

**ESTIMASI BIOMASSA TANAMAN NANAS BERDASARKAN LEBAR
DAUN MENGGUNAKAN ALGORITMA *MACHINE LEARNING***

(Skripsi)

Oleh

FANI KOSASIH



**JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG
2023**

ABSTRACT

ESTIMATION OF PINEAPPLE PLANTS BIOMASS BASED ON LEAF WIDTH USING MACHINE LEARNING ALGORITHM

By

FANI KOSASIH

A pineapple plant has a biomass value, the most critical indicator to determine whether a plant can produce politely. The problem fronted by pineapple plantation companies is that it is challenging to decide on the right time to bear out the flowering stage. That action is with forcing based on the age of the plant at 12-14 months. However, forcing on pineapple plants is considered more effective in terms of plant weight, which is a 2,3 – 2,5 kg range leaf width of 6–7 cm. This study aims to create a system that can estimate pineapple plant biomass based on leaf width by utilizing technology artificial intelligence using data from aerial photographs of pineapple farmsteads. The method developed consists of the YOLOv5 method, image processing for feature extraction of the number of pixels from objects in the top view, and linear regression to predict the width of leaves on objects. Test results using k-fold cross-validation show the average value of R-squared and MAPE at k=5 is 85% and 0.94% and at k=10 is 81% and 0.95%. The predicted results of the linear regression resulted in 180 plants with a leaf width of 6-7 cm from a total of 216 plants with the predicted data. Based on the results obtained, it can be concluded that the designed algorithm already has good accuracy.

Key Words : Biomass, Image Processing, YOLOv5, Linear Regression

ABSTRAK

ESTIMASI BIOMASSA TANAMAN NANAS BERDASARKAN LEBAR DAUN MENGGUNAKAN ALGORITMA MACHINE LEARNING

Oleh

FANI KOSASIH

Sebuah tanaman nanas memiliki nilai biomassa yang menjadi indikator paling kritis menentukan apakah suatu tanaman dapat berproduksi dengan baik. Masalah yang dihadapi oleh perusahaan perkebunan nanas adalah sulit untuk menentukan kapan waktu yang tepat untuk melakukan tahapan pembungaan. Upaya yang adalah dengan *forcing* berdasarkan usia tanaman saat 12-14 bulan. Namun *forcing* pada tanaman nanas dinilai lebih efektif jika melihat dari segi berat tanaman yaitu 2,3 – 2,5 kg yang memiliki *range* lebar daun 6–7 cm. Penelitian ini bertujuan untuk membuat sistem yang dapat mengestimasi biomassa tanaman nanas berdasarkan lebar daun dengan memanfaatkan teknologi *artificial intelligence* menggunakan data dari foto udara lahan perkebunan nanas. Metode yang dibangun terdiri dari metode YOLOv5, pengolahan citra untuk ekstraksi fitur jumlah piksel dari objek pada posisi tampak atas dan regresi linier untuk memprediksi lebar daun pada objek. Hasil pengujian menggunakan *k-fold cross validation* menunjukkan nilai rata *R-squared* dan MAPE pada k=5 adalah 85% dan 0.94% dan pada k=10 adalah 81% dan 0.95%. Hasil prediksi regresi linier menghasilkan 180 tanaman dengan lebar daun 6-7 cm dari total 216 tanaman data hasil prediksi. Berdasarkan hasil yang diperoleh dapat disimpulkan bahwa algoritma yang dibangun sudah memiliki akurasi yang baik.

Kata Kunci: Biomassa, Pengolahan Citra, YOLOv5, Regresi Linier

**ESTIMASI BIOMASSA TANAMAN NANAS BERDASARKAN LEBAR
DAUN MENGGUNAKAN ALGORITMA *MACHINE LEARNING***

Oleh

FANI KOSASIH

Skripsi

**Sebagai salah satu syarat untuk mendapat gelar
SARJANA TEKNIK**

Pada

**Jurusan Teknik Elektro
Fakultas Teknik
Universitas Lampung**

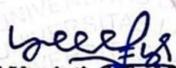


**FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG
2023**

Judul : ESTIMASI BIOMASSA TANAMAN NANAS
BERDASARKAN LEBAR DAUN MENGGUNAKAN
ALGORITMA *MACHINE LEARNING*

Nama Mahasiswa : Fani Kosasih
NPM : 1815031067
Program Studi : S1 Teknik Elektro
Jurusan : Teknik Elektro
Fakultas : Teknik




Yetti Yuniati, S.T., M.T.
NIP. 19800113 200912 2 002


Dr. Ing. Ardian Ulvan, S.T., M.Sc.
NIP. 19731128 199903 1 005

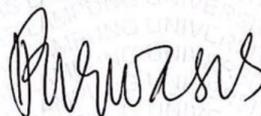
2. Mengetahui

**Ketua Jurusan
Teknik Elektro**



Heritnawati, S.T., M.T.
NIP. 19710314 199903 2 001

**Ketua Program Studi
Teknik Elektro**

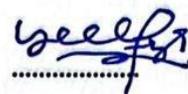


Dr. Eng. Nining Purwasih, S.T., M.T.
NIP. 19740422 200012 2 001

MENGESAHKAN

1. Tim Penguji

Ketua : Yetti Yuniati, S.T., M.T.



Sekretaris : Dr. Ing. Ardian Ulvan, S.T., M.Sc.



Penguji : Dr. Misfa Susanto, S.T., M.Sc.



2. Dekan Fakultas Teknik



Dr. Eng. Ir. Helmy Fitriawan, S.T., M.Sc.
NIP. 19750928 200112 1 002

Tanggal Lulus Ujian Skripsi: 24 Januari 2023

SURAT PERNYATAAN

Saya yang bertanda tangan di bawah ini, menyatakan bahwa skripsi saya yang berjudul "Estimasi Biomassa Tanaman Nanas Berdasarkan Lebar Daun Menggunakan Algoritma *Machine Learning*" merupakan karya saya sendiri dan bukan karya orang lain. Semua tulisan yang tertuang dalam skripsi ini telah mengikuti kaidah penulisan karya ilmiah Universitas Lampung. Apabila di kemudian hari terbukti skripsi saya merupakan hasil penjiplakan atau dibuat orang lain, maka bersedia menerima sanksi sesuai dengan hukum yang berlaku.

Bandar Lampung, 31 Januari 2023



Fani Kosasih
NPM. 1815031067

RIWAYAT HIDUP



Penulis dilahirkan di Sumber Agung, pada tanggal 05 Maret 2001. Penulis merupakan anak pertama dari tiga bersaudara dari pasangan Bapak Mat Sopan dan Ibu Jumiati. Penulis memulai pendidikan di MI MA Sumber Agung pada tahun 2006 hingga 2012, MTsN 1 Pesawaran pada tahun 2012 hingga 2015, dan MAN 1 Pesawaran pada tahun 2015 hingga 2018. Penulis diterima di Program Studi Teknik Elektro Universitas Lampung pada tahun 2018 melalui jalur Seleksi Bersama Masuk Perguruan Tinggi Negeri (SBMPTN).

Selama menjadi mahasiswa, Penulis aktif di organisasi Himpunan Mahasiswa Teknik Elektro Universitas Lampung (HIMATRO UNILA) sebagai Anggota Departemen Komunikasi dan Media Informasi Divisi Media Informasi pada periode 2019 dan periode 2020. Penulis aktif sebagai Asisten Laboratorium Pengukuran Besaran Listrik Jurusan Teknik Elektro Universitas Lampung pada tahun 2020 sampai tahun 2022. Penulis berkesempatan menjadi asisten praktikum mata kuliah Pengukuran Besaran Listrik, Fisika Dasar TI, dan Rangkaian Listrik. Penulis melaksanakan kerja praktik (KP) di PT. Telekomunikasi Indonesia Tbk. Divisi Infrastruktur Telekomunikasi Area *Network* Lampung pada 16 Agustus 2021 sampai 14 September 2021 dengan Menyusun laporan yang berjudul “Analisis dan Monitoring pada Jaringan *Metro Ethernet* Teluk Betung Pt. Telekomunikasi Indonesia Tbk. Lampung”. Pada tanggal 14 Februari 2022 sampai 31 Juli 2022, penulis mengikuti program kampus merdeka studi independen Bisnis Kecerdasan Artifisial (AI Hustler) di PT. Bisa Artifisial Indonesia.

PERSEMBAHAN

Kupersembahkan karya sederhana ini dengan rasa hormat, cinta dan kasih sayang kepada:

Allah SWT, sebagai ucapan rasa syukur kepada-Nya yang telah memberi kemudahan untuk menyusun skripsi ini. Setiap goresan tinta ini adalah wujud dari keagungan dan kasih sayang yang diberikan Allah Swt kepada hamba-Nya.

Kedua orang tua tercinta, Bapak Mat Sopan who tried his very best not to be his worst. Dan Ibu Jumiati, who made sure we never saw him at his worst.

Adik-adikku tersayang Nada Nadidah dan Fathin Muayyad.

Keluarga besar, Dosen, Teman, serta Almamater yang telah menjadi motivasi dan inspirasi, terima kasih atas kekeluargaan, do'a, serta dukungan yang selalu mengiringi hingga merasa lebih baik dari hari ke hari.

MOTTO

“So remember Me, I will remember you. And be grateful to Me and do not deny Me” — Q.S Al Baqarah : 152

“Believers! Seek help in patience and in as-Salaat (Prayer). Truly, Allah is with As-Sabiroon (the patient)” — Q.S Al Baqarah : 153

“Barangsiapa yang hendak menginginkan dunia, maka hendaklah ia menguasai ilmu. Barangsiapa menginginkan akhirat hendaklah ia menguasai ilmu, dan barangsiapa yang menginginkan keduanya (dunia dan akhirat) hendaklah ia menguasai ilmu,” — HR Ahmad

“Kau akan berhasil dalam setiap pelajaran, dan kau percaya akan berhasil, dan berhasil lah kau. Anggap semua pelajaran mudah, dan semua akan menjadi mudah. Jangan takut pada pelajaran apapun, karena ketakutan itu kebodohan awal yang akan membodohkan semua.” — Pramoedya Ananta Toer

“Jangan pernah meremehkan diri sendiri, karena diri sendiri ada kelebihan tersendiri” — Portgas D. Ace, One Piece

“Education never ends. It is a series of lessons, with the greatest for the last” — Sir Arthur Conan Doyle, Sherlock Holmes: His Last Bow

“People say parenting is the hardest job in the world – they’re wrong – growing up is. We all just forget how hard it was” — Draco Malfoy, Harry Potter and the Cursed Child

“And, when you want something, all the universe conspires in helping you to achieve it.” — Paulo Coelho, The Alchemist

“If you can’t survive, just try” — The 1975, I Always Wanna Die (Sometimes)

“CARPE DIEM — enjoy life while you can” — Horace, The Odes

SANWACANA

Segala puji bagi Allah Swt, atas limpahan nikmat-Nya yang diberikan kepada penulis sehingga dapat menyelesaikan Tugas Akhir ini. Shalawat dan salam senantiasa dicurahkan kepada Nabi Muhammad saw. suri teladan yang mampu membuka sesuatu yang terkunci, penutup dari semua yang terdahulu, penolong kebenaran dengan jalan yang benar, dan petunjuk kepada jalan-Mu yang lurus.

Skripsi dengan judul “Estimasi Biomassa Tanaman Nanas Berdasarkan Lebar Daun Menggunakan Algoritma *Machine Learning*” yang merupakan salah satu syarat untuk memperoleh gelar Sarjana Teknik pada Jurusan Teknik Elektro, Fakultas Teknik, Universitas Lampung. Dalam penulisan skripsi ini, penulis mendapatkan bantuan dan dukungan dari berbagai pihak baik secara langsung maupun tidak langsung. Oleh karena itu pada kesempatan ini menyampaikan rasa terima kasih kepada:

1. Ibu Prof. Dr. Ir. Lusmeilia Afriani, D.E.A., IPM., selaku Rektor Universitas Lampung.
2. Bapak Dr. Eng., Helmy Fitriawan, S.T., M.Sc. selaku Dekan Fakultas Teknik Universitas Lampung.
3. Ibu Herlinawati, S.T., M.T. selaku Ketua Jurusan Teknik Elektro Universitas Lampung.
4. Bapak Ir. Meizano Ardhi Muhammad, S.T., M.T. selaku Sekretaris Jurusan Teknik Elektro Universitas Lampung.
5. Ibu Dr. Eng. Nining Purwasih, S.T., M.T. selaku Ketua Program Studi Teknik Elektro Universitas Lampung.
6. Ibu Yetti Yuniati S.T., M.T. selaku Dosen Pembimbing Utama serta Kepala Laboratorium Pengukuran Besaran Listrik Jurusan Teknik Elektro yang telah memberikan ilmu, bimbingan, bantuan, arahan, masukan, motivasi, dan

pandangan kehidupan kepada penulis disetiap kesempatan dengan baik dan ramah.

7. Bapak Dr. Ing. Ardian Ulvan, S.T., M.Sc. selaku Dosen Pembimbing Pendamping yang telah memberikan ilmu, bimbingan, bantuan, arahan, masukan, motivasi, dan pandangan kehidupan kepada penulis disetiap kesempatan dengan baik dan ramah.
8. Bapak Dr. Misfa Susanto, S.T., M.Sc. selaku Dosen Penguji yang telah memberikan kritik dan saran yang membangun kepada penulis.
9. Ibu Dr. Eng. Ir. Dikpride Despa, S.T., M.T., IPM., ASEAN Eng. selaku pembimbing akademik dari semester 1 hingga sampai sekarang yang telah memberikan arahan, nasehat, dan bimbingan yang membangun bagi penulis dalam masa perkuliahan.
10. Seluruh dosen di Jurusan Teknik Elektro yang telah memberikan ilmu yang bermanfaat kepada penulis selama menempuh pendidikan perkuliahan.
11. Seluruh staff di Jurusan Teknik Elektro yang telah membantu penulis baik dalam hal administrasi dan lain-lain.
12. Teristimewa kedua Orang Tua tercinta yang telah memberikan dukungan, doa, moril, dan materil kepada penulis.
13. Adik-adikku yang telah memberikan doa dan semangat selama penulis menyelesaikan skripsi.
14. Kaira, Eni, Natasyah, Fara, dan Raffi yang telah membantu dan memberikan doa dan dukungan kepada penulis.
15. Segenap Keluarga Besar Laboratorium Pengukuran Besaran Listrik yang telah memberikan dukungan kepada penulis.
16. Rekan-rekan Teknik Elektro dan Teknik Informatika Universitas Lampung Angkatan 2018 yang telah memberikan dukungan moril kepada penulis.
17. Keluarga Besar HIMATRO yang telah memberikan ilmu dan pengalaman kepada penulis selama pendidikan baik secara langsung maupun tak langsung.
18. Semua pihak yang terlibat dalam menyelesaikan Skripsi ini yang tidak dapat penulis sebutkan satu persatu.

19. Last, I wanna thank me for every hard work that I give, thank me for never quitting, thank me for believing God's timing is always perfect, thank me for having no days off, thank me for the tears that I hold myself and I wanna thank me for being me at all times. You did well Fan!!

Penulis menyadari bahwa masih banyak kekurangan dalam penulisan skripsi ini. Penulis mengharapkan adanya kritik dan saran yang membangun dari berbagai pihak demi kemajuan bersama. Semoga skripsi ini dapat bermanfaat bagi penulis dan pembaca.

Bandar Lampung, 31 Januari 2023

Penulis



Fani Kosasih

DAFTAR ISI

	Halaman
ABSTRACT	i
ABSTRAK	ii
HALAMAN JUDUL	iii
LEMBAR PERSETUJUAN	iv
LEMBAR PENGESAHAN	v
SURAT PERNYATAAN	vi
RIWAYAT HIDUP	vii
PERSEMBAHAN	viii
MOTTO	ix
SANWACANA	x
DAFTAR ISI	xiii
DAFTAR GAMBAR	xvi
DAFTAR TABEL	xviii
I. PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	4
1.3 Tujuan.....	4
1.4 Batasan Masalah	5
1.5 Manfaat.....	5
1.6 Sistematika Penulisan.....	6
II. TINJAUAN PUSTAKA	7
2.1 Penelitian Terkait	7
2.2 Estimasi Biomassa	13

2.3	Nanas	14
2.4	<i>Forcing</i>	16
2.5	<i>Artificial Intelligence</i>	16
2.6	<i>Machine Learning</i>	17
2.7	<i>Deep Learning</i>	18
2.8	<i>Convolutional Neural Network (CNN)</i>	20
2.9	<i>Computer Vision</i>	21
2.10	<i>Object Detection</i>	22
2.11	<i>You Only Look Once (YOLO)</i>	23
	2.11.1 Deskripsi YOLO v5	26
	2.11.2 Arsitektur YOLO v5	27
	2.11.3 Tipe YOLOv5.....	29
2.12	Regresi.....	30
	2.12.1. Regresi Linier	31
2.13	Pengelolaan Citra Digital.....	31
2.14	Citra.....	32
2.15	Citra Digital	33
	2.15.1 Citra Biner	34
	2.15.2 Citra <i>Grayscale</i>	35
	2.15.3 Citra RGB (<i>Red Green Blue</i>).....	36
2.16	Segmentasi.....	37
2.17	<i>Thresholding</i>	37
2.18	Python.....	39
2.19	Roboflow	41
2.20	Google Colaboratory	42
2.21	Jupyter Notebook	42
2.22	Populasi dan Sampel	43
2.23	<i>Confusion Matrix</i>	43
2.24	<i>K-fold Cross Validation</i>	45
III.	METODELOGI PENELITIAN	49
3.1	Tempat dan Waktu Penelitian.....	49
3.2	Alat dan Bahan.....	49

3.3	Prosedur Penelitian.....	49
3.4	Perancangan Program.....	51
3.4.1	Metode YOLOv5	51
3.4.2	Pengolahan Citra.....	57
3.4.3	Populasi dan Sampel	59
3.4.4	Regresi Linier	60
3.5	Analisis Data.....	61
IV.	HASIL DAN PEMBAHASAN	62
4.1	Dataset	62
4.2	Deteksi dengan Algoritma <i>You Only Look Once</i> (YOLO).....	63
4.3	Populasi dan Sampel	73
4.4	Pengolahan Citra	74
4.5	Regresi Linier.....	78
4.6	Hasil Pengujian dan Analisa Regresi Linear	82
4.7	Konversi Lebar Daun Terprediksi ke Biomassa	87
V.	KESIMPULAN DAN SARAN	89
	DAFTAR PUSTAKA	91

DAFTAR GAMBAR

	Halaman
Gambar 2.1 Morfologi tanaman nanas	14
Gambar 2. 2 Hubungan antara <i>artificial intelligence</i> , <i>machine learning</i> dan <i>deep learning</i>	19
Gambar 2. 3 <i>Convolution Neural Network</i> (CNN)	20
Gambar 2. 4 Tugas dalam <i>computer vision</i>	22
Gambar 2. 5 Proses deteksi pada YOLO	24
Gambar 2. 6 Arsitektur YOLOv5	28
Gambar 2. 7 Peforma tipe model YOLOv5	29
Gambar 2. 8 Logo Python	39
Gambar 2. 9 Roboflow	41
Gambar 2. 10 Ilustrasi <i>K-fold Cross Validation</i>	46
Gambar 3. 1 Diagram alir <i>AI Project Cycle</i> untuk metode YOLOv5	51
Gambar 3. 2 Dataset lahan perkebunan nanas.....	53
Gambar 3. 3 Diagram alir <i>data exploration</i>	53
Gambar 3. 4 Labeling objek nanas menggunakan software Roboflow	54
Gambar 3. 5 Rincian <i>preprocessing</i> data dan augmentasi data.....	55
Gambar 3. 6 Alur proses tahap modelling	56
Gambar 3. 7 Diagram alir pengolahan citra	57
Gambar 3. 8 Memilih citra yang akan dihitung lebar daunnya	58
Gambar 3. 9 <i>Cropping</i> pada citra biner.....	59
Gambar 3. 10 Diagram alir regresi linier dan konversi lebar daun ke biomassa ..	60
Gambar 4. 1 Dataset lahan tanaman nanas.....	62
Gambar 4. 2 Jumlah dataset dan jenis <i>preprocessing</i> dan augmentasi yang digunakan.....	67

Gambar 4. 3 Inisiasi awal menghubungkan <i>respository</i> YOLO v5 dengan Google Colab.....	68
Gambar 4. 4 Ekstrak dataset dari Roboflow	69
Gambar 4. 5 Konfigurasi <i>training</i> dataset.....	69
Gambar 4. 6 <i>Heatmap confusion matrix</i> untuk model.....	70
Gambar 4. 7 <i>Running script</i> hasil deteksi	72
Gambar 4. 8 <i>Bounding image</i> hasil deteksi dataset nanas1	72
Gambar 4. 9 <i>Running script</i> citra <i>grayscale</i>	74
Gambar 4. 10 <i>Running script</i> citra <i>thresholding</i>	75
Gambar 4. 11 <i>Running script</i> dan hasil perhitungan jumlah elemen piksel.....	76
Gambar 4. 12 <i>Output</i> pada citra biner.....	77
Gambar 4. 13 <i>Import library</i> dan <i>load data</i>	79
Gambar 4. 14 Hasil pemodelan regresi linier	80
Gambar 4. 15 Garis regresi hubungan X dengan Y	82
Gambar 4. 16 Grafik hasil pengujian <i>R-squared</i> K=5.....	84
Gambar 4. 17 Grafik hasil pengujian MAPE K=5.....	84
Gambar 4. 18 Grafik hasil Pengujian <i>R-squared</i> K=10	86
Gambar 4. 19 Grafik hasil Pengujian MAPE K=10	86

DAFTAR TABEL

	Halaman
Tabel 2. 1 Penelitian terkait.....	11
Tabel 2. 2 <i>Machine Learning: Supervised, Unsupervised dan Reinforcement</i>	18
Tabel 2. 3 Peforma tipe model YOLOv5	30
Tabel 2. 4 Warna dan nilai penyusun warna	36
Tabel 2. 5 <i>Confusion matrix</i>	44
Tabel 2. 6 <i>Range</i> nilai <i>R-squared</i>	47
Tabel 2. 7 <i>Range</i> nilai MAPE	48
Tabel 3. 1 <i>Problem Scoping</i>	52
Tabel 4. 1 Informasi citra satelit	63
Tabel 4. 2 Jumlah dataset yang digunakan.....	65
Tabel 4. 3 Hasil akurasi dari <i>training</i> kinerja YOLO v5	71
Tabel 4. 4 Perbandingan jumlah tanaman nanas data tes secara <i>labelling</i> , jumlah tanaman hasil deteksi dengan algoritma YOLOv5 dan akurasinya.....	73
Tabel 4. 5 Hasil dari perhitungan piksel putih pada citra biner	78
Tabel 4. 6 Data regresi linier sederhana.....	79
Tabel 4. 7 Hasil Pengujian <i>5-fold cross validation</i>	83
Tabel 4. 8 Hasil pengujian <i>10-fold cross validation</i>	85
Tabel 4. 9 Lebar daun hasil prediksi menggunakan regresi linier	87

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Indonesia adalah negara dengan penghasil nanas terbesar ketiga di Asia Tenggara setelah Filipina dan Thailand dengan nilai kontribusi pasar sebesar 23% [1]. Tanaman nanas (*Ananas comosus*) tersebar dan tumbuh baik di Indonesia. Tanaman ini berasal dari Amerika Selatan dengan daerah yang beriklim tropik. Hampir di seluruh wilayah Indonesia merupakan daerah penghasil nanas karena didukung iklim tropis yang sesuai. Produksi nanas Indonesia di tahun 2018 menempati posisi ke-4 produksi tanaman buah-buahan dengan total produksi mencapai 1.795.986 ton. Provinsi dengan produksi nanas terbesar adalah provinsi Lampung dengan total mencapai 622.881 ton, diikuti oleh Jawa Tengah (202.823 ton), Jawa Barat (180.802 ton), Sumatera Utara (145.618 ton), Jawa Timur (139.234 ton), dan provinsi-provinsi lainnya menghasilkan 514.148 ton nanas [2].

PT Great Giant Pineapple adalah perusahaan yang bergerak dalam bidang agroindustri yaitu penanaman nanas. Jenis nanas yang ditanam adalah '*Smooth Cayenne*' (nanas tanpa duri) dengan jenis bibit *sucker* dan kelas bibit GP-3 atau klon nanas ukuran yang lebih besar, biasanya varietas ini digunakan sebagai buah nanas kalengan. Nanas telah menjadi komoditas buah unggulan dengan volume ekspor paling tinggi di Indonesia. Badan Pusat Statistik (BPS) mencatat, volume ekspor nanas mencapai 236.226 ton pada 2019, meningkat sebesar 7.693 ton dari tahun 2018 [3]. Menurut Dinas Ketahanan Pangan, Tanaman Pangan dan Hortikultura Provinsi Lampung PT. Great Giant Pineapple sejak dua tahun terakhir menjadi perusahaan pengeksport nanas terbesar di dunia. Dengan lebih

dari 13.000 kontainer nanas pertahun, PT. Great Giant Pineapple mampu melampaui ekspor nanas dari Thailand dan beberapa negara penghasil nanas lainnya [4].

Sebuah tanaman nanas memiliki nilai biomassa tanaman. Biomassa merupakan salah satu indikator yang paling kritis menentukan apakah suatu tanaman dapat berproduksi dengan baik [5]. Biomassa adalah produk fotosintesis yang menyerap energi sinar matahari dan mengubah karbon dioksida dengan air menjadi campuran karbon, hidrogen, dan oksigen. Pendekatan yang dipakai untuk mengukur berat biomassa tanaman adalah berat tanaman basah dan tanaman kering. Tanaman nanas adalah tanaman *annual* dengan fase pertumbuhan yang panjang yakni sekitar 12–18 bulan tergantung jenis bibit yang digunakan.

Salah satu masalah yang dihadapi oleh PT Great Giant Pineapple adalah membutuhkan waktu yang cukup lama dalam tahap pembungaan dan keseragaman pembungaan. Tanaman nanas secara alamiah akan berbunga dengan sendirinya bila telah matang secara fisiologis, tetapi tidak serempaknya pertumbuhan bunga dapat menimbulkan masalah pada tinggi rendahnya produksi. Salah satu upaya yang dilakukan adalah dengan cara *forcing*. *Forcing* adalah teknik perangsangan pembungaan demi menyeragamkan perubahan pertumbuhan dari vegetatif ke generatif. Kandungan kimia yang digunakan mengandung *etilen* atau *asetilen* yang secara alamiah diproduksi tanaman nanas sebagai hormon di tubuh tanaman.

Selama ini PT Great Giant Pineapple menggunakan cara *forcing* berdasarkan usia tanaman saat 12-14 bulan. Cara *forcing* pada tanaman nanas dinilai lebih efektif jika menilai dari segi berat tanaman, agar mendapatkan waktu *forcing* yang tepat. Berat tanaman nanas yang baik untuk *forcing* adalah 2,3 - 2,5 kg, dimana berat tanaman nanas tersebut memiliki *range* lebar daun 6 – 7 cm [6]. Oleh karena itu, diperlukan sistem untuk dapat mengestimasi berat tanaman nanas yang dilihat dari lebar daun guna mengefisiensi waktu panen.

Sekarang ini dunia berada di era digital. Era dimana hampir setiap aspek dalam kehidupan manusia sangat berhubungan erat dengan teknologi komputasi. Semakin berkembangnya zaman, manusia terus mengembangkan pengetahuan dan teknologi guna membantu dan meringankan pekerjaannya. Salah satunya pada bidang penelitian yang hingga saat ini berkembang adalah kecerdasan buatan atau lebih dikenal dengan sebutan *Artificial Intelligence* (AI).

Pengembangan cabang ilmu pengetahuan *artificial intelligence*, salah satunya adalah *computer vision*. Dalam *computer vision* terdapat beberapa permasalahan diantaranya adalah *object detection* dan *image classification*. Terdapat beberapa metode dalam mendeteksi dan mengenali objek pada sebuah gambar, salah satunya adalah metode *Convolutional Neural Network* (CNN). CNN merupakan metode yang ada di *deep learning*, memiliki fungsi menyelesaikan permasalahan yang berkaitan dengan *object detection* dan *image classification*. Metode *You Only Look Once* (YOLO) merupakan salah satu metode yang cepat dan akurat dalam melakukan pendeteksian objek. Metode ini dapat melakukan deteksi objek hingga 2 kali lebih cepat daripada algoritma yang lain. Metode ini pernah diimplementasikan oleh Iqbal Nurhabib [7] untuk program penghitungan pengenalan objek kelapa sawit dengan akurasi sebesar 85,4%.

Teknologi lain yang berkembang pada cabang ilmu pengetahuan *artificial intelligence* adalah *machine learning* yang merupakan suatu algoritma atau program komputer yang dapat membuat sistem menjadi cerdas dengan mempelajari data-data yang tersedia di mana algoritma atau program tersebut tidak didefinisikan secara eksplisit [8]. Dalam perkembangannya, *machine learning* dijalankan dengan tiga metode salah satunya merupakan *supervised learning*. Regresi merupakan metode pada *supervised learning* yang mampu untuk melihat hubungan yang terdiri pada satu atau lebih variabel bebas yang ada. Metode regresi ini pernah dilakukan oleh Hurriyatul Fitriyah [9] guna memprediksi panjang dan berat objek menggunakan *computer vision* dan *machine learning*.

Berdasarkan uraian di atas, pada penelitian ini akan dibuatlah sebuah sistem untuk mengestimasi biomassa tanaman nanas berdasarkan lebar daunnya dengan cara mendeteksi tanaman nanas, menghitung jumlah tanaman, dan mengukur lebar daun tanaman nanas dengan foto udara lahan perkebunan nanas yang diambil menggunakan UAV, lalu mengkonversi lebar tersebut ke dalam berat tanaman nanas. Adapun metode yang dipakai pada sistem adalah YOLOv5, pengolahan citra dan regresi linier. Oleh karena itu, peneliti membuat penelitian yang berjudul “Estimasi Biomassa Tanaman Nanas Berdasarkan Lebar Daun Menggunakan Algoritma *Machine learning*” dengan studi kasus pada tanaman nanas di PT Great Giant Pineapple. Harapan dari penelitian ini yaitu mampu untuk mengestimasi biomassa nanas dengan akurasi baik sehingga nantinya informasi tersebut dapat berguna bagi pihak yang membutuhkannya.

1.2 Rumusan Masalah

Rumusan masalah pada penelitian yang dibahas ialah sebagai berikut:

1. Bagaimana mendeteksi tanaman nanas berdasarkan citra foto udara menggunakan *You Only Look Once* (YOLO).
2. Bagaimana menghitung jumlah tanaman nanas berdasarkan citra foto udara menggunakan *You Only Look Once* (YOLO).
3. Bagaimana menghitung lebar daun tanaman nanas dari piksel ke centimeter menggunakan metode regresi.
4. Bagaimana Mengkonversi lebar daun terprediksi hasil regresi linier ke biomassa tanaman nanas.

1.3 Tujuan

Tujuan dari penelitian penelitian ini adalah sebagai berikut:

1. Melakukan proses deteksi tanaman nanas berdasarkan foto udara menggunakan *You Only Look Once* (YOLO).

2. Menghitung jumlah tanaman nanas berdasarkan foto udara menggunakan *You Only Look Once* (YOLO).
3. Menghitung lebar daun tanaman nanas dari piksel ke centimeter menggunakan metode regresi.
4. Mengkonversi lebar daun terprediksi hasil regresi linier ke biomassa tanaman nanas.

1.4 Batasan Masalah

Batasan masalah dalam penelitian ini meliputi :

1. Data yang digunakan dalam penelitian ini merupakan data gambar yang diambil dari perkebunan nanas di PT Great Giant Pineapple.
2. Dataset gambar diambil melalui *Unmanned Aerial Vehicle* (UAV).
3. Metode yang digunakan adalah metode YOLOv5s, regresi linier dan pengolahan citra.
4. Estimasi biomassa berat tanaman nanas dilakukan dengan mengukur lebar daun tanaman nanas.
5. Sistem yang dibuat berjalan di Sistem Operasi Windows 10.
6. Menggunakan bahasa pemrograman *Python*.

1.5 Manfaat

Manfaat dari penelitian ini adalah sebagai berikut :

1. Dapat memudahkan suatu pihak dalam mengestimasi berat tanaman nanas menggunakan lebar daun tanaman nanas dengan bantuan teknologi melalui model hasil penelitian ini.
2. Penggunaan *machine learning* untuk menghitung biomassa diharapkan dapat menjadi solusi untuk mengatasi masalah efisiensi pada usia tanamana nanas saat akan di *forcing*.

1.6 Sistematika Penulisan

Adapun sistematika penulisan laporan penelitian ini adalah sebagai berikut:

I. PENDAHULUAN

Pendahuluan memuat latar belakang, tujuan penelitian, perumusan masalah, batasan masalah, manfaat penelitian, dan sistematika penulisan.

II. TINJAUAN PUSTAKA

Tinjauan pustaka memaparkan beberapa teori pendukung dan referensi materi yang diperoleh dari berbagai sumber mengenai mengenai estimasi biomassa tanaman nanas berdasarkan lebar daun menggunakan algoritma *machine learning*.

III. METODOLOGI PENELITIAN

Metodologi penelitian memuat waktu dan tempat, alat dan bahan, metode penelitian, dan diagram alir penelitian.

IV. HASIL DAN PEMBAHASAN

Hasil dan pembahasan memuat perancangan dan analisis dari hasil pengujian, pembahasan hasil penelitian, dan perhitungan kinerja metode yang diusulkan.

V. PENUTUP

Memuat kesimpulan dari penelitian yang dilakukan dan saran yang didasarkan pada hasil data mengenai perbaikan dan pengembangan lebih lanjut agar didapatkan hasil lebih baik.

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Sehubungan dengan penelitian yang dilakukan, referensi terhadap penelitian sebelumnya sangat penting untuk menghindari plagiarisme atau duplikasi penelitian sebelumnya, hal ini juga dimaksudkan sebagai bahan kontribusi penelitian agar tema penelitian ini terus berkembang. Berikut beberapa ulasan penelitian terdahulu yang telah dilakukan mengenai metode yang digunakan dan hasil yang diperoleh.

Penelitian yang berkaitan dengan penggunaan metode YOLO terdapat pada penelitian yang berjudul “Pengenalan dan Perhitungan Pohon Kelapa Sawit dengan *Deep Learning* Menggunakan Citra Satelit” yang dilakukan oleh Iqbal Nurhabib (2021). Dalam penelitian ini menggunakan algoritma *Deep Learning* dan algoritma YOLO untuk mengidentifikasi dan menghitung pohon kelapa sawit. *Training dataset* menggunakan YOLO dengan jumlah iterasi sebanyak 2500 step dan mendapatkan nilai *loss* 0,6, nilai akurasi 85,4%, Presisi 97,8% dan *recall* 86,6% [7].

Penelitian mengenai pengukuran panjang dan berat tanaman terdapat pada penelitian “Pengukuran Panjang-Berat Ikan dan Sayuran secara Otomatis pada Budikdamber (Budi Daya Ikan dalam Ember) Menggunakan Visi Komputer dan Regresi Linier” oleh Hurriyatul Fitriyah (2020) menggunakan metode regresi linier. Hasil pengujian menunjukkan nilai R^2 yang tinggi pada panjang & berat lele (0,92 & 0,88), namun rendah pada panjang & berat kangkung (0,43 & 0,07).

Meski demikian, *Mean Absolute Percentage Error* (MAPE) berat dan panjang kangkung yaitu 13,37% 5,78%. MAPE dari pengukuran panjang dan berat lele yaitu 1,49% dan 4,49% [9].

Penelitian yang berkaitan dengan metode estimasi lebar terdapat pada penelitian yang berjudul “Pengukuran Badan Ikan Berupa Estimasi Panjang, Lebar, dan Tinggi Berdasarkan *Visual Capture*” yang dilakukan oleh Raihan Islamadina, Nuriza Pramita, Fitri Arnia, Khairul Munadi, TWK Muhammad Iqbal (2018). Dalam penelitian ini menggunakan metode *grayscale*, segmentasi dan proses ekstraksi fitur perhitungan badan ikan dari nilai rerata kalibrasi dan secara otomatis didapatkan nilai estimasi panjang, lebar, dan tinggi ikan. estimasi badan ikan berdasarkan *visual capture* mampu menghasilkan tingkat akurasi kebenaran sebesar 80% hingga 95% [10].

Penelitian terkait metode *machine learning Automated image identification, detection, dan counting* terdapat pada penelitian yang berjudul “*Automated image identification, detection, and fruit counting of top-view pineapple crown using machine learning*” yang dilakukan oleh Wan Nurazwin Syazwani R, Muhammad Asraf H, Megat Syahirul Amin M.A, Nur Dalila K.A. (2021). Penelitian ini menggunakan algoritma *Artificial Neural Network (ANN), Support Vector Machine (SVR), Random Forest (RF), Naive Bayes (NB), decision tree (DT), k-nearest neighbours (k-NN)* untuk mengidentifikasi dan mengenali gambar mahkota nanas di plot dan menghitung gambar yang terdeteksi menggunakan klasifikasi *machine learning*. Penelitian ini menghasilkan akurasi 94.4% pada ANN-GDX *machine learning algorithm* sebagai klasifikasi terbaik dibandingkan dengan algoritma pengklasifikasi lainnya [11].

Penelitian terkait perhitungan biomassa dan luas daun terdapat pada penelitian yang berjudul “*Modeling strawberry biomass and leaf area using object-based analysis of high-resolution images*” yang dilakukan oleh Zhen Guana, Amr Abd-Elrahmana, Zhen Fanc, Vance M. Whitakerc, Benjamin Wilkinson (2020). Dalam penelitian ini menggunakan metode *Object-Based Image Analysis (OBIA)* dan *Multiple Linear Regression*. Hasil penelitian ini menunjukkan bahwa Analisis

regresi yang dilakukan dengan menggunakan *image-derived canopy variable* sebagai prediktor untuk memodelkan luas daun ($R^2 = 0,79$; sepuluh kali lipat validasi silang $RMSE = 0,056 \text{ m}^2$) dan biomassa kering ($R^2 = 0,84$; sepuluh kali lipat validasi silang $RMSE = 7,72\text{g}$) yang diperoleh melalui pengukuran destruktif. Hasil menunjukkan daya prediksi yang konsisten sepanjang musim dan di 17 genotipe stroberi [12].

Penelitian berkaitan deteksi objek terdapat pada penelitian yang berjudul “Deteksi Tanaman Tebu Pada Lahan Pertanian Menggunakan Metode *Convolutional Neural Network*” yang dilakukan oleh Muhammad Alfin Jimly Asshiddiqie, Basuki Rahmat, Fetty Tri Anggraeny (2020). Penelitian ini menggunakan metode algoritma CNN YOLO (*You Only Look Once*). Dihasilkan rata-rata nilai *confidence* pada pengujian video sebesar 95%. Pengujian menggunakan nilai *threshold* 0.1, menghasilkan *presisi* 1.00, *recall* 0.95 dan akurasi 0.95 [13].

Penelitian mengenai deteksi objek terdapat pada penelitian yang berjudul “*identification of plant stomata based on YOLO v5 Deep Learning model*” yang dilakukan oleh Fangtao Ren, Yawei Zhang, Xi Liu, ying zhang, ying liu, fan zhang (2021). Penelitian ini menggunakan metode algoritma CNN (*Convolutional Neural Network*) dengan menggunakan YOLO v5. Hasil penelitian ini menunjukkan bahwa tingkat presisi dan *recall* dari model yang ditingkatkan pada set data stomata daun jagung masing-masing telah mencapai 94,8% dan 98,7% [14].

Penelitian terkait perhitungan objek dan deteksi objek terdapat pada penelitian yang berjudul “*Leaf counting in rice (Oryza Sativa L.) using object detection: a Deep Learning approach*” yang dilakukan oleh Mukesh Kumar Vishal, Biplab Banerjee, Rohit Saluja, Dhandapani Raju, Viswanathan Chinnusamy, Sudhir Kumar, Rabi Narayan Sahoo and Jagarlapudi Adinarayana (2020). Dalam penelitian ini menggunakan metode YOLO. Model mendeteksi daun tanaman padi (gambar RGB) dengan mendeteksi ujung daun yang sesuai dengan YOLO yang memiliki akurasi rata-rata hingga 82% dan IOU sekitar 0,53-0,60 dan

memperkirakan jumlah daun di tanaman dengan menghitung prediksi kotak pembatas di sekitar ujung [15].

Penelitian mengenai luas daun terdapat pada penelitian yang berjudul “*Non-destructive estimation of the leaf weight and leaf area in cacao (Theobroma cacao L.)*” yang dilakukan oleh Juan Carlos Suárez Salazar, Luz Marina Melgarejo, Ervin Humprey Durán Bautista, Julio A. Di Rienzo, Fernando Casanoves (2018). Penelitian ini menggunakan metode regresi linier. Luas daun dan berat daun dimodelkan menggunakan model regresi linier yang berbeda berdasarkan panjang dan lebar daun. Pada penelitian ini regresi polinomial yang melibatkan panjang dan lebar daun memberikan model yang sangat baik untuk memperkirakan luas yang diharapkan ($R^2 = 0,98$) dan berat ($R^2 = 0,91$) dari daun [16].

Penelitian mengenai perhitungan objek terdapat pada penelitian yang berjudul “Perancangan Aplikasi Perhitungan Jumlah Pohon Mangga Berbasis *Deep Learning* Memanfaatkan *Low-Altitude Remote Sensing*” yang dilakukan oleh Maheqsa Alamsyah Sutikno Putra (2020). Dalam penelitian ini menggunakan peneliti menggunakan metode *faster RCNN*. Penelitian ini Menggunakan 2 model yaitu *Inception V2* dan *ResNet-50*. Hasil penelitian menunjukkan data dengan nilai tertinggi didapatkan oleh data RGB dengan nilai presisi 97,62%, *recall* 74,55%, akurasi 73,21% dan *error* 26,79% pada model *inception V2*. Sedangkan pada model *ResNet-50* didapatkan nilai presisi 95,45%, *recall* 95,45%, akurasi 91,30%, dan *error* 8,70% [17].

Penelitian mengenai pengukuran lebar tanaman terdapat pada penelitian yang berjudul “*An image processing approach for measurement of chili plant height and width under field conditions*” oleh Chanchal Gupta, V.K. Tewari, Rajendra Machavaram, Prateek Shrivastava (2021). Penelitian ini menggunakan metode *OpenCV* dan *Python*. Hasil penelitian dari jurnal ini menunjukkan nilai *RMSE* 0.30 - 60 cm dan *R-squared* adalah 0.80–0.95 [18].

Tabel 2. 1 Penelitian terkait

No	Judul Jurnal	penulis/tahun	Metode	Hasil
1	<i>Pineapple biomass estimation using unmanned aerial vehicle in various forcing stage: Vegetation index approach from ultra-high-resolution image</i>	Aditya Nugraha Putraa, Wanda Kristiawati, Dewi Camila Mumtazydaha, Tiaranita Anggarwati, Renata Annisa, Dinna Hadi Sholikaha, Dwi Okiyanto, Sudarto (2021) [6]	<i>Index Vegetations:</i> GDVI, OSAVI, NDVI, dan TDVI, dan regresi linier	Regresi formula terpilih : F-5: $Y=30.08(\text{GDVI})-20.48$ dengan $R^2=50.70\%$; F-4: $Y=56.00(\text{GDVI})-44.30$ with $R^2=40.60\%$; when F-3 is $Y=67.20(\text{GDVI})-51.30$ with $R^2=28.20\%$; when F-2 is $Y=25.43(\text{OSAVI})+1.48$ with $R^2=45.20\%$; when F-1 is $Y=54.90(\text{OSAVI})-7.55$ with $R^2=29.80\%$; when F0 is $Y=38.90(\text{TDVI})-22.80$ with $R^2=17.60\%$; when F+1 is $Y=67.90(\text{GDVI})-40.70$ with $R^2=30.10\%$;
2	Pengenalan dan Perhitungan Pohon Kelapa Sawit Dengan <i>Deep Learning</i> Menggunakan Citra Satelit	Iqbal Nurhabib (2021) [7]	YOLO	<i>Training</i> dataset menggunakan YOLO dengan iterasi 2500 step dan mendapatkan nilai <i>loss</i> sebesar 0,6 dengan nilai akurasi 85,4%, Presisi 97,8% dan <i>recall</i> 86,6%.
3	Pengukuran Panjang-Berat Ikan dan Sayuran secara Otomatis pada Budikdamber (Budi Daya Ikan dalam Ember) Menggunakan Visi Komputer dan Regresi Linier	Hurriyatul Fitriyah (2020) [9]	Regresi linier	nilai R^2 pada panjang & berat lele (0,92 & 0,88), nilai R^2 pada panjang & berat kangkung (0,43 & 0,07). <i>Mean Absolute Percentage Error</i> (MAPE) berat dan panjang kangkung yaitu 13,37% 5,78%. MAPE dari pengukuran panjang dan berat lele yaitu 1,49% dan 4,49%.
4	Pengukuran Badan Ikan Berupa Estimasi Panjang, Lebar, dan Tinggi Berdasarkan <i>Visual Capture</i>	Raihan Islamadina, Nuriza Pramita, Fitri Arnia, Khairul Munadi, TWK Muhammad Iqbal (2018) [10]	Segmentasi <i>grayscale</i> , ekstraksi fitur	akurasi kebenaran sebesar 80% - 95%

No	Judul Jurnal	penulis/tahun	Metode	Hasil
5	<i>Automated Image Identification, detection, And Fruit Counting Of Top-view Pineapple Crown Using Machine learning</i>	Wan Nurazwin Syazwani R, Muhammad Asraf H, Megat Syahirul Amin M.A, Nur Dalila K.A. (2021) [11]	<i>Artificial Neural Netwrok (ANN), Support Vector Machine (SVR), Random Forest (RF), Naive Bayes (NB), decision tree (DT), k-NN</i>	ANN-GDX <i>machine learning</i> algoritma mendapatkan akurasi 94.4% sebagai klasifikasi terbaik dibandingkan dengan algoritma pengklasifikasi lainnya.
6	<i>Modeling strawberry biomass and leaf area using object-based analysis of high-resolution images</i>	Zhen Guana, Amr Abd-Elrahmana, Zhen Fanc, Vance M. Whitakerc, Benjamin Wilkinson (2020) [12]	<i>Object-Based Image Analysis (OBIA), Multiple Linear Regression</i>	R2 = 0,79; sepuluh kali lipat validasi silang, RMSE = 0,056 m2 dan biomassa kering (R2 = 0,84; sepuluh kali lipat validasi silang RMSE = 7,72 g)
7	Deteksi Tanaman Tebu Pada Lahan Pertanian Menggunakan Metode <i>Convolutional Neural Network</i>	Basuki Rahmat, Muhammad Alfin Jimly Asshiddiqie, Fetty Tri Anggraeny (2020) [13]	algoritma CNN (<i>Convolutional Neural Network</i>) dengan menggunakan YOLO (<i>You Only Look Once</i>).	metode CNN dapat mendeteksi tebu dengan baik dan menghasilkan rata-rata nilai confidence 95% pada pengujian video. Pengujian menggunakan nilai <i>ambang batas</i> 0.1, menghasilkan <i>presisi</i> 1.00, <i>recall</i> 0.95 dan <i>akurasi</i> 0.95
8	<i>identification of plant stomata based on YOLO v5 Deep Learning model</i>	Fangtao Ren, Yawei Zhang, Xi Liu, yingi zhang, ying liu, fan zhang (2021) [14]	YOLOv5	Presisi 94,8% dan recall 98,7%.
9	<i>Leaf counting in rice (Oryza Sativa L.) using object detection: a Deep Learning approach</i>	Mukesh Kumar Vishal, Biplab Banerjee, Rohit Saluja, Dhandapani Raju, Viswanathan Chinnusamy, Sudhir Kumar, Rabi Narayan Sahoo, Jagarlapudi Adinarayana (2020) [15]	YOLO	akurasi rata-rata hingga 82% dan IOU sekitar 0,53-0,60
10	<i>Non-destructive estimation of the leaf weight and leaf area in cacao</i>	Juan Carlos Suárez Salazar, Luz Marina Melgarejo, Ervin	<i>linear regression</i>	Luas daun dan berat daun dimodelkan menggunakan model regresi linier yang

No	Judul Jurnal	penulis/tahun	Metode	Hasil
	<i>(Theobroma cacao L.)</i>	Humprey Durán Bautista, Julio A. Di Rienzo, Fernando Casanoves (2018) [16]		berbeda berdasarkan panjang dan lebar daun. Luas ($R^2 = 0,98$) dan berat ($R^2 = 0,91$) dari daun
11	Perancangan Aplikasi Perhitungan Jumlah Pohon Mangga Berbasis <i>Deep Learning</i> Memanfaatkan <i>Low-Altitude Remote Sensing</i>	Maheqsa Alamsyah Sutikno Putra (2020) [17]	Faster-RCNN	Menggunakan 2 model yaitu <i>Inception V2</i> dan <i>ResNet-50</i> . Data dengan nilai tertinggi didapatkan oleh data RGB dengan nilai presisi 97,62%, recall 74,55%, akurasi 73,21% dan eror 26,79% pada model <i>inception V2</i> . Sedangkan pada model <i>ResNet-50</i> didapatkan nilai presisi 95,45%, recall 95,45%, akurasi 91,30%, dan eror 8,70%.
12	<i>An image processing approach for measurement of chili plant height and width under field conditions</i>	Chanchal Gupta, V.K. Tewari, Rajendra Machavaram, Prateek Shrivastava (2021) [18]	OpenCV dan Python	nilai RMSE 0.30 - 60 cm dan R^2 adalah 0.80–0.95.

2.2 Estimasi Biomassa

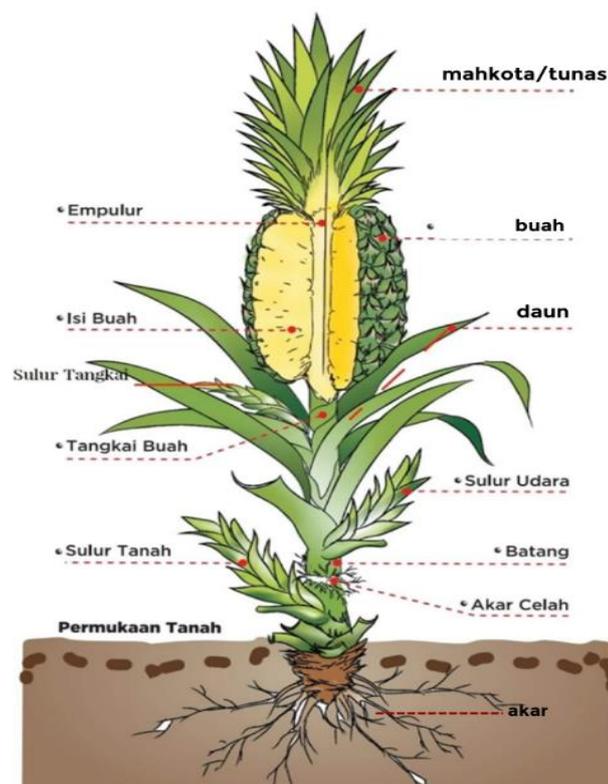
Dalam Kamus Besar Bahasa Indonesia (KBBI), estimasi memiliki definisi pendapat, penilaian, atau perkiraan. Sedangkan secara istilah, beberapa ahli yang menyebutkan bahwa estimasi akurasi adalah pengukuran yang didasarkan pada hasil numerik (kuantitatif) atau hasil yang diukur dalam angka.

Biomassa adalah total berat atau volume organisme dalam area tertentu. Biomassa adalah produk fotosintesis yang menyerap energi sinar matahari dan mengubah karbon dioksida dengan air menjadi campuran karbon, hidrogen, dan oksigen. Produksi biomassa menggambarkan pertumbuhan tanaman. Biomassa termasuk limbah hutan dan pabrik, tanaman pertanian dan limbah kayu kotoran hewan,

limbah operasi ternak, tanaman air, pertumbuhan pohon dan tanaman, sampah kota dan industri [19]. Pendekatan yang digunakan untuk mengukur berat biomassa tanaman adalah berat tanaman basah dan tanaman kering.

2.3 Nanas

Nanas (*Ananas comosus*, L. Merr) datang dari Amerika Selatan dengan daerah yang beriklim tropik. Tanaman nanas mempunyai banyak manfaat terutama pada buahnya. Buah nanas dapat dikonsumsi secara langsung atau diolah menjadi berbagai produk, seperti jus, selai, sirup, dan keripik. Buah nanas mengandung senyawa *bromelin* yang dapat melunakkan daging. Setiap 100 g buah mengandung air 80 - 86.2 %, gula 10 - 18 g, asam organik 0.5 - 1.6 g, mineral 0.3 - 0.6 g, nitrogen 4.5 - 12 mg, dan protein 180 mg. Nanas mengandung semua vitamin esensial meskipun dalam jumlah kecil, kecuali vitamin D. Selain daging buah, kulit buah bisa diolah menjadi sirup atau diekstraksi cairannya untuk pakan ternak. Serat terutama pada daun bisa dimanfaatkan sebagai bahan kertas dan tekstil.



Gambar 2.1 Morfologi Tanaman Nanas [20]

Tanaman nanas masuk ke jenis tumbuhan berbuku-buku yang menjulang keatas dan banyak jenisnya tergantung dari mana bibit yang didapatkan. Klasifikasi tanaman nanas menurut Bartholomew D P, Paull R E dan Rohrbach [21], yaitu sebagai berikut:

Kingdom : *Plantae* (tumbuh-tumbuhan)

Divisi : *Spermatophyta* (tumbuhan berbiji)

Kelas : *Angiosperma* (berbiji tertutup)

Ordo : *Farinosae* (Bromeliales)

Famili : *Bromiliaceae*

Genus : *Ananas* dan *Pseudoananas*

Species : *Ananas comosus [L] Merr*

Tanaman nanas merupakan tanaman yang masuk ke golongan tanaman tahunan. Morfologi tanaman nanas terdiri atas mahkota (*crown*), *axillary crownlets*, tunas buah (*slip*), *stem shoot*, dan tunas batang (*ground shoot* atau *sucker*). kedalaman perakaran pada media tanah yang baik antara 30-50 cm.

Batang tanaman nanas dapat dilihat jika daun-daun dihilangkan. Hal ini disebabkan batang tumbuhan nanas sangat pendek yaitu 20-25 cm dengan diameter bawah 2-3,5 cm, sedangkan diameter bagian tengah 5,5-6,5 cm dan mengecil di bagian puncak 2,0-3,5 cm. Batang tanaman nanas berbentuk ruas-ruas dengan panjang yang antara 1-10 cm. Batang memiliki fungsi sebagai tempat melekat akar, bunga, tunas bunga, buah, dan daun, sehingga terlihat dengan mata batang tersebut tidak nampak, karena di sekitarnya tertutup daun. Tangkai bunga atau buah merupakan perpanjangan batang.

Daun nanas berbentuk pedang dengan panjang sekitar \pm 100 cm dan lebar 2-8 cm, ujung daun berbentuk lancip dan tepi daun memiliki duri dan warna hijau atau hijau kemerahan. Daun nanas bersatu dalam roset akar, dimana sisi pangkalnya melebar menjadi pelepah. Pada mulanya daun nanas akan tumbuh melambat

setelah beberapa lama dan menjadi cepat seiring dengan penambahan umur tanaman.

Bunga tanaman nanas bersifat majemuk terdiri dari 50-200 kuntum bunga tunggal atau lebih. Letak bunga tegak lurus pada tangkai buah, dan bertumbuh menjadi buah mejemuk. Bunga nanas bersifat *hermaprodit*, mempunyai tiga mahkota, tiga kelopak, sebuah putik dengan kepala putik bercabang tiga dan enam benang sari. Penyerbukan tanaman nanas bersifat *self incompatible* atau *cross pollinated* melalui perantara lebah dan burung. Masa pertumbuhan bunga nanas membutuhkan sekitar 10-20 hari. Waktu dari menanam sampai terbentuk bunga antara 6-16 bulan.

2.4 Forcing

Forcing adalah proses pemaksaan munculnya bunga secara seragam. Upaya mempercepat pembungaan, pada tanaman nanas lazim dilakukan *forcing*, yaitu pemberian zat pengatur pertumbuhan *succinic acid*, *2-chloroetilfosfonic acid* (*ethephon*), yang digunakan untuk memicu *primordia* bunga. Kegiatan ini dilakukan pada siang hari sehingga hasil buah kurang maksimal. *Forcing* harus dilakukan pada malam hari karena tanaman nanas mempunyai sifat fisiologis stomata daun terbuka pada malam hari [22]. Keberhasilan *forcing* dapat diketahui setelah 50 hari melalui aktivitas pengamatan persentase pembungaan.

2.5 Artificial Intelligence

Artificial Intelligence (AI) merupakan bidang ilmu komputer yang spesifik untuk memecahkan hal kognitif yang umumnya terkait dengan kecerdasan manusia, seperti pembelajaran, pemecahan masalah, dan pengenalan pola. Bisa disebut juga tiruan dari kecerdasan yang dimiliki oleh manusia yang dimodelkan di dalam mesin dan diprogram agar bisa berpikir seperti halnya manusia, tetapi *artificial intelligence* tidak harus membatasi dirinya terhadap metode yang diamati secara biologi.

Menurut Luckin dan Colen dalam jurnal milik Ahmad Sudi Pratikno [23] menyatakan bahwa *artificial intelligence* sebagai sistem komputer yang dibuat untuk berinteraksi dengan dunia lewat kemampuan-kemampuan tertentu dan perilaku *intelligence* yang kita sadari layak seperti manusia pada umumnya. Hal tersebut dikuatkan oleh Colen yang menyatakan bahwa *artificial intelligence* adalah teori dasar mekanisme kecerdasan dan metode empiris untuk membangun dan menguji model yang mungkin untuk mendukung teori tersebut.

2.6 Machine Learning

Machine Learning (ML) atau Mesin Pembelajaran merupakan bagian dari AI yang fokus belajar dari data (*learn from data*), yang berarti berfokus pada pengembangan sistem yang dapat dipelajari "secara mandiri" tanpa diprogram berulang kali oleh manusia. ML membutuhkan data yang valid sebagai objek pembelajaran (selama proses pelatihan) sebelum digunakan ketika pengujian untuk hasil keluaran yang optimal [24].

Machine learning adalah serangkaian teknik yang dapat membantu dalam menangani dan memprediksi data yang sangat besar dengan cara mempresentasikan data-data tersebut dengan algoritma pembelajaran [25]. Jadi *machine learning* adalah sebuah pemrograman komputer yang menggunakan data masa lalu yang digunakan untuk pembelajaran model sehingga mendapatkan performa yang optimal dalam menggali informasi dari suatu kumpulan data.

Model yang dibuat oleh *machine learning* berdasarkan teori statistika dengan tujuan mengambil kesimpulan dari sampel data. Sehingga terdapat dua komponen penting dalam *computer science*. Yang pertama *training*, algoritma yang dibutuhkan peneliti untuk menyelesaikan suatu permasalahan. Yang kedua *testing*, hasil representasi dari suatu model dalam mencari solusi haruslah efisien, maka dilakukan pengujian menggunakan nilai akurasi prediksi. Secara sederhana *machine learning* membangun sebuah algoritma yang memungkinkan program komputer untuk belajar dan melakukan tugasnya sendiri tanpa adanya instruksi

dari penggunaannya. Algoritma semacam ini bekerja dengan cara membangun sebuah model dari *input* atau masukan untuk dapat menghasilkan suatu prediksi atau pengambilan keputusan berdasarkan data yang ada.

Machine learning berhubungan dengan *computational statistics* yang berfokus pada suatu prediksi atau pembuatan keputusan berdasarkan penggunaan komputer. Beberapa implementasi dari *machine learning* adalah *text analysis*, *image processing*, *fincance*, *search and recommendation engine*, *speech understanding*. Dalam pembelajaran *machine learning*, terdapat tiga kategori utama yang dapat dilihat pada Tabel 2.2.

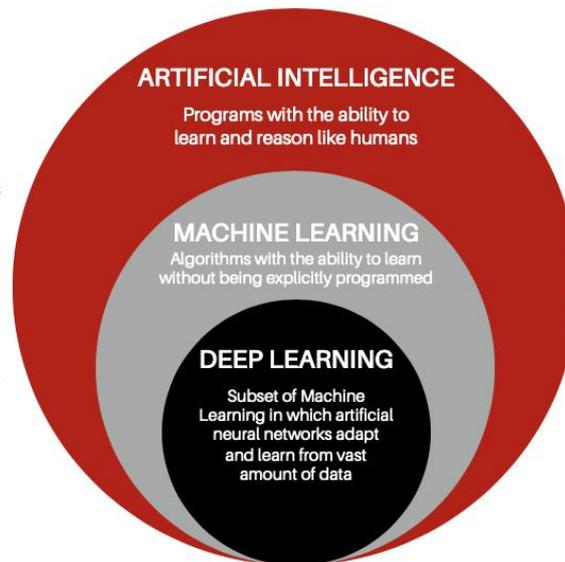
Tabel 2. 2 *Machine Learning: Supervised, Unsupervised dan Reinforcement*

Kriteria	<i>Supervised Learning</i>	<i>Unsupervised Learning</i>	<i>Reinforcement Learning</i>
Definisi	Mesin belajar dengan menggunakan data berlabel	Mesin dilatih pada data tak berlabel, tanpa panduan apapun	Agen melakukan tindakan dan belajar dari errors/rewards.
Jenis Masalah	Regresi dan Klasifikasi	Asosiasi dan Clustering	<i>Reward-based</i>
Jenis Data	Data berlabel	Data tidak berlabel	<i>No predefined data</i>
<i>Training</i>	Diawasi	Tidak diawasi	Tidak diawasi
Pendekatan	Memetakan <i>input</i> berlabel ke <i>output</i> yang di ketahui	Memahami pola dan menemukan <i>output</i>	Mengikuti metode <i>trial-and-error</i>

2.7 *Deep Learning*

Deep learning adalah salah satu bidang *machine learning* yang memanfaatkan banyak layer pengolahan informasi nonlinier untuk melakukan ekstraksi fitur, pengenalan pola, dan klasifikasi [26]. Menurut Goodfellow dkk [27] *deep learning* adalah sebuah pendekatan dalam penyelesaian masalah pada sistem pembelajaran komputer yang menggunakan konsep hierarki. Konsep hierarki memungkinkan komputer untuk mempelajari konsep kompleks dengan menggabungkan konsep yang lebih sederhana. Gambar 2.2 memperlihatkan bahwa *deep learning* adalah sub-bidang yang spesifik dari *machine learning* dan

merupakan cara baru untuk mempelajari representasi dari data yang menekankan pada pembelajaran *layer* (lapisan) yang berturut-turut dan semakin bermakna.



Gambar 2. 2 Hubungan antara *artificial intelligence*, *machine learning* dan *deep learning* [28]

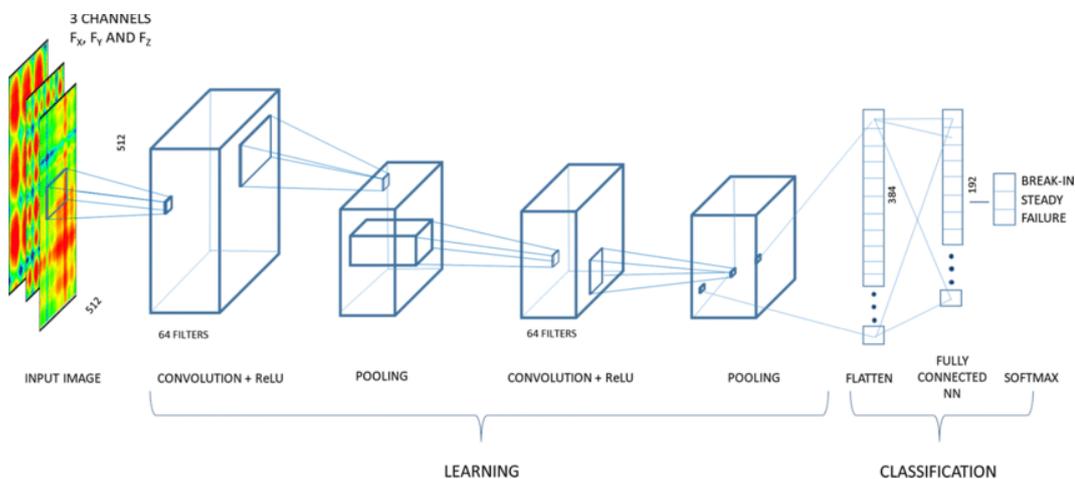
Kata '*deep*' pada *deep learning* tidak mengacu pada pemahaman yang lebih dalam untuk pendekatan, sebaliknya ini merupakan gagasan dari layer yang berurutan. Banyak lapisan yang digunakan pada model data disebut kedalaman ('*depth*') dari model. *Deep learning* moderen sering melibatkan puluhan atau bahkan ratusan layers yang berurutan, dan semuanya dipelajari secara otomatis dari data pelatihan. Sementara itu, pendekatan lain untuk *machine learning* cenderung berfokus pada pembelajaran hanya satu atau dua lapis layer.

Meskipun *deep learning* adalah sub-bidang dari *machine learning* yang cukup lama, (mulai terkenal di awal tahun 2010-an). Dalam beberapa tahun kemudian, *deep learning* telah mencapai tidak lebih dari sebuah revolusi di lapangan, dengan hasil yang luar biasa pada masalah persepsi seperti melihat dan mendengar, masalah yang melibatkan keterampilan yang tampak alami dan intuitif bagi manusia tetapi telah lama sulit dipahami oleh mesin. Beberapa terobosan yang telah dicapai *deep learning* adalah peningkatan konversi *text-to-speech*, asisten digital seperti Google Now dan Amazon Alexa dan kemampuan untuk menjawab

pertanyaan-pertanyaan bahasa alami. Beberapa metode dari *deep learning* adalah *Long Short Term Memory* (LSTM) dan *Convolutional Neural Network* (CNN) [29].

2.8 Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) adalah tipe khusus dari *neural network* untuk memproses data yang mempunyai topologi jala atau *grid-like topology*. Pemberian nama *convolutional neural network* menandakan bahwa jaringan tersebut menggunakan operasi matematika yang disebut konvolusi. Konvolusi sendiri adalah sebuah operasi linear. Jadi *convolutional network* adalah *neural network* yang menggunakan konvolusi minimal pada salah satu lapisannya [30]. *Convolutional Neural Network* (*ConvNets*) merupakan *special case* dari *artificial neural network* (ANN) yang sekarang diakui sebagai model terbaik untuk memecahkan masalah *object recognition* dan *detection*.



Gambar 2. 3 Convolution Neural Network (CNN) [30]

Struktur CNN terdiri dari *input*, proses ekstraksi fitur, proses klasifikasi dan *output*. Proses ekstraksi dalam CNN terdiri dari beberapa *hidden layer*, yaitu lapisan konvolusi, fungsi aktivasi (ReLU), dan *pooling*. CNN bekerja secara hierarki, sehingga *output* pada lapisan konvolusi pertama berfungsi sebagai *input* pada lapisan konvolusi selanjutnya. Pada proses klasifikasi terdiri dari *fully-*

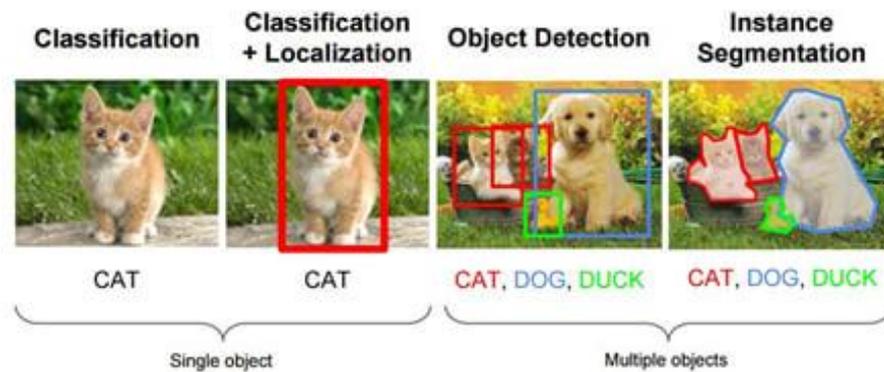
connected dan fungsi aktivasi (*softmax*) yang *outputnya* berupa hasil klasifikasi [31].

2.9 Computer Vision

Computer Vision atau yang sering disebut CV didefinisikan sebagai bidang studi yang berupaya mengembangkan teknik untuk membantu komputer "melihat" dan memahami konten gambar dan video digital. Itu tidak lain adalah bidang ilmiah yang memungkinkan komputer untuk menangkap, menafsirkan, memahami, dan memproses objek-objek yang ada sehingga dapat dipahami secara visual [32]

Computer vision adalah bidang multidisiplin yang bisa disebut subbidang dari *artificial intelligence* (AI) dan *machine learning* (ML), CV menggunakan beberapa metode khusus dan menggunakan algoritma [32]. Tujuan dari *computer vision* adalah untuk memahami isi gambar digital. *Computer vision* adalah kombinasi dari *image processing* dan *pattern recognition*. *Image processing* merupakan suatu bidang yang berhubungan dengan proses transformasi citra (*image*). Proses ini bertujuan untuk mencapai kualitas gambar yang lebih baik. Di sisi lain, *pattern recognition* mengacu pada proses mengidentifikasi objek dalam gambar atau interpretasi gambar. Proses ini bertujuan untuk mengekstrak informasi/pesan yang dikirimkan oleh citra.

Computer vision menggunakan dua teknologi yaitu *deep learning* dan *Convolutional Neural Network* (CNN). *Deep learning* menggunakan model algoritmik yang dapat membuat komputer mempelajari sendiri perbedaan pada sebuah foto atau video. CNN disini sangat membantu model *deep learning* untuk "melihat" dengan memecah gambar menjadi piksel yang diberi label yang kemudian dilakukan konvolusi dan pada akhirnya dapat membuat prediksi apa yang komputer tersebut lihat. Beberapa tugas yang dapat dilakukan dengan *computer vision* adalah klasifikasi gambar (*image classification*), deteksi objek (*object detection*), pelacakan objek (*object tracking*), dan pengambilan gambar berbasis konten (*content-based image retrieval*). Gambar 2.4 Adalah contoh beberapa tugas *computer vision*.



Gambar 2. 4 Tugas dalam *computer vision* [33]

2.10 Object Detection

Detektor objek berbasis CNN dapat dibagi menjadi beberapa tipe, yaitu *one-stage detector* (YOLOX, FCOS, DETR, Scaled-YOLOv4, dan EfficientDet), *two-stage detector* (VFNet dan CenterNet2), *anchor-based detector* (Scaled YOLOv4 dan YOLOv5), dan *anchor-free detector* (CenterNet, YOLOX, dan RepPoints). Dari perspektif komponen, mereka biasanya terdiri dari dua bagian, yaitu *backbone* berbasis CNN untuk ekstraksi fitur gambar, dan *head* untuk memprediksi kelas objek dan kotak pembatas (*bounding box*). Selain itu, detektor objek yang dikembangkan dalam beberapa tahun terakhir sering mencakup beberapa lapisan: lapisan antara *backbone* dan *head*, yang biasanya disebut *neck* [34].

Backbone yang sering digunakan termasuk VGG, ResNet, DenseNet, MobileNet, EfficientNet, CSPDarknet53, Swin Transformer, dan lain-lain alih-alih jaringan yang dirancang sendiri. Karena jaringan ini telah membuktikan bahwa mereka memiliki kemampuan ekstraksi fitur yang kuat pada masalah seperti klasifikasi. Tetapi para peneliti juga menyempurnakan *backbone* agar lebih cocok untuk tugas-tugas tertentu. *Neck* dirancang untuk memanfaatkan fitur yang diekstraksi oleh *backbone* dengan lebih baik. Ini memproses ulang dan secara rasional menggunakan peta fitur yang diekstraksi oleh *backbone* pada tahap yang berbeda. Sebagai jaringan klasifikasi, *backbone* tidak dapat menyelesaikan tugas pemosisian, dan *head* dirancang untuk menggunakan peta fitur yang diekstraksi

dari *backbone* untuk bertanggung jawab mendeteksi lokasi dan kategori objek. *Head* umumnya diklasifikasikan menjadi dua jenis: *one-stage detector* dan *two-stage detector*. *Two-stage detector* telah lama menjadi metode dominan di bidang deteksi objek, yang paling representatif adalah keluarga RCNN. Dibandingkan dengan *two-stage detector*, *one-stage detector* memprediksi kotak pembatas dan kelas objek secara bersamaan. Keuntungan kecepatan *one-stage detector* sudah jelas, tetapi akurasinya lebih rendah. Untuk *one-stage detector*, model yang paling representatif adalah seri YOLO, SSD, dan RetinaNet [34].

2.11 You Only Look Once (YOLO)

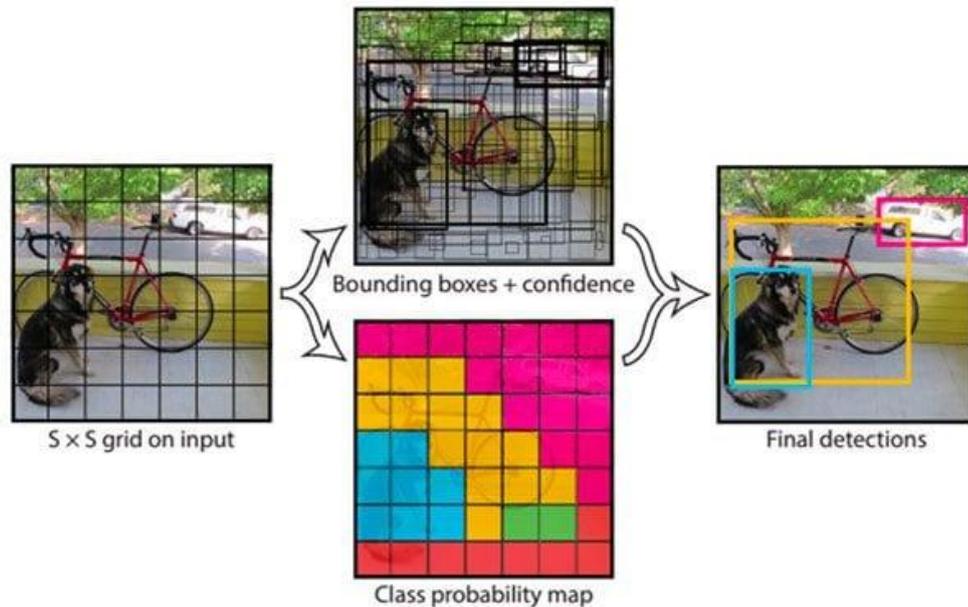
You Only Look Once merupakan algoritma *object detection* yang hadir pada tahun 2016. Cara kerja dari YOLO ini tidak rumit, yakni dengan memanfaatkan CNN sebagai pendeteksi objek. YOLO menerima sebuah *input image* yang dibagi menjadi kotak sebesar $S \times S$ yang dikirimkan oleh sebuah *neural network* untuk pembuatan *bounding box* kemudian sistem memprediksi menggunakan pembagian dimensi sebagai *anchor box* [35].

YOLO membagi gambar menjadi $S \times S$ grid cell sebelum memprediksi objek. Apabila titik pusat dari suatu objek berada dalam sebuah *grid cell*, maka *grid cell* tersebut bertanggung jawab untuk mendeteksi objek yang bersangkutan. Setiap *grid cell* memprediksi *bounding box* dan nilai *confidence*. Nilai *confidence* menunjukkan seberapa akurat model memprediksi objek yang terdapat dalam *bounding box*. Nilai *confidence* dari prediksi objek yang terdeteksi didapatkan dengan menggunakan Persamaan 2.1 [36]:

$$\text{Confidence score} = p(\text{Object}) * IOU \frac{\text{truth}}{\text{pred}} \dots\dots\dots 2.1$$

$p(\text{Object})$ merupakan probabilitas adanya objek yang terdeteksi dalam sebuah *grid cell* sedangkan $IOU \text{ truth pred}$ merupakan perkiraan irisan antara hasil deteksi dengan data yang ada dalam model. Jika tidak ada objek dalam *bounding box* maka *confidence* bernilai nol.

Bounding box memiliki empat buah nilai yaitu x , y , w , h . Nilai (x, y) merepresentasikan titik pusat dari *bounding box*, w merepresentasikan ukuran lebar dari *bounding box* sedangkan h merepresentasikan ukuran tinggi dari *bounding box*.



Gambar 2. 5 Model algoritma deteksi YOLO [35]

Selain itu, *grid cell* juga menentukan probabilitas kelas objek yang kemungkinan terdeteksi didalamnya. Probabilitas kelas hanya ditentukan pada *grid cell* yang memiliki objek terdeteksi. Model algoritma hanya mendeteksi satu set probabilitas kelas objek untuk setiap *grid cell* tanpa memerdulikan jumlah *bounding box* yang terdapat dalam *grid cell* tersebut. Gambar 2.5 memperlihatkan model algoritma untuk mendeteksi objek pada YOLO.

Saat ini, algoritma pengenalan objek yang utama adalah seri R-CNN dan seri YOLO. Seri R-CNN lebih unggul dalam deteksi target dan membutuhkan akurasi yang lebih tinggi, tetapi kecepatan deteksinya lebih rendah daripada seri YOLO. Dalam skenario praktis, kinerja deteksi target R-CNN secara *real time* tidak dapat dipenuhi. Dalam konteks ini, himpunan algoritma YOLO menggunakan ide regresi, sehingga lebih mudah untuk mempelajari generalisasi, menargetkan

karakteristik dan memecahkan masalah kecepatan. Keluarga algoritma YOLO menggunakan jaringan saraf *one stage* untuk menyelesaikan lokalisasi dan klasifikasi objek yang terdeteksi secara *real time* [36].

YOLO memperlakukan deteksi gambar sebagai masalah regresi dengan proses yang sederhana dan kecepatan tinggi. Oleh karena itu YOLO dapat memproses *video streaming* secara *real time* dengan latensi kurang dari 25 detik. Selama pelatihan, YOLO dapat melihat seluruh gambar dan lebih memperhatikan informasi global dalam deteksi objek. Ide inti dari YOLO adalah menggunakan seluruh gambar sebagai *input* jaringan dan langsung mengembalikan posisi kotak pembatas dan kelas kotak pembatas pada output. Di YOLO, setiap kotak pembatas diprediksi oleh fitur dari keseluruhan gambar, dan setiap kotak pembatas berisi lima prediksi dan *confidence*, yang relatif terhadap sel grid di tengah kotak pembatas. YOLO terdiri dari tiga bagian utama, yaitu [36]:

- *Backbone*: Jaringan saraf convolutional yang menggabungkan dan membentuk fitur gambar pada berbagai jenis detail gambar.
- *Neck*: Serangkaian lapisan jaringan yang mencampur dan menggabungkan fitur gambar dan melewati fitur gambar ke lapisan prediksi.
- *Head*: Komponen ini dapat memprediksi fitur gambar, menghasilkan kotak pembatas, dan memprediksi kategori. Keyakinan mewakili keakuratan klasifikasi dalam kondisi tertentu.

Model YOLO memiliki beberapa kelebihan dari model objek deteksi lainnya. Berikut adalah kelebihan model algoritma YOLO [35]:

1. Proses deteksi pada YOLO sangat cepat. Pemberian *bounding box* menggunakan regresi sehingga tidak diperlukan jaringan syaraf rumit yang digunakan berulang. YOLO menggunakan model pembelajaran sebagai acuan deteksi untuk gambar uji baru sehingga tidak diperlukan pembelajaran berulang untuk deteksi pada gambar baru. Jaringan YOLO dapat mencapai lebih dari dua kali rata-rata presisi deteksi dari sistem *real-time* lainnya.
2. YOLO menggunakan citra global saat memprediksi objek pada sebuah gambar. Tidak seperti sistem penggeseran yang berbasis proposal wilayah

(*region proposal-based techniques*). YOLO melihat keseluruhan gambar dan *feature-feature*-nya selama proses pelatihan dan pengujian sehingga secara implisit YOLO akan mengkodekan informasi kontekstual tentang kelas dan visualnya. *Fast R-CNN* sebagai metode deteksi tertinggi saat ini memiliki kesalahan *patch* latar belakang (*background patches*) dalam gambar karena tidak dapat melihat konteks uji yang lebih besar daripada *input* awal. YOLO membuat kurang dari setengah jumlah kesalahan latar belakang dibandingkan dengan *Fast R-CNN*.

3. YOLO mempelajari representasi objek yang dapat digeneralisasi dengan membuat model yang dapat digunakan untuk deteksi nantinya. Ketika YOLO dilatih dengan gambar alami kemudian diujikan dengan gambar berupa gambar seni, YOLO mengungguli metode deteksi teratas seperti DPM dan R-CNN dengan selisih nilai yang besar. Hal itu dikarenakan YOLO dapat digeneralisasikan sehingga kemungkinan kecil untuk menimbulkan *error* saat diterapkan ke domain baru atau *input* tidak terduga.

Dengan beberapa keunggulan tersebut, YOLO tetap memiliki kelemahan di mana YOLO masih tertinggal di sisi akurasi. Namun itu yang menjadi perkembangan untuk setiap versi YOLO kedepannya. Sistem YOLO membagi gambar *input* menjadi $S \times S$ *grid*. Jika pusat objek jatuh pada *grid* tersebut, maka sel *grid* tersebut yang akan bertanggung jawab untuk mendeteksi objek tersebut. Setiap sel *grid* memprediksi *bounding box*, B dan skor *confidence* atau kepercayaan terhadap objek untuk kotak tersebut. Skor kepercayaan ini mencerminkan seberapa yakin model bahwa objek tersebut berisi objek yang dituju dan seberapa akurat prakiraannya [35].

2.11.1 Deskripsi YOLO v5

YOLOv5 merupakan model pendeteksi objek generasi kelima yang dirilis pada April 2020. Secara umum arsitektur model ini tidak jauh berbeda dengan generasi YOLO sebelumnya. YOLOv5 ditulