanaan pra-panen untuk mengoptimalkan hasil panen. Perencanaan pra-panen melibatkan estimasi hasil panen yang dilakukan dengan menimbang Tandan Buah Segar (TBS) dari beberapa pohon secara acak untuk mengukur massa rata-rata. Metode ini memiliki kelemahan pada ketidakakuratan *sampling* dan proses penimbangan manual yang memerlukan waktu dan tenaga yang signifikan. Oleh karena itu, diperlukan pendekatan yang lebih efektif dengan menggunakan visi komputer dan *machine learning* untuk memudahkan proses estimasi hasil panen.

Tujuan dari penelitian ini yaitu merancang sistem estimasi massa TBS kelapa sawit menggunakan citra dari *Unmanned Aerial Vehicle* (UAV). Citra dari UAV akan disegmentasi menggunakan *YOLOv8-Segmentation* untuk mengekstraksi fitur geometris TBS yang kemudian digunakan untuk melatih model *Random Forest Regression* guna mengestimasi massa TBS kelapa sawit.

Sebanyak 200 sampel TBS kelapa sawit yang sudah diketahui massa aktualnya digunakan sebagai dataset dengan pembagian: 100 TBS tertutup pelepah dan 100 TBS tidak tertutup pelepah. Dari dataset ini, 3640 citra diekstraksi untuk melatih model *YOLOv8-Segmentation* dengan terlebih dahulu diaugmentasi dengan penyesuaian saturasi, kecerahan dan eksposur sehingga menghasilkan 8736 citra. Model *YOLOv8-Segmentation* akan mengekstraksi fitur geometri TBS kelapa sawit yakni sumbu mayor, sumbu minor, luas, keliling dan eksentrisitas untuk melatih model *Random Forest Regression*. Metrik evaluasi *Root Mean Square Error* (RMSE) dan *Mean Absolute Error* (MAE) digunakan untuk mengukur akurasi estimasi massa TBS kelapa sawit.

Hasil penelitian menunjukkan bahwa model *Random Forest Regression* yang menggunakan dua fitur-sumbu minor dan eksentrisitas yang diekstrak menggunakan *YOLOv8Small-Segmentation* memiliki akurasi terbaik pada dataset TBS tertutup pelepah dengan RMSE sebesar 4,73 dan MAE sebesar 3,46. Sedangkan model *Random Forest Regression* yang menggunakan dua fitur-sumbu minor dan luas yang diekstrak menggunakan *YOLOv8Small-Segmentation* memiliki akurasi terbaik pada dataset TBS tidak tertutup pelepah dengan RMSE sebesar 3,89 dan MAE sebesar 2,92. Model dengan akurasi terbaik diimplementasikan dalam sistem estimasi massa TBS kelapa sawit, menawarkan pendekatan praktis untuk membantu perencanaan pra-panen yang lebih efektif.

Kata Kunci: estimasi massa TBS, *Unmanned Aerial Vehicle*, *YOLOv8-Segmentation*, *Random Forest Regression*



ABSTRACT

MUHAMMAD IJLAL NURHADI. *Fresh Fruit Bunch Mass Estimation System for Oil Palm Using Unmanned Aerial Vehicle Imagery with Random Forest Regression Model* (supervised by Indrabayu and Elly Warni)

Effective management of oil palm plantations necessitates accurate pre-harvest planning to optimize yield outcomes. Traditionally, pre-harvest planning involves estimating yields by manually weighing Fresh Fruit Bunches (FFBs) from a random sample of trees to determine the average mass. This method is hindered by sampling inaccuracies and the substantial time and labor required for manual weighing. Therefore, a more efficient approach utilizing computer vision and machine learning is essential to streamline the estimation process.

This study aims to develop a system for estimating the mass of oil palm FFBs using imagery captured by Unmanned Aerial Vehicles (UAVs). The UAV images are segmented using the YOLOv8-Segmentation model to extract geometric features of the FFBs, which are subsequently employed to train a Random Forest Regression model for mass estimation.

The dataset comprised 200 samples of oil palm FFBs with known actual masses, divided equally into 100 FFBs covered with fronds and 100 FFBs without fronds. From this dataset, 3640 images were extracted and augmented with adjustments in saturation, brightness, and exposure, resulting in a total of 8736 images. The YOLOv8-Segmentation model was utilized to extract geometric features of the FFBs, including the major axis, minor axis, area, perimeter, and eccentricity, which were then used to train the Random Forest Regression model. The evaluation metrics employed to assess the accuracy of the mass estimation were Root Mean Square Error (RMSE) and Mean Absolute Error (MAE).

The findings indicate that the Random Forest Regression model incorporating two features—minor axis and eccentricity—extracted using YOLOv8Small-Segmentation demonstrated the highest accuracy for the frond-covered FFB dataset, with an RMSE of 4.73 and an MAE of 3.46. Conversely, for the frond-uncovered FFB dataset, the model using the features minor axis and area, also extracted using YOLOv8Small-Segmentation, achieved the best accuracy with an RMSE of 3.89 and an MAE of 2.92. The model with the highest accuracy was subsequently implemented into a practical system for estimating the mass of oil palm FFBs, thereby offering a more effective approach to pre-harvest planning.

Keywords: FFB mass estimation, Unmanned Aerial Vehicle, YOLOv8-Segmentation, Random Forest Regression



DAFTAR ISI

LEMBAR PENGESAHAN SKRIPSI	i
PERNYATAAN KEASLIAN.....	ii
ABSTRAK.....	iii
ABSTRACT.....	iv
DAFTAR ISI.....	v
DAFTAR GAMBAR	vii
DAFTAR TABEL.....	ix
DAFTAR SINGKATAN DAN ARTI SIMBOL	x
DAFTAR LAMPIRAN.....	xi
KATA PENGANTAR	xii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah.....	3
1.3 Tujuan Penelitian	3
1.4 Manfaat Penelitian	3
1.5 Ruang Lingkup.....	4
BAB II TINJAUAN PUSTAKA.....	5
2.1 Kelapa Sawit	5
2.2 Citra Digital.....	7
2.3 <i>Drone</i> DJI Mavic Air	8
2.4 Visi Komputer.....	11
2.5 <i>Convolutional Neural Networks</i>	11
2.6 <i>Preprocessing</i>	13
2.7 Augmentasi Citra	14
2.8 <i>You Only Look Once</i>	17
2.9 <i>Confusion Matrix</i>	21
2.10 <i>Mean Average Precision</i>	22
2.11 Convex Hull	23
2.12 Ekstraksi Fitur	24
2.13 <i>Random Forest Regression</i>	28
2.14 Uji <i>Mean Decrease in Accuracy</i>	30
2.15 <i>Grid Search</i>	31
2.16 <i>Root Mean Square Error</i>	31
2.17 <i>Mean Absolute Error</i>	32
BAB III METODOLOGI PENELITIAN.....	34
3.1 Lokasi Penelitian.....	34
3.2 Benda Uji dan Alat.....	35
3.3 Tahapan Penelitian	36
3.4 Teknik Pengambilan Data.....	37
3.5 Perancangan dan Implementasi Sistem.....	39
BAB IV HASIL DAN PEMBAHASAN	66
4.1 Hasil Akuisisi Data	66
4.2 Hasil Evaluasi Model Segmentasi.....	67
4.3 Hasil Ekstraksi Fitur.....	71
4.4 Hasil Evaluasi Model Estimasi Massa	74



BAB V KESIMPULAN DAN SARAN.....	86
5.1 Kesimpulan	86
5.2 Saran.....	86
DAFTAR PUSTAKA	87



DAFTAR GAMBAR

Gambar 1 Perbedaan *Dura*, *Tenera*, dan *Pisifera* 6

Gambar 2 Jarak tanam kelapa sawit..... 6

Gambar 3 Representasi model warna RGB dan komponen individualnya..... 7

Gambar 4 *Drone* DJI Mavic Air 8

Gambar 5 Stereo kamera pada DJI Mavic Air 9

Gambar 6 Prinsip triangulasi menggunakan stereo kamera 10

Gambar 7 Estimasi jarak pada *drone* DJI Mavic Air..... 10

Gambar 8 Arsitektur jaringan CNN untuk klasifikasi gambar..... 12

Gambar 9 Perbedaan saturasi warna pada citra..... 14

Gambar 10 Perbedaan kecerahan pada gambar 15

Gambar 11 Perbedaan eksposur pada gambar 16

Gambar 12 Output prediksi sebelum dan sesudah (penerapan NMS) 18

Gambar 13 Arsitektur YOLOv8 19

Gambar 14 Segmentasi sampah bawah laut menggunakan *YOLOv8-Segmentation* 20

Gambar 15 Struktur dari *Confusion Matrix* 22

Gambar 16 Convex hull pada titik-titik acak 24

Gambar 17 Elips dengan sumbu mayor dan sumbu minor 25

Gambar 18 Arsitektur random forest untuk analisis klasifikasi dan regresi 29

Gambar 19 Implementasi Grid Search 31

Gambar 20 Lokasi akuisisi data 34

Gambar 21 Lokasi penelitian 34

Gambar 22 Tahapan penelitian 36

Gambar 23 Kertas penanda sampel TBS yang akan direkam 38

Gambar 24 Ilustrasi akuisisi dataset..... 38

Gambar 25 Timbangan duduk..... 39

Gambar 26 Alur perancangan sistem 39

Gambar 27 *Labeling* pada TBS kelapa sawit..... 41

Gambar 28 Resize gambar 41

Gambar 29 Augmentasi saturasi pada citra..... 42

Gambar 30 Augmentasi kecerahan pada citra..... 43

Gambar 31 Augmentasi eksposur pada citra..... 44

Gambar 32 Alur pengembangan model *YOLOv8-Segmentation* 45

Gambar 33 Konversi *mask* segmentasi ke biner 51

Gambar 34 Fitting Elips pada convex hull hasil segmentasi 51

Gambar 35 Matriks biner 10×11 52

Gambar 36 Dataset sebelum normalisasi 59

Gambar 37 Dataset setelah normalisasi 59

Gambar 38 Alur pengembangan model *Random Forest Regression* 60

Gambar 39 Cabang logika estimasi massa TBS 64

Gambar 40 Hasil akuisisi citra TBS kelapa sawit..... 66

Gambar 41 *Training YOLOv8-Segmentation* berdasarkan nilai mAP 67

Gambar 42 Perbandingan MDA pada fitur-fitur hasil ekstraksi 72

Gambar 43 Perbandingan performa *RMSE* pada sembilan skenario yang diuji ... 74

Gambar 44 Perbandingan performa *MAE* pada sembilan skenario yang diuji..... 75



Gambar 45 Perbandingan performa *RMSE* pada sembilan skenario yang diuji ... 79
Gambar 46 Perbandingan performa *MAE* pada sembilan skenario yang diuji..... 81



DAFTAR TABEL

Tabel 1 Perbandingan <i>weight YOLOv8-Segmentation</i>	20
Tabel 2 <i>Default hyperparameter YOLOv8-Segmentation</i>	46
Tabel 3 Contoh perhitungan mAP	49
Tabel 4 <i>Precision</i> dan <i>Recall</i> dari contoh perhitungan mAP	50
Tabel 5 <i>Hyperparameter</i> dicari dengan <i>Grid Search</i>	61
Tabel 6 Hasil evaluasi model segmentasi YOLOv8	67
Tabel 7 Performa segmentasi TBS kelapa sawit yang tertutup pelepah	68
Tabel 8 Performa segmentasi TBS kelapa sawit yang tidak tertutup pelepah	70
Tabel 9 Ekstraksi fitur geometri TBS kelapa sawit dengan YOLOv8Large-Segmentation	71
Tabel 10 Skenario pengembangan model estimasi massa	72
Tabel 11 Hasil <i>Grid Search Random Forest Regression</i>	73
Tabel 12 Estimasi massa TBS kelapa sawit pada data uji tertutup pelepah	76
Tabel 13 Estimasi massa TBS kelapa sawit pada data uji tidak tertutup pelepah	82



DAFTAR SINGKATAN DAN ARTI SIMBOL

Lambang/Singkatan	Arti dan Keterangan
TBS	Tandan buah segar kelapa sawit
UAV	<i>Unmanned Aerial Vehicle</i>
RGB	Model warna (<i>Red, Green, Blue</i>)
HSV	Model warna (<i>Hue, Saturation, Value</i>)
CNN	<i>Convolutioanl Neural Network</i>
YOLO	<i>You Only Look Once</i>
NMS	<i>Non-Maximum Suppresion</i>
FPN	<i>Feature Pyramid Network</i>
mAP	<i>mean Average Precision</i>
RMSE	<i>Root Mean Square Error</i>
MAE	<i>Mean Absolut Error</i>
a	Semi-sumbu mayor
b	Semi-sumbu minor
h, k	Titik koordinat pusat elips
β	Faktor skala augmentasi
GUI	<i>Graphical User Interface</i>



DAFTAR LAMPIRAN

Lampiran 1 Dataset primer	92
Lampiran 2 Hasil ekstraksi fitur dari <i>YOLOv8-Segmentation</i>	109
Lampiran 3 <i>Source code training</i> dan <i>testing</i> model segmentasi	116
Lampiran 4 <i>Source code training</i> dan <i>testing</i> model estimasi massa	117
Lampiran 5 Tampilan program sistem estimasi massa TBS kelapa sawit.....	121
Lampiran 6 Dokumentasi akuisisi dataset.....	123



KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh.

Puji syukur penulis panjatkan kepada Tuhan yang Maha Esa, karena berkat rahmat serta karunia-Nya sehingga penulis dapat menyelesaikan tugas akhir atau skripsi dengan judul “Sistem Estimasi Massa Tandan Buah Segar Kelapa Sawit Menggunakan Citra *Unmanned Aerial Vehicle* Dengan Model *Random Forest Regression*” sebagai salah satu syarat utama dalam menyelesaikan jenjang pendidikan Strata-1 pada Program Studi Teknik Informatika, Fakultas Teknik, Universitas Hasanuddin.

Penulis sadar bahwa untuk mencapai titik ini, banyak bantuan dan dukungan yang penulis terima dari berbagai pihak. Oleh karena itu, di kesempatan ini penulis ingin menyampaikan banyak terima kasih kepada:

1. Allah SWT atas berkat dan karunia-Nya sehingga penulis dimudahkan dalam mengerjakan tugas akhir ini.
2. Keluarga kecil penulis, kedua orang tua yang selalu penulis sayangi, Bapak Sumingan dan Ibu Jasmiati, S.Pd yang selalu mendidik penulis, memberikan dukungan tiada henti dan pengertian atas segala proses yang penulis jalani selama ini. Serta adik penulis, Ahmad Alzam sebagai *support system* penulis dalam mendukung impiannya kelak di masa depan.
3. Bapak Prof. Dr. Ir. Indrabayu, S.T., M.T., M.Bus.Sys., IPM., ASEAN.Eng selaku Pembimbing I dan Ibu Elly Warni, S.T., M.T. selaku Pembimbing II yang banyak menghabiskan tenaga, waktu dan ilmu untuk membimbing penulis dalam menyelesaikan tugas akhir ini dengan baik.
4. Bapak Sofyan Tandungan, S.T., M.T. yang banyak membantu penulis dalam menyelesaikan berbagai permasalahan yang penulis hadapi dalam mengerjakan tugas akhir ini.
5. Bapak Ibu Dosen dan Staf Departemen Teknik Informatika, Fakultas Teknik, Universitas Hasanuddin yang sudah bekerja keras dan berusaha sekuat tenaga mengajarkan segala yang bisa beliau berikan ke penulis dan mendukung banyak usaha dari penulis untuk menjadi lulusan yang lebih baik.



6. Farhan, Satria, Rayyan, Leon, Briel, Khaedar, Ilham, Fatur, Arif, Arfani, Mhia, Reskita, Dea, Putri, dan teman-teman kelas C yang jadi tempat pulang penulis bila penat dengan kehidupan. Menjadi orang-orang pertama yang membantu penulis bila penulis dalam kesulitan. Panjang umur ki' semua.
7. Giga, Sabda, Khalik, Wira, Agun, Dila, Uca, Ikhsan, Kak Herlina, Kak Abdi, Pak Baizul dan kakak-kakak senior di lab Kecerdasan Buatan Universitas Hasanuddin yang tidak lelah mendengar keluh kesah penulis dan menjadi teman diskusi atas kendala-kendala yang penulis dapatkan selama mengerjakan tugas akhir ini.
8. S19NIFIER, sang gajah yang berdiri gagah memandang masa depan yang cerah. Semua doa tulus penulis kirimkan ke seluruh pejuang S19NIFIER agar dimudahkan segala hal yang mereka impikan dan kelak bisa berkumpul kembali bagi kawan-an gajah yang tak kan pernah goyah.
9. Serta seluruh pihak yang tak sempat penulis sebutkan satu per satu yang telah terlibat dalam penyusunan tugas akhir ini.

Akhirnya dengan segala kerendahan hati, penulis menyadari bahwa tugas akhir ini masih memiliki kekurangan dan jauh dari kata sempurna. Oleh karena itu, penulis berharap adanya saran dan kritik membangun demi kebermanfaatan tugas akhir ini. Dan juga, semoga tugas akhir ini dapat memberikan manfaat bagi pembaca terlebih juga untuk penulis.

Wassalamu'alaikum Warahmatullahi Wabarakatuh.

Gowa, Juni 2024

Penulis,
Muhammad Ijlal Nurhadi



BAB I PENDAHULUAN

1.1 Latar Belakang

Industri kelapa sawit memiliki peran krusial dalam sektor pertanian di banyak negara tropis, termasuk Indonesia, yang merupakan salah satu produsen utama minyak kelapa sawit di dunia. Minyak kelapa sawit merupakan bahan baku utama dalam banyak produk, seperti makanan, kosmetik, dan biodiesel (WWF, 2022). Keberlanjutan dan efisiensi dalam industri kelapa sawit tidak hanya mempengaruhi pertumbuhan ekonomi negara-negara produsen, tetapi juga berdampak besar pada ekonomi global secara keseluruhan.

Salah satu aspek penting dalam pengelolaan perkebunan kelapa sawit adalah perencanaan pra-panen. Perencanaan pra-panen yang baik dapat mengoptimalkan hasil panen dan memaksimalkan keuntungan (Henson, 2012). Proses ini melibatkan estimasi hasil panen, yang biasanya dilakukan dengan menimbang Tandan Buah Segar (TBS) kelapa sawit dari beberapa pohon secara acak untuk mengukur massa rata-rata. Keakuratan estimasi hasil panen bukan hanya memiliki implikasi signifikan pada perkebunan individual, tetapi juga membawa dampak langsung pada industri kelapa sawit secara keseluruhan.

Metode tradisional, meskipun umum digunakan, memiliki beberapa kelemahan yang signifikan. Ketidakakuratan *sampling* sering terjadi karena variasi besar dalam ukuran dan massa TBS antar pohon, sementara proses penimbangan manual memerlukan waktu dan tenaga yang signifikan, sehingga kurang efisien untuk perkebunan berskala besar (Oushabi dkk., 2015). Selain berguna dalam perancangan strategi pra-panen, evaluasi kinerja perkebunan, termasuk analisis terhadap hasil panen dan produktivitas, memerlukan data massa TBS kelapa sawit untuk memberikan gambaran yang sebenarnya (Fackrurrozi dkk., 2019).

Estimasi massa TBS kelapa sawit menjadi elemen krusial dalam konteks pemantauan keberlanjutan, analisis produktivitas dan optimasi perencanaan produksi pra-panen dalam industri kelapa sawit. Dengan analisis produktivitas dan optimasi perencanaan produksi pra-panen yang lebih akurat, para pelaku industri



dapat meningkatkan efisiensi operasional, mengurangi dampak lingkungan, dan merancang strategi produksi yang berkelanjutan secara ekonomi (Corley, 2018).

Penerapan teknologi visi komputer dan regresi diusulkan sebagai solusi untuk mengatasi kelemahan dalam estimasi massa TBS kelapa sawit. Visi komputer dapat digunakan untuk mendeteksi dan mengukur TBS kelapa sawit secara dari citra *Unmanned Aerial Vehicle* (UAV), sementara regresi dapat memberikan model matematis yang dapat memprediksi massa TBS kelapa sawit berdasarkan variabel dari citra tersebut.

Citra UAV dapat menghasilkan gambar yang rinci dan memberikan data spasial yang penting untuk identifikasi dan pengukuran TBS kelapa sawit (Perros dkk., 2021). Teknologi visi komputer bekerja secara otomatis dalam mendeteksi dan mengukur TBS secara individual dari citra ini. Dengan analisis citra yang cermat, informasi yang diperoleh dapat digunakan sebagai dasar untuk membangun model matematis menggunakan regresi. Regresi, sebagai alat analisis sistematis, memperoleh variabel-variabel yang diperoleh dari citra UAV untuk mengestimasi massa TBS kelapa sawit. Dengan demikian, estimasi yang dihasilkan dapat menjadi dasar yang baik untuk perencanaan pra-panen kelapa sawit.

Dari penelitian yang dilakukan oleh Kalantar dkk. (2020) mengembangkan algoritma deteksi melon menggunakan *RetinaNet*. Model tersebut mencapai tingkat deteksi tinggi (*average precision* $\approx 0,92$, *F1 score* $> 0,9$) dalam berbagai kondisi pertanian. Estimasi massa individu melon menggunakan regresi berbasis fitur geometri mendapat MAPE sekitar 16% sementara estimasi massa keseluruhan memiliki kesalahan sekitar 3% (Kalantar dkk., 2020).

Penelitian lainnya juga dilakukan oleh Basak dkk. (2022) untuk mengestimasi massa buah stroberi dengan membandingkan algoritma Regresi Linear dan *Support Vector Regression* (SVR). Hasilnya, regresi linear lebih unggul dengan $R^2 = 0,896$ dan $RMSE = 1,054$, sementara SVR mendapat $R^2 = 0,856$ dan $RMSE = 1,239$ (Basak dkk., 2022).

Berdasarkan latar belakang permasalahan tersebut, penulis membangun sistem estimasi massa tandan buah segar kelapa sawit. Hasil dari penelitian ini dapat memberikan informasi yang berharga bagi petani dan produsen. Mereka dapat menggunakan data ini untuk membuat keputusan yang lebih baik dalam



perencanaan pra-panen, pengelolaan perkebunan, dan pemrosesan hasil. Dengan cara ini dapat meningkatkan efisiensi produksi dan mengurangi kerugian yang disebabkan oleh panen yang tidak tepat sasaran.

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah, maka rumusan masalah dalam penelitian ini adalah:

1. Bagaimana merancang dan mengimplementasikan model visi komputer yang bisa mengestimasi massa tandan buah segar kelapa sawit?
2. Bagaimana kinerja model visi komputer dalam mengestimasi massa tandan buah segar kelapa sawit?

1.3 Tujuan Penelitian

Tujuan dari penelitian ini adalah:

1. Merancang dan mengimplementasikan model visi komputer yang bisa mengestimasi massa tandan buah segar kelapa sawit.
2. Mengevaluasi kinerja model visi komputer dalam mengestimasi massa tandan buah segar kelapa sawit.

1.4 Manfaat Penelitian

Penelitian ini dapat membantu:

1. Bagi industri, penelitian ini memungkinkan efisiensi produksi, mengoptimalkan manajemen sumber daya, estimasi massa yang lebih baik dan efisiensi biaya yang semuanya berkontribusi pada peningkatan produktivitas.
2. Bagi akademisi, penelitian ini dapat memberikan peluang penelitian lanjutan, publikasi ilmiah, dan kolaborasi industri-akademisi, juga menyumbangkan pengetahuan dan teknologi untuk mendukung pertumbuhan sektor pertanian dan inovasi teknologi.



1.5 Ruang Lingkup

Ruang lingkup dari penelitian ini adalah:

1. Data yang digunakan adalah data citra tandan buah segar kelapa sawit serta massa aktualnya.
2. Menggunakan *drone* DJI Mavic Air untuk akuisisi citra dengan kualitas 1080p 30fps.
3. Jarak dari *drone* ke TBS kelapa sawit ditentukan oleh stereo kamera *drone* DJI Mavic Air.
4. Akuisisi citra dilakukan pada siang hari dengan kondisi cuaca baik (tidak mendung/hujan).
5. Proses komputasi dan visualisasi hasil dilakukan di laptop.



BAB II TINJAUAN PUSTAKA

2.1 Kelapa Sawit

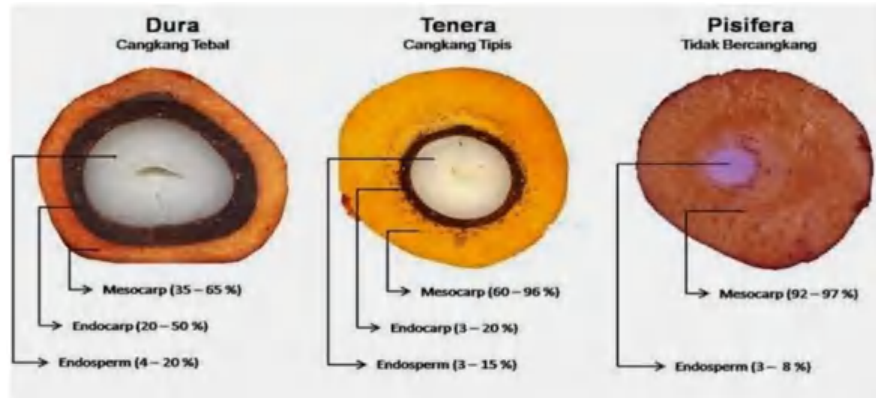
Kelapa sawit/*oil palm* adalah sejenis tumbuhan yang termasuk ke dalam genus *Elaeis* dan ordo *Arecaceae* yang dimana morfologi dari ordo *Arecaceae* itu sendiri merupakan sejenis tumbuhan palm yang memiliki buah berwarna kemerahan dengan ukuran plum besar dan tumbuh dalam tandan besar. Kelapa sawit sendiri tumbuhan industri/perkebunan yang buahnya dapat dimanfaatkan menjadi minyak baik dalam olahan minyak masak, minyak industri, maupun bahan bakar. Kelapa sawit terdiri dari dua spesies yaitu *Elaeis guineensis* dan *Elaeis oleifera* yang digunakan untuk pertanian komersial dalam pengeluaran minyak kelapa sawit. Pohon *Elaeis guineensis*, berasal dari Afrika barat di antara Angola dan Gambia, sedangkan *Elaeis oleifera*, berasal dari Amerika tengah dan Amerika Selatan (Ratnaningsih dkk., 2022).

Kelapa sawit merupakan tumbuhan monokotil yang tidak memiliki akar tunggang, umumnya memiliki batang yang tidak bercabang, memiliki daun (*Fronde*) yang pangkal pelepah daun terbentuk dua baris duri yang sangat tajam dan keras di kedua sisinya dengan anak daun (*Foliage leaflet*) tersusun berbaris dua sampai ke ujung daun. Kelapa sawit yang berumur tiga tahun sudah mulai dewasa dan mulai mengeluarkan bunga jantan yang berbentuk lonjong memanjang, dan bunga betina yang agak bulat. Kelapa sawit cocok dibudidayakan pada ketinggian lahan antara 0-500 meter dari permukaan laut dengan kemiringan lereng sebesar 0-3% dan pH 5,5-7,0 (Probolinggo, 2024).

Kelapa sawit jenis *Elaeis guineensis* yang pertama kali dan terluas dibudidayakan di Indonesia. Penangkar sering kali melihat tipe kelapa sawit berdasarkan ketebalan cangkang yang terdiri dari: *Dura*, *Pisifera* dan *Tenera*. *Dura* merupakan sawit yang buahnya memiliki cangkang tebal sehingga dianggap memperpendek umur mesin pengolah namun biasanya tandan buahnya besar-besar dan kandungan minyak per tandannya berkisar 18%. *Pisifera* buahnya tidak memiliki cangkang namun bunga betinanya steril sehingga sangat jarang menghasilkan buah. *Tenera* adalah persilangan antara induk *Dura* dan jantan

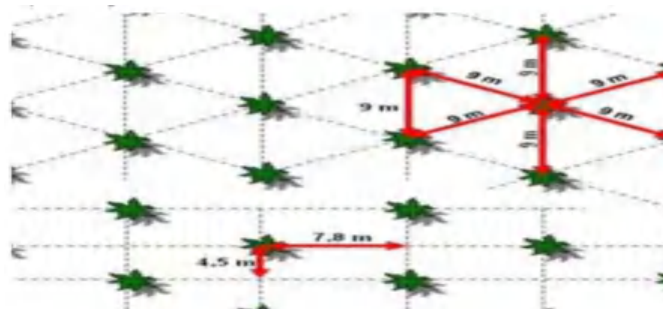


Pisifera. Jenis ini dianggap bibit unggul sebab melengkapi kekurangan masing-masing induk dengan sifat cangkang buah tipis namun bunga betinanya tetap fertil. Beberapa *Tenera* unggul memiliki persentase daging per buahnya mencapai 90% dan kandungan minyak pertandannya dapat mencapai 28% (Silitonga dkk., 2020). Perbedaan *Dura*, *Tenera*, dan *Pisifera* dapat dilihat pada Gambar 1.



Gambar 1 Perbedaan *Dura*, *Tenera*, dan *Pisifera*
Sumber: (Silitonga dkk., 2020)

Jarak tanam yang optimal untuk kelapa sawit adalah faktor penting yang mempengaruhi pertumbuhan tanaman dan hasil produksi. Rekomendasi umum untuk jarak tanam kelapa sawit adalah sekitar 8 hingga 9 meter antar pohon, yang biasanya menghasilkan jarak tanam berbentuk segitiga sama sisi atau pola segitiga (*triangular planting*). Ini menghasilkan kepadatan sekitar 136 hingga 148 pohon per hektar. Pola tanam ini memastikan bahwa setiap pohon memiliki ruang yang cukup untuk tumbuh dengan baik, mendapatkan sinar matahari yang cukup, dan tidak bersaing secara berlebihan untuk air dan nutrisi dari tanah (Silitonga dkk., 2020). Ilustrasi jarak tanam antar pohon kelapa sawit dapat dilihat pada Gambar 2.



Gambar 2 Jarak tanam kelapa sawit
sumber: (Silitonga dkk., 2020)



Kelapa sawit merupakan tanaman yang dapat dipanen tiga sampai empat tahun setelah tanam, akan tetapi buah termasuk kategori matang apabila sudah berumur 6 bulan setelah proses penyerbukan. Di perkebunan rakyat umumnya proses panen dikelola oleh petani dengan luas lahan yang lebih terbatas. Proses panen di perkebunan rakyat sering kali lebih sederhana dan bergantung pada tenaga kerja manual. Tahapan panen dimulai dengan memeriksa kematangan buah. Buah kelapa sawit dianggap matang ketika 5-10 buah telah jatuh dari tandan. Petani kemudian menggunakan alat tradisional seperti dodos dan egrek untuk memotong tandan buah segar (TBS) dari pohon. Massa TBS berada di rentang sepuluh kilogram hingga sembilan puluh lima kilogram (GAPKI, 2018). Setelah TBS dipotong, kemudian dikumpulkan dan diangkut ke tempat pengumpulan sementara menggunakan keranjang atau gerobak kecil sebelum akhirnya dibawa ke pabrik pengolahan (Rist dkk., 2010).

2.2 Citra Digital

Citra digital adalah representasi visual dari objek atau pemandangan yang disimpan dalam format digital. Citra ini terdiri dari piksel-piksel yang disusun dalam matriks, dengan setiap piksel menyimpan informasi tentang warna dan intensitas cahaya. Sumber citra digital dapat berasal dari kamera digital, pemindai atau perangkat lunak grafis komputer (C. Gonzalez & E. Woods, 2018).

Warna pada citra digital biasanya direpresentasikan dengan model warna RGB (*Red, Green, Blue*). Setiap piksel dalam citra berwarna terdiri dari tiga komponen warna dasar ini, dengan setiap komponen memiliki nilai intensitas yang berkisar dari 0 hingga 255. Kombinasi dari ketiga komponen ini menentukan warna akhir dari piksel tersebut (Szeliski, 2010). Representasi warna RGB dapat dilihat pada Gambar 3.



Gambar 3 Representasi model warna RGB dan komponen individualnya.
 Sumber: (Szeliski, 2010)



Model warna RGB paling umum digunakan karena mendekati cara mata manusia menangkap warna. Selain RGB, ada model warna lain seperti CMYK (*Cyan, Magenta, Yellow, Key*) yang digunakan dalam pencetakan, dan model warna HSV (*Hue, Saturation, Value*) yang sering digunakan dalam grafis komputer untuk memudahkan manipulasi warna (C. Gonzalez & E. Woods, 2018).

Citra berwarna menawarkan lebih banyak informasi dibandingkan citra grayscale (hitam putih) yang hanya menyimpan intensitas Cahaya tanpa informasi warna. Oleh karena itu, citra berwarna lebih kaya dan memungkinkan analisis yang lebih mendetail dalam aplikasi seperti pengenalan objek, pengolahan citra medis, dan visi komputer (Szeliski, 2010).

2.3 Drone DJI Mavic Air

Drone DJI Mavic Air merupakan *drone* yang dirancang untuk keseimbangan antara portabilitas dan fitur-fitur canggih. *Drone* ini dilengkapi dengan kamera 12 megapiksel yang mampu merekam video 4K pada 30 *frame per second* (fps). Kamera tersebut dipasang pada gimbal tiga sumbu, yang memastikan stabilitas gambar dan video yang diambil tetap *smooth* meskipun dalam kondisi terbang yang tidak stabil (DJI, 2018). Tampilan *drone* DJI Mavic Air dapat dilihat pada Gambar 4.



Gambar 4 *Drone* DJI Mavic Air
 Sumber: (DJI, 2018)

Waktu terbang *drone* DJI Mavic Air bisa mencapai 21 menit per pengisian baterai yang cukup untuk banyak keperluan fotografi udara dan video. *Drone* ini memiliki jarak kontrol hingga 4 kilometer yang memberikan fleksibilitas besar dalam eksplorasi dan pengambilan gambar dari jarak jauh (Joshua Goldman, 2018).

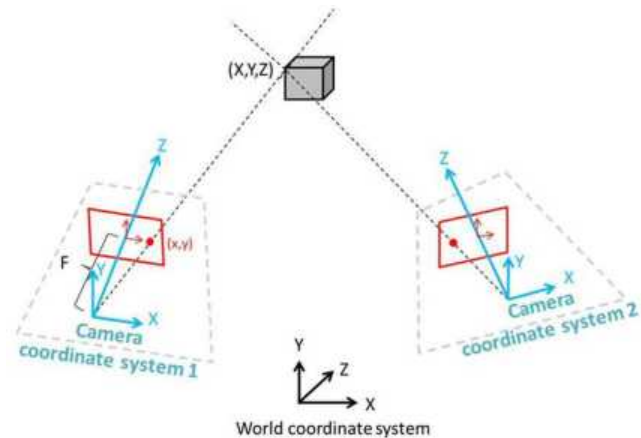


Salah satu fitur utama *drone* DJI Mavic Air adalah kemampuannya dalam mendeteksi jarak. Fitur ini berguna untuk menghindarkan *drone* dari rintangan yang ada di sekitarnya. Untuk mendeteksi jarak di depannya, *drone* DJI Mavic Air dilengkapi *obstacle sensing system* yang terdiri dari stereo kamera. Stereo kamera bekerja dengan mengukur jarak antara *drone* dan objek di depannya melalui teknologi *stereo vision*, menggunakan dua kamera untuk menangkap gambar dan kemudian memproses dan memperkirakan jarak (DJI, 2020). Letak stereo kamera pada *drone* DJI Mavic Air dapat dilihat pada Gambar 5.



Gambar 5 Stereo kamera pada DJI Mavic Air
 Sumber: (DJI, 2020)

Kamera-kamera ini menangkap gambar secara simultan dari sudut pandang yang sedikit berbeda. Kedua kamera ini kemudian mengambil gambar dari lingkungan yang ada di depannya. Setiap kamera menghasilkan gambar 2D dari perspektifnya sendiri. Drone akan menganalisis dua gambar ini untuk menentukan titik-titik yang cocok di kedua gambar. Proses ini dikenal sebagai *stereo matching*. Dengan membandingkan posisi titik-titik yang cocok di kedua gambar, sistem dapat mengukur perbedaan paralaks, yaitu perbedaan posisi antara titik yang sama di dua gambar. Ilustrasi prinsip triangulasi dapat dilihat pada Gambar 6.



Gambar 6 Prinsip triangulasi menggunakan stereo kamera
 Sumber: (Ekberg dkk., 2017)

Sistem kemudian menghitung jarak setiap titik dari kamera menggunakan prinsip triangulasi. Semakin besar perbedaan paralaks, semakin dekat objek tersebut ke *drone*. Hasil dari kalkulasi jarak di berbagai titik kemudian digabungkan dan membentuk peta kedalaman 3D dari lingkungan yang ada di depan *drone*. Peta kedalaman ini memberikan informasi tentang jarak relatif dari berbagai objek di depan *drone*. Tampilan deteksi jarak muncul pada *remote control drone* DJI Mavic Air sebagaimana ditunjukkan pada Gambar 7.



Gambar 7 Estimasi jarak pada *drone* DJI Mavic Air

Selain stereo kamera di depan, *drone* DJI Mavic Air juga dilengkapi dengan sensor visual di bawah dan belakang drone. Sistem ini dikenal sebagai *Advanced Pilot Assistance Systems (APAS)* yang memberikan DJI Mavic Air deteksi yang menyeluruh dan meningkatkan stabilitas penerbangan, terutama saat terbang rendah atau mendarat (DJI, 2020).



2.4 Visi Komputer

Visi komputer adalah bidang dalam ilmu komputer yang berfokus pada kemampuan komputer untuk memahami dan menginterpretasi dunia visual dengan cara yang mirip dengan manusia. Tujuan utamanya adalah untuk memungkinkan komputer mengekstrak informasi yang bermakna dari gambar dan video. Ini melibatkan proses pengenalan, pelacakan, dan interpretasi objek dan pola dalam data visual untuk mengidentifikasi fitur-fitur penting dan memahami konteks keseluruhan dari gambar tersebut (Voulodimos dkk., 2018).

Visi komputer memiliki hubungan yang erat dengan pengolahan citra. Visi komputer melibatkan proses otomatis yang menggabungkan sejumlah langkah untuk memahami informasi visual. Ini termasuk akuisisi citra, *preprocessing* citra, klasifikasi, pengenalan pola, dan pengambilan keputusan. Proses ini mirip dengan cara manusia memproses informasi visual, meskipun dilakukan secara otomatis oleh komputer.

Visi komputer memiliki aplikasi yang luas dalam berbagai bidang. Salah satunya dalam bidang pertanian dimana visi komputer dapat meningkatkan efisiensi dan hasil produksi. Salah satu contoh penerapannya adalah dalam pemantauan tanaman. Dengan menggunakan *drone* yang dilengkapi dengan kamera dan algoritma visi komputer, petani dapat memantau kesehatan tanaman, mendeteksi penyakit, dan serangan hama dengan menganalisis perubahan warna dan pola pada tanaman. Hal ini memungkinkan tindakan pencegahan yang cepat dan tepat, mengurangi kerugian dan meningkatkan hasil panen (Kamilaris & Prenafeta-Boldú, 2018).

2.5 Convolutional Neural Networks

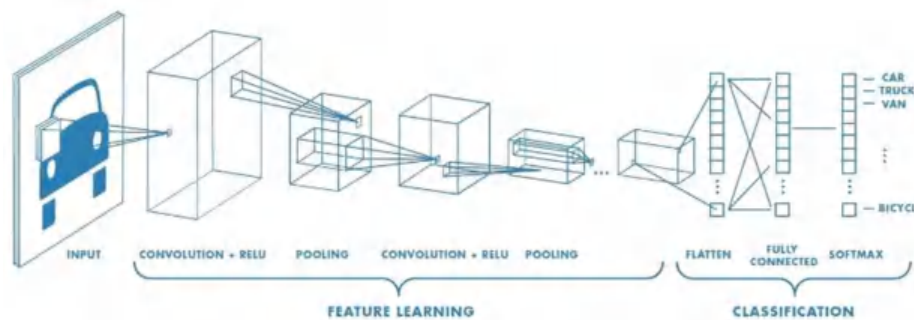
Convolutional Neural Networks (CNN) adalah jenis arsitektur *deep learning* yang dirancang khusus untuk pemrosesan data *grid* seperti citra. CNN terdiri dari beberapa *layer* yang saling berhubungan yang masing-masing menjalankan fungsi tertentu untuk mengekstrak fitur dari input. Struktur dasar CNN terdiri dari *convolution layer*, *pooling layer*, dan *fully connected layer* (Goodfellow dkk., 2016).



Convolution layer adalah komponen utama dari CNN, dimana filter kecil (atau kernel) diaplikasikan pada seluruh citra input untuk menghasilkan *feature map*. Filter ini bergerak melintasi gambar dan menghitung produk titik antara filter dan bagian dari gambar input, menghasilkan aktivasi yang menunjukkan fitur seperti tepi, tekstur, atau pola.

Setelah *convolution layer*, *pooling layer* diterapkan untuk mengurangi dimensi *feature map* sambil tetap mempertahankan informasi pentingnya. *Max pooling* adalah teknik yang paling umum digunakan, di mana nilai maksimum dalam setiap jendela filter diambil sebagai representasi dari fitur tersebut. *Pooling* membantu mengurangi jumlah parameter dan komputasi dalam jaringan, serta membuat model lebih tahan terhadap pergeseran dan distorsi gambar.

Fully connected layer, yang sering ditempatkan di bagian akhir CNN, menggabungkan semua *neuron* dari *layer* sebelumnya dan menghubungkannya ke setiap *neuron* di *layer* berikutnya. Ini mirip dengan *layer* di *feedforward neural networks* tradisional dan digunakan untuk melakukan tugas klasifikasi berdasarkan fitur-fitur yang telah diekstraksi oleh *layer* sebelumnya (Goodfellow dkk., 2016). Arsitektur jaringan CNN untuk klasifikasi gambar ditunjukkan pada Gambar 8.



Gambar 8 Arsitektur jaringan CNN untuk klasifikasi gambar
 Sumber: (Saha, 2018)

Training CNN dilakukan melalui proses *backpropagation*, di mana kesalahan prediksi model dihitung dan digunakan untuk memperbarui *weight filter* melalui algoritma *gradient descent*. Ini memungkinkan model belajar dan mengoptimalkan filter untuk mengekstraksi fitur yang relevan secara otomatis dari data input (Lecun dkk., 2015).

CNN telah terbukti sangat efektif dalam berbagai aplikasi visi komputer seperti pengenalan objek, deteksi, dan segmentasi citra. CNN juga digunakan dalam



bidang lain seperti pengenalan suara, analisis teks, dan bahkan video game. Keberhasilan CNN Sebagian besar disebabkan oleh kemampuannya untuk menangkap hierarki fitur, dari yang sederhana hingga kompleks, yang penting untuk tugas pemrosesan citra (Lecun dkk., 2015).

2.6 Preprocessing

Preprocessing adalah tahap awal yang penting dalam pengolahan data yang bertujuan untuk mempersiapkan data mentah agar siap digunakan dalam analisis lebih lanjut. Proses ini melibatkan berbagai teknik seperti normalisasi, pembersihan, dan transformasi data untuk memastikan data berada dalam format yang konsisten dan dapat dianalisis secara efektif (Bishop, 2006).

1. *Image Resizing*

Image Resizing adalah proses mengubah ukuran gambar dengan mempertahankan aspek rasio atau proporsinya. Proses ini melibatkan beberapa metode, termasuk interpolasi *bilinear* dan *bicubic*, yang digunakan untuk menentukan nilai piksel baru berdasarkan piksel-piksel sekitarnya.

Interpolasi *bilinear* menghitung nilai piksel baru dengan menggunakan nilai rata-rata dari piksel terdekat di sekitar piksel yang ingin diubah. Teknik ini relatif cepat dan menghasilkan gambar yang halus namun bisa sedikit buram. Interpolasi *bicubic*, di sisi lain, menggunakan 16 piksel terdekat untuk menghitung nilai piksel baru, memberikan hasil yang lebih halus dan lebih detail, namun memerlukan lebih banyak komputasi.

Resizing juga melibatkan pemeliharaan aspek rasio, yang berarti perbandingan lebar dan tinggi gambar tetap sama agar gambar tidak terdistorsi. Ini sering dilakukan dengan menambahkan *padding* atau memotong gambar sesuai kebutuhan. Dalam aplikasi praktisnya, seperti pengolahan citra dalam visi komputer, ukuran gambar sering diubah agar sesuai dengan dimensi input model, memastikan konsistensi dan efisiensi dalam pemrosesan data (C. Gonzalez & E. Woods, 2018).



2. Normalisasi Data

Normalisasi data adalah proses transformasi data mentah menjadi bentuk yang lebih konsisten dan terstruktur. Proses ini sangat penting dalam analisis data dan *machine learning* karena membantu memastikan bahwa fitur-fitur dalam dataset berkontribusi secara proporsional terhadap hasil akhir model. Tanpa normalisasi, fitur-fitur dengan skala nilai yang besar dapat mendominasi fitur-fitur dengan skala nilai yang kecil, yang dapat mempengaruhi kinerja dan akurasi model (Bishop, 2006).

2.7 Augmentasi Citra

Augmentasi citra adalah teknik dalam pengolahan citra yang digunakan untuk meningkatkan variasi data *training* tanpa menambah jumlah gambar secara fisik. Teknik ini dilakukan dengan menerapkan transformasi pada citra sehingga menghasilkan dataset yang lebih bervariasi dan membantu dalam meningkatkan kinerja model dengan membuatnya lebih *robust* terhadap perubahan dan variasi dalam data input (Shorten & Khoshgoftaar, 2019).

1. Saturasi

Saturasi dalam augmentasi citra melibatkan penyesuaian intensitas warna dalam gambar, yang dapat dilakukan dengan mengubah nilai saturasi setiap piksel dalam model warna seperti HSV (*Hue, Saturation, Value*). Contoh perubahan saturasi pada citra ditunjukkan pada Gambar 9.



Gambar 9 Perbedaan saturasi warna pada citra

Sumber: (Bansi, 2020)



Dengan menyesuaikan saturasi, warna citra bisa menjadi lebih cerah atau lebih pudar, memungkinkan model untuk mengenali objek dalam berbagai kondisi pencahayaan dan warna, yang juga meningkatkan kemampuan generalisasi dan akurasi model selama proses *training*. Rumus dasar untuk penyesuaian saturasi menggunakan persamaan (1).

$$I_{new} = I_{original} + \beta \quad (1)$$

dimana,

I_{new} = intensitas piksel baru,

$I_{original}$ = intensitas piksel asli,

β = faktor skala yang menentukan tingkat saturasi.

β menjadi penentu dimana berguna untuk menentukan seberapa banyak saturasi ditingkatkan atau dikurangi. Nilai β lebih besar dari 1 akan meningkatkan saturasi, sementara nilai antara 0 dan 1 akan menguranginya (Shorten & Khoshgoftaar, 2019).

2. Kecerahan

Kecerahan dalam augmentasi citra mengacu pada penyesuaian tingkat kecerahan gambar. Teknik ini menambah variasi dalam data *training* dengan mengubah intensitas cahaya pada gambar, membantu model untuk lebih *robust* terhadap perubahan pencahayaan. Contoh perubahan kecerahan pada citra ditunjukkan pada Gambar 10.



Gambar 10 Perbedaan kecerahan pada gambar

Sumber: (Bansi, 2020)



Pengaturan kecerahan pada citra dilakukan dengan menambahkan atau mengurangi nilai konstan pada setiap piksel dalam gambar. Rumus dasar untuk penyesuaian brightness menggunakan persamaan (2).

$$I_{new} = I_{original} + \beta \quad (2)$$

dimana,

I_{new} = intensitas piksel baru,

$I_{original}$ = intensitas piksel asli,

β = faktor kecerahan yang ditambahkan atau dikurangi.

Faktor β dapat berupa nilai positif untuk meningkatkan kecerahan atau negatif untuk menguranginya (Shorten & Khoshgoftaar, 2019).

3. Eksposur

Eksposur dalam augmentasi citra mengacu pada penyesuaian jumlah cahaya yang mempengaruhi keseluruhan citra. Teknik ini digunakan untuk menambah variasi dalam data *training* dengan mengubah tingkat kecerahan gambar secara keseluruhan, sehingga membantu model untuk lebih *robust* terhadap variasi pencahayaan dalam data nyata. Contoh perubahan kecerahan pada citra ditunjukkan pada

Gambar 11.



Gambar 11 Perbedaan eksposur pada gambar
 Sumber: (Bansi, 2020)

Penyesuaian eksposur dilakukan dengan mengubah nilai intensitas setiap piksel dalam gambar menggunakan persamaan (3).

$$I_{new} = I_{original} + \beta \quad (3)$$



dimana,

I_{new} = intensitas piksel baru,

$I_{original}$ = intensitas piksel asli,

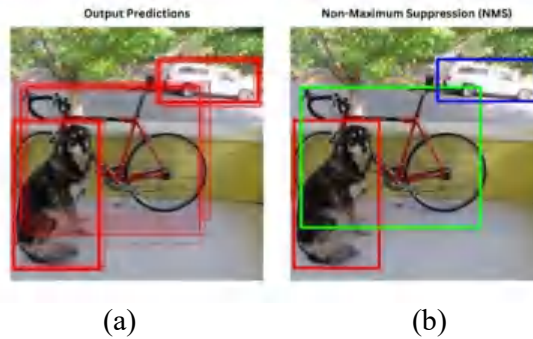
β = faktor exposur yang ditambahkan atau dikurangi.

Faktor β dapat berupa nilai positif untuk meningkatkan eksposur atau negatif untuk menguranginya (Shorten & Khoshgoftaar, 2019).

2.8 You Only Look Once

You Only Look Once (YOLO) adalah algoritma deteksi objek dalam gambar yang menggunakan *Convolutional Neural Network* (CNN). Algoritma ini dirancang untuk mendeteksi dan mengklasifikasikan objek dalam gambar secara *real-time* dengan memproses gambar secara keseluruhan hanya dalam satu kali perhitungan. Pendekatan ini berbeda dengan metode deteksi objek tradisional yang memproses gambar bagian per bagian, yang membuat YOLO sangat cepat dan efisien (Redmon dkk., 2016).

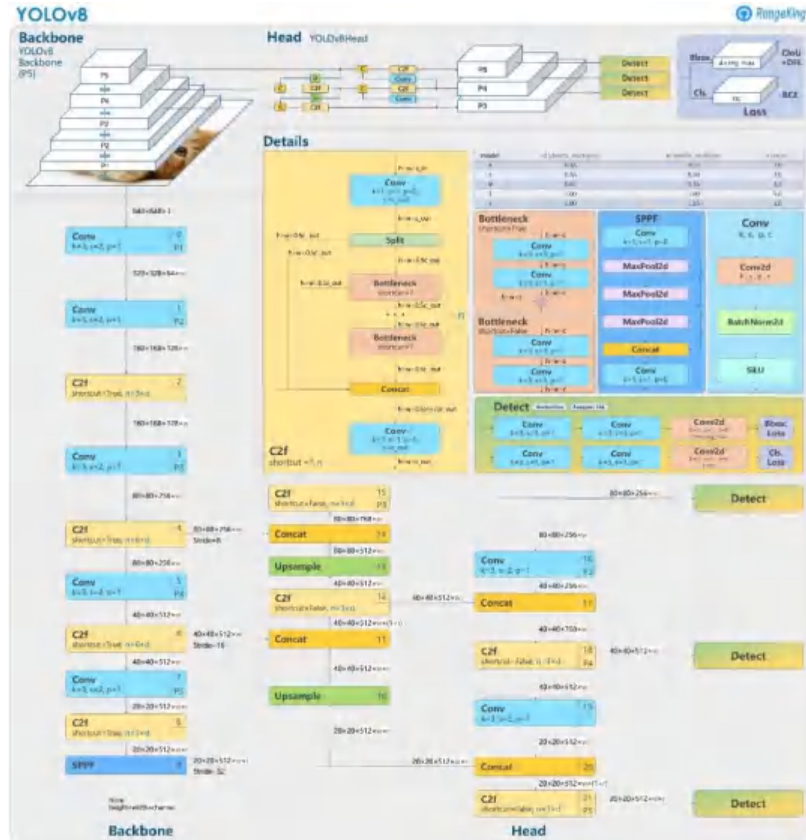
Citra akan dibagi menjadi grid berukuran $S \times S$, dan setiap *sel grid* bertanggung jawab untuk mendeteksi objek di dalam area tersebut. Setiap sel memprediksi beberapa *bounding box* yang mencakup koordinat pusat, lebar, tinggi, *confidence score*, dan kelas objek. YOLO menggabungkan prediksi *bounding box* dan kelas dengan *confidence score* untuk menghasilkan prediksi akhir. Teknik *Non-Maximum Suppression* (NMS) digunakan untuk menghilangkan prediksi duplikat dengan mempertahankan *bounding box* dengan skor tertinggi dan menghapus kotak lain yang memiliki *overlap* signifikan (Redmon dkk., 2016). Perbandingan *output* sebelum dan sesudah menggunakan NMS ditunjukkan pada Gambar 12.



Gambar 12 Output prediksi sebelum dan sesudah (penerapan NMS)
 Sumber: (Redmon dkk., 2016)

Berdasarkan Gambar 12 dapat dilihat bahwa gambar yang belum menerapkan NMS (a) menghasilkan banyak *bounding box* untuk setiap objek yang terdeteksi pada citra, kemudian gambar yang telah menerapkan NMS (b) hanya menyisahkan satu *bounding box* untuk tiap objek yang terdeteksi pada citra.

Arsitektur YOLO terdiri dari beberapa *convolution layer* untuk mengekstraksi fitur dari citra, diikuti oleh *pooling layer* untuk mengurangi dimensi data fitur, dan *fully connected layer* untuk membuat prediksi akhir. Jaringan ini memproses gambar melalui beberapa tahap: *input image processing*, *feature extraction*, *bounding box*, *class prediction*, dan *post-processing* dengan NMS. YOLOv3, misalnya menggunakan *Darknet-53* sebagai *backbone*, yang terdiri dari 53 *convolution layer* dengan *residual connections*. Versi terbaru, YOLOv8 memperkenalkan kemampuan segmentasi objek dan memiliki arsitektur yang terdiri dari *backbone*, *neck*, dan *head* untuk menghasilkan prediksi yang lebih akurat. Arsitektur dari YOLOv8 ditunjukkan pada Gambar 13.



Gambar 13 Arsitektur YOLOv8

Sumber: (Ultralytics, 2023)

Gambar input diubah ukurannya menjadi resolusi 640x640 piksel untuk memastikan konsistensi dalam pemrosesan. Gambar ini kemudian diproses melalui beberapa tahap dalam CNN. YOLOv8 segmentation menggunakan *backbone CSPNet* untuk mengekstraksi fitur dari gambar input. *Backbone* ini terdiri dari beberapa lapisan konvolusi yang bertujuan untuk menangkap fitur visual pada berbagai tingkatan kompleksitas.

Intermediate Processing atau *Neck* bertugas mengumpulkan dan memproses fitur dari *backbone*. *Neck* menggunakan teknik *Feature Pyramid Network (FPN)* untuk menggabungkan informasi dari berbagai level resolusi, sehingga menghasilkan representasi fitur yang kaya. Bagian *Head* dari YOLOv8 Segmentation bertanggung jawab menghasilkan prediksi *bounding boxes*, klasifikasi objek, dan *mask* segmentasi. Untuk setiap *sel grid*, *head* memprediksi sejumlah *bounding box* dengan koordinat, *confidence score* dan label kelas. Selain itu, *head* juga menghasilkan *mask* segmentasi untuk setiap objek yang terdeteksi.



Setelah prediksi awal dibuat, YOLOv8 menerapkan *Non-Maximum Suppression* (NMS) untuk menghilangkan prediksi bounding box yang duplikat dengan mempertahankan *bounding box* dengan skor tertinggi. *Mask* segmentasi kemudian diterapkan untuk menentukan area spesifik dari setiap objek yang terdeteksi. *Mask* segmentasi yang dihasilkan oleh *head* digunakan untuk memisahkan area objek dari latar belakang. Setiap piksel dalam *mask* diberi nilai yang sesuai dengan kelas objek, memungkinkan model untuk menghasilkan segmentasi yang tepat. Contoh hasil segmentasi dari YOLOv8 ditunjukkan pada Gambar 14.



Gambar 14 Segmentasi sampah bawah laut menggunakan *YOLOv8-Segmentation*
 Sumber: (Rath, 2023)

Dengan menggabungkan deteksi objek dan segmentasi dalam satu model, *YOLOv8-Segmentation* memungkinkan deteksi dan segmentasi objek dengan akurasi yang tinggi. Ini sangat berguna dalam aplikasi yang memerlukan pemahaman detail tentang area objek dalam citra. *YOLOv8-Segmentation* memiliki 5 *weight pre-trained model* yang detail performanya disajikan pada

Tabel 1.

Tabel 1 Perbandingan *weight YOLOv8-Segmentation*

Model	Size (pixels)	mAP50- 95 mask	Speed	Speed	Params (M)	FLOPs (B)
			CPU (ms)	A100 ONNX (ms)		
YOLOv8n- seg	640	30,5	96,1	1,21	3,4	12,6



Model	Size (pixels)	mAP50- 95 mask	Speed	Speed	Params (M)	FLOPs (B)
			CPU	A100		
			ONNX (ms)	TensorRT (ms)		
YOLOv8s- seg	640	36,8	155,7	1,47	11,8	42,6
YOLOv8m- seg	640	40,8	317,0	2,18	27,3	110,2
YOLOv8l- seg	640	42,6	572,4	2,79	46,0	220,5
YOLOv8x- seg	640	43,4	712,1	4,02	71,8	344,1

Sumber: (Ultralytics, 2023)

2.9 Confusion Matrix

Confusion Matrix adalah alat yang sangat berguna dalam evaluasi kinerja model klasifikasi, yang menunjukkan jumlah prediksi yang benar dan salah yang dibuat oleh model, dibandingkan dengan nilai aktual dalam dataset. Confusion Matrix berisi empat kategori utama: *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). *True Positive* adalah jumlah kasus di mana model benar-benar memprediksi kelas positif, sementara *True Negative* adalah jumlah kasus di mana model benar-benar memprediksi kelas negatif. *False Positive* terjadi ketika model salah memprediksi kelas positif, dan *False Negative* terjadi ketika model salah memprediksi kelas negatif (Bishop, 2006). Struktur *Confusion Matrix* bisa ditunjukkan pada Gambar 15.



		Actual Values	
		1 (Positive)	0 (Negative)
Predicted Values	1 (Positive)	TP (True Positive)	FP (False Positive) <i>Type I Error</i>
	0 (Negative)	FN (False Negative) <i>Type II Error</i>	TN (True Negative)

Gambar 15 Struktur dari *Confusion Matrix*
Sumber: (Nugroho, 2019)

Menggunakan *Confusion Matrix*, dapat menghitung metrik evaluasi penting seperti *Precision* dan *Recall*. *Precision* adalah rasio prediksi yang benar-benar positif dari semua prediksi positif yang dibuat oleh model. *Precision* dihitung dengan persamaan (4).

$$Precision = \frac{True\ Positives\ (TP)}{True\ Positives\ (TP) + False\ Positives\ (FP)} \quad (4)$$

Precision menunjukkan seberapa baik model dalam menghindari prediksi positif yang salah. Ini sangat penting dalam situasi dimana kesalahan positif memiliki konsekuensi yang serius, seperti dalam diagnosis medis atau deteksi spam email (Goodfellow dkk., 2016).

Recall, juga dikenal sebagai *Sensitivity* atau *True Positive Rate*, adalah rasio prediksi yang benar-benar positif dari semua kasus yang sebenarnya positif. *Recall* dihitung dengan persamaan (5).

$$Recall = \frac{True\ Positives\ (TP)}{True\ Positives\ (TP) + False\ Negatives\ (FN)} \quad (5)$$

Recall menunjukkan seberapa baik model dalam menemukan semua kasus positif. Ini sangat penting dalam situasi di mana kegagalan untuk mendeteksi kasus positif memiliki konsekuensi yang serius, seperti dalam deteksi penyakit atau sistem keamanan (Bishop, 2006).

2.10 Mean Average Precision

Mean Average Precision (mAP) adalah ukuran yang digunakan untuk menilai kinerja model deteksi objek. mAP adalah rata-rata dari *Average Precision* (AP)



untuk semua kelas dan semua *Intersection over Union (IoU) threshold* yang dipertimbangkan. AP adalah metrik yang mengukur ketepatan model dalam mendeteksi dan mengklasifikasikan objek dengan benar. Untuk menghitung AP, perlu diketahui *precision* dan *recall* terlebih dahulu pada berbagai *threshold confidence*, dan kemudian menghitung area di bawah kurva *precision-recall* (Everingham dkk., 2010).

IoU dihitung sebagai rasio antara area tumpang tindih dan area gabungan dari *bounding box* dan *ground truth*. IoU digunakan untuk menentukan apakah deteksi dianggap benar (*True Positive*) atau salah (*False Negative*). Dalam konteks mAP, berbagai *IoU threshold* digunakan untuk mengevaluasi kinerja model pada berbagai tingkat ketepatan tumpang tindih. mAP dapat dihitung dengan menggunakan persamaan (6).

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad (6)$$

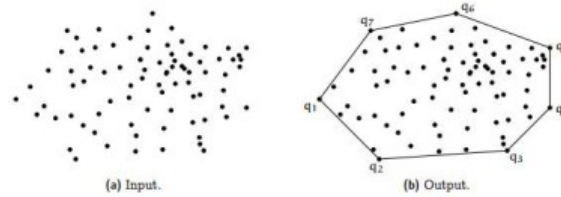
dimana,

N = jumlah kelas

AP_i = Average Precision untuk kelas ke-i

2.11 Convex Hull

Convex hull adalah bentuk poligon terkecil yang dapat menutupi semua titik dalam sebuah kumpulan titik pada bidang dua dimensi. Dalam istilah yang lebih sederhana, *convex hull* adalah bentuk yang paling mirip dengan karet gelang yang diikatkan mengelilingi kumpulan titik tersebut. Poligon ini memiliki sifat bahwa setiap segmen garis yang menghubungkan dua titik dalam poligon tidak akan berada di luar poligon itu sendiri. *Convex hull* sering digunakan dalam berbagai aplikasi seperti pengenalan pola, grafis komputer, dan pemrosesan gambar (Kosanke, 2019). Implementasi *convex hull* pada titik-titik acak terlihat pada Gambar 16.



Gambar 16 Convex hull pada titik-titik acak
 Sumber: (Sikchi, 2017)

Salah satu algoritma yang efisien untuk menghitung *convex hull* adalah algoritma *Andrew's monotone chain*. Algoritma ini bekerja dengan cara mengurutkan titik-titik terlebih dahulu berdasarkan koordinat x (dan koordinat y sebagai *tie-breaker* jika diperlukan). Setelah titik-titik diurutkan, algoritma membangun dua *hull*: *hull* bawah dan *hull* atas. *Hull* bawah dibangun dengan menelusuri titik-titik dari kiri ke kanan, dan *hull* atas dibangun dengan menelusuri titik-titik dari kanan ke kiri (Cormen et al., 2022).

Selama proses ini, orientasi dihitung menggunakan tiga titik untuk menentukan apakah titik tersebut membentuk sudut cekung (*counter-clockwise turn*) atau cembung (*clockwise turn*). Titik-titik yang membentuk sudut cembung akan diabaikan untuk memastikan bahwa hasil akhir adalah poligon *convex*. Orientasi tiga titik P , Q , dan R dihitung dengan persamaan (7).

$$\text{Orientasi} = (Q_y - P_y) \times (R_x - Q_x) - (Q_x - P_x) \times (R_y - Q_y) \quad (7)$$

dimana,

P_x, P_y = Koordinat x dan y dari titik pertama (P),

Q_x, Q_y = Koordinat x dan y dari titik kedua (Q),

R_x, R_y = Koordinat x dan y dari titik ketiga (R)

2.12 Ekstraksi Fitur

Ekstraksi fitur adalah proses mengambil informasi penting atau karakteristik dari data mentah, seperti gambar untuk digunakan dalam analisis atau pemodelan. Dalam konteks pengolahan citra, ekstraksi fitur melibatkan identifikasi dan pengukuran berbagai atribut gambar seperti tepi, tekstur, bentuk, dan warna. Proses



ini bertujuan mengubah data visual menjadi bentuk yang lebih informatif dan berguna bagi pengembangan model (Bishop, 2006).

1. Sumbu Mayor dan Sumbu Minor

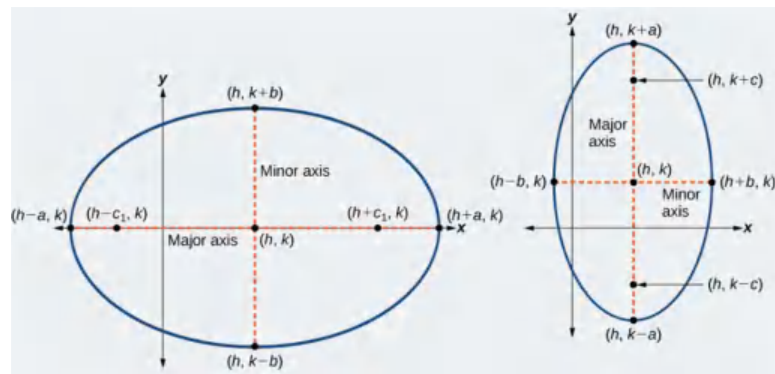
Sumbu mayor dan sumbu minor adalah dua parameter utama dari elips yang paling sesuai dengan *mask* hasil segmentasi. Sumbu mayor adalah sumbu terpanjang dari elips, sedangkan sumbu minor adalah sumbu terpendek (Varoslavskiy, 1977). Persamaan elips umum dalam koordinat kartesian ditunjukkan pada persamaan (8).

$$\frac{(x - h)^2}{a^2} + \frac{(y - k)^2}{b^2} = 1 \quad (8)$$

dimana,

- (h, k) = koordinat pusat elips,
- a = panjang sumbu mayor,
- b = panjang sumbu minor.

Sumbu mayor dan sumbu minor dihitung dengan melakukan *fitting elips* pada area hasil segmentasi. *Fitting elips* akan menyesuaikan elips yang sesuai dengan kontur objek menggunakan *least-squares* (C. Gonzalez & E. Woods, 2018). Gambar 17 menunjukkan elips dengan sumbu mayor dan sumbu minor.



Gambar 17 Elips dengan sumbu mayor dan sumbu minor
Sumber: (Sidekick, 2024)



Kemudian ditentukan *covariance matrix* dari titik-titik kontur elips. *Covariance matrix* akan memberikan informasi tentang distribusi titik-titik kontur dalam ruang dua dimensi dan didefinisikan seperti persamaan (9).

$$C = \begin{pmatrix} \sigma_{xx} & \sigma_{xy} \\ \sigma_{xy} & \sigma_{yy} \end{pmatrix} \quad (9)$$

dimana,

σ_{xx} = variansi dari koordinat x

σ_{xy} = kovarians antara koordinat x dan y

σ_{yy} = variansi dari koordinat y

Setelah *covariance matrix* diperoleh, langkah berikutnya adalah menghitung *eigenvalues* dan *eigenvectors* dari matriks tersebut. *Eigenvalues* (λ_1 dan λ_2) dari *covariance matrix* diperoleh dengan menyelesaikan persamaan (10).

$$\det(C - \lambda I) = 0 \quad (10)$$

Matrix $C - \lambda I$ menggunakan persamaan (11).

$$C - \lambda I = \begin{pmatrix} \sigma_{xx} - \lambda & \sigma_{xy} \\ \sigma_{xy} & \sigma_{yy} - \lambda \end{pmatrix} \quad (11)$$

Dengan menghitung determinan dan menyelesaikan persamaan kuadrat pada persamaan (12).

$$\lambda^2 - (\sigma_{xx} + \sigma_{yy})\lambda + (\sigma_{xx}\sigma_{yy} - \sigma_{xy}^2) = 0 \quad (12)$$

Diperoleh dua *eigenvalue* dimana $\lambda_1 \geq \lambda_2$. *Eigenvalue* berhubungan langsung dengan panjang semi-sumbu mayor (a) dan semi-sumbu minor (b) dari elips, yang dihitung menggunakan persamaan (13) dan persamaan (14).

$$\text{Sumbu mayor} = 2\sqrt{\lambda_1} \quad (13)$$

$$\text{Sumbu minor} = 2\sqrt{\lambda_2} \quad (14)$$



2. Luas

Luas adalah ukuran dari area yang diidentifikasi sebagai objek dalam hasil segmentasi. Proses ini penting dalam berbagai aplikasi pengolahan citra, termasuk analisis video, pengenalan objek, dan pelacakan objek (Bradski & Kaehler, 2009). Setelah mendapatkan panjang sumbu mayor dan sumbu minor, luas dapat dihitung dengan menggunakan pendekatan matematis. Rumus menghitung elips ditunjukkan pada persamaan (15).

$$A = \pi \times a \times b \quad (15)$$

dimana,

A = luas,

a = semi-sumbu mayor,

b = semi-sumbu minor.

3. Keliling

Keliling dari *mask* hasil segmentasi adalah panjang garis yang mengelilingi objek yang terdeteksi. Dalam konteks pengolahan citra digital, keliling ini dapat dihitung dengan berbagai metode tergantung pada bentuk dan kompleksitas objek yang tersegmentasi. Ketika *mask* hasil segmentasi berbentuk menyerupai elips, pendekatan matematis dapat digunakan dengan menggunakan rumus yang didefinisikan oleh Ramanujan, yang memberikan pendekatan akurat untuk panjang keliling elips (Villarino, 2005). Rumus menghitung keliling elips ditunjukkan pada persamaan (16).

$$P \approx \pi[3(a + b) - \sqrt{(3a + b)(a + 3b)}] \quad (16)$$

dimana,

P = keliling,

a = semi-sumbu mayor,

b = semi-sumbu minor.



4. Eksentrisitas

Eksentrisitas adalah ukuran matematis yang menggambarkan seberapa jauh bentuk elips menyimpang dari bentuk lingkaran sempurna. Dalam konteks pengolahan citra, eksentrisitas digunakan untuk menggambarkan bentuk objek yang tersegmentasi, memberikan informasi tentang elongasi objek. Eksentrisitas adalah nilai antara 0 dan 1, di mana 0 menunjukkan lingkaran sempurna dan 1 menunjukkan garis lurus sempurna (Fitzgibbon dkk., 1996).

Eksentrisitas dihitung menggunakan rumus matematis yang melibatkan semi-sumbu mayor dan semi-sumbu minor. Rumus menghitungnya ditunjukkan pada persamaan (17).

$$e = \sqrt{1 - \left(\frac{b}{a}\right)^2} \quad (17)$$

dimana,

e = eccentricity,

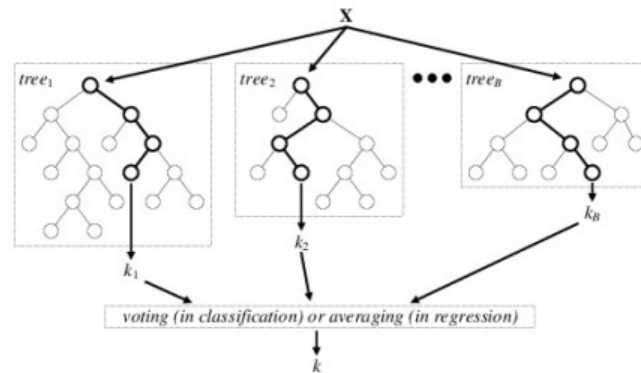
a = semi-sumbu mayor,

b = semi-sumbu minor.

Persamaan ini berasal dari definisi eksentrisitas dalam geometri elips, dimana nilai $\frac{b}{a}$ menunjukkan rasio antara panjang semi-sumbu minor dan semi-sumbu mayor. Semakin besar perbedaan antara a dan b , semakin besar nilai eksentrisitas, yang menunjukkan elips yang lebih lonjong.

2.13 Random Forest Regression

Random Forest Regression adalah algoritma *machine learning ensemble* yang digunakan untuk tugas regresi, yang bertujuan untuk memprediksi nilai kontinu berdasarkan input fitur. Algoritma ini terdiri dari banyak *decision tree* yang digabungkan untuk menghasilkan prediksi yang lebih akurat dan stabil. Setiap *tree* dalam *forest* menghasilkan prediksi yang berbeda, dan hasil akhir adalah rata-rata dari semua prediksi pohon tersebut. Teknik ini memperbaiki masalah *overfitting* yang sering terjadi pada *decision tree* (Breiman, 2001). Arsitektur *random forest* ditunjukkan pada Gambar 18.



Gambar 18 Arsitektur random forest untuk analisis klasifikasi dan regresi
 Sumber: (Suparwito, 2019)

Arsitektur *Random Forest* terdiri dari beberapa komponen utama: *decision tree*, proses *bootstrap sampling*, dan metode agregasi. Setiap *decision tree* dibangun menggunakan subset dari data *training* yang diambil melalui *bootstrap sampling*, yang berarti bahwa setiap sampel data dipilih dengan penggantian. Setiap tree tersebut kemudian dilatih pada subset data yang berbeda, sehingga memperkenalkan variasi dan mengurangi korelasi antara *tree* (Breiman, 2001).

Proses *bootstrap sampling* dan *bagging* (*Bootstrap Aggregating*) adalah teknik kunci dalam *Random Forest*. *Bootstrap sampling* menghasilkan beberapa *subset* dari dataset asli dengan penggantian, memastikan bahwa beberapa sampel mungkin muncul beberapa kali dalam satu *subset* sementara yang lain mungkin tidak muncul sama sekali. *Bagging* kemudian menggabungkan prediksi dari setiap *decision tree* dengan rata-rata (untuk regresi) atau voting mayoritas (untuk klasifikasi), meningkatkan stabilitas dan akurasi model secara keseluruhan (Efron & Tibshirani, 1993).

Setiap *decision tree* dalam *Random Forest* dibangun dengan memilih *subset* acak dari fitur pada setiap *node split*. Proses ini disebut sebagai *random feature selection*, yang mengurangi korelasi antar pohon dan meningkatkan kemampuan generalisasi model. Pohon-pohon tersebut tumbuh hingga kedalaman maksimum atau hingga setiap *node* hanya berisi sejumlah kecil sampel, tanpa pemangkasan seperti pada *decision tree* tradisional (Breiman, 2001).

Setelah semua *decision tree* dalam *forest* dibangun, prediksi akhir untuk regresi diperoleh dengan menggabungkan hasil dari semua *decision tree*. Pada



regresi, agregasi dilakukan dengan menghitung rata-rata prediksi dari setiap pohon. Pendekatan ini mengurangi varian prediksi akhir, menghasilkan model yang lebih stabil dan akurat dibandingkan dengan *decision tree* tunggal (Breiman, 2001; Hastie dkk., 2009).

Salah satu kelebihan utama *Random Forest Regression* adalah kemampuannya untuk menangani dataset dengan banyak fitur dan sampel tanpa risiko *overfitting* yang signifikan. Algoritma ini juga memiliki kemampuan bawaan untuk mengukur pentingnya fitur, yang membantu dalam proses seleksi fitur dan interpretasi model. Selain itu, *Random Forest Regression* relatif mudah untuk diimplementasikan dan di-*tune*, membuatnya menjadi pilihan populer untuk banyak aplikasi *machine learning* (Breiman, 2001).

2.14 Uji *Mean Decrease in Accuracy*

Mean Decrease in Accuracy (MDA) adalah metode untuk mengevaluasi pentingnya fitur dalam *machine learning*, khususnya dalam konteks *Random Forest Regression*. MDA mengukur kontribusi setiap fitur terhadap kinerja model dengan cara menghitung penurunan akurasi model ketika nilai fitur tersebut secara acak dipermutasi. Pendekatan ini memberikan wawasan tentang seberapa kritis setiap fitur dalam menentukan prediksi model (Louppe, 2014).

Untuk setiap fitur, nilai-nilai tersebut secara acak dipermutasi di seluruh sampel data. Ini memutus hubungan antara fitur dan target, sehingga model tidak dapat menggunakan informasi dari fitur tersebut. Penurunan akurasi dihitung sebagai selisih antara akurasi awal dan akurasi setelah permutasi (Louppe, 2014). Secara matematis, MDA untuk fitur i , dapat dihitung seperti pada persamaan (18).

$$MDA_i = \frac{1}{n} \sum_{j=1}^n (ACC_{original} - ACC_{Permuted_i}^{(j)}) \quad (18)$$

dimana,

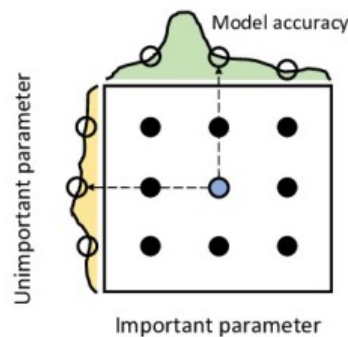
- n = jumlah uji permutasi,
- $ACC_{original}$ = akurasi model dengan dataset asli,
- $ACC_{Permuted_i}^{(j)}$ = akurasi model dengan fitur i yang dipermutasi.



2.15 Grid Search

Grid search adalah teknik pencarian *hyperparameter* yang sistematis dalam *machine learning* yang bertujuan untuk menemukan kombinasi parameter yang memberikan kinerja terbaik untuk model. Teknik ini melibatkan evaluasi model menggunakan berbagai kombinasi parameter yang telah ditentukan sebelumnya dalam ruang pencarian yang *grid-like*.

Untuk menerapkan *Grid Search*, dimulai dengan mendefinisikan ruang *hyperparameter* yang ingin dioptimalkan. kemudian membuat *grid* dari semua kombinasi nilai *hyperparameter*. Setiap kombinasi *hyperparameter* dievaluasi menggunakan *cross-validation* dan kombinasi yang memberikan kinerja terbaik dipilih sebagai konfigurasi optimal untuk model (Bishop, 2006). Ilustrasi implementasi grid search ditunjukkan pada Gambar 19.



Gambar 19 Implementasi Grid Search
Sumber: (Pilario dkk., 2021)

Grid search memungkinkan semua kombinasi *hyperparameter* yang mungkin diuji secara lengkap, memberikan jaminan bahwa kombinasi terbaik ditemukan. Meskipun metode ini bisa memberikan akurasi dalam mencari *hyperparameter* optimal, metode ini sangat memakan waktu dan sumber daya komputasi, terutama dengan dataset besar atau model dengan banyak *hyperparameter* (Bishop, 2006).

2.16 Root Mean Square Error

Root Mean Square Error (RMSE) adalah ukuran statistik yang sering digunakan untuk menilai akurasi model prediksi. RMSE mengukur rata-rata besarnya kesalahan antara nilai yang diprediksi oleh model dan nilai sebenarnya



dalam dataset. Dalam konteks ini, error merujuk pada selisih antara nilai yang diprediksi dan nilai yang sebenarnya. RMSE sangat berguna dalam menilai kinerja model karena memberikan indikasi seberapa jauh prediksi dari model menyimpang dari nilai sebenarnya. Nilai RMSE yang lebih rendah menunjukkan model yang lebih akurat dan lebih baik dalam melakukan prediksi (Chai & Draxler, 2014).

Untuk menghitung RMSE, pertama-tama kita menghitung selisih antara setiap nilai yang diprediksi \hat{y}_i dan nilai sebenarnya y_i . Selisih ini kemudian dikuadratkan untuk menghilangkan nilai negatif dan memperbesar kesalahan yang lebih besar. Setelah itu, kita mengambil rata-rata dari semua selisih kuadrat ini dan akhirnya mengambil akar kuadrat dari nilai rata-rata tersebut. Rumus matematis menghitung RMSE seperti pada persamaan (19).

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (19)$$

dimana,

y_i = nilai sebenarnya,

\hat{y}_i = nilai prediksi,

n = jumlah data.

2.17 Mean Absolute Error

Mean Absolute Error (MAE) adalah ukuran statistik yang digunakan untuk menilai akurasi model prediksi. MAE mengukur rata-rata besarnya kesalahan absolut antara nilai yang diprediksi oleh model dan nilai sebenarnya dalam dataset. Kesalahan absolut disini berarti selisih antara nilai prediksi dan nilai sebenarnya, diabaikan tanda negatifnya. MAE memberikan gambaran langsung mengenai seberapa jauh prediksi model dari nilai sebenarnya dalam satuan yang sama dengan data aslinya. Nilai MAE yang lebih rendah menunjukkan model prediksi yang lebih akurat (Chai & Draxler, 2014).

Untuk menghitung MAE, kita terlebih dahulu menghitung selisih absolut antara nilai yang diprediksi yang diprediksi \hat{y}_i dan nilai sebenarnya y_i . Kemudian kita mengambil rata-rata dari semua selisih absolut ini. Rumus matematis menghitung MAE seperti pada persamaan (20).



$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

(20)

dimana,

y_i = nilai aktual,

\hat{y}_i = nilai prediksi,

n = jumlah data.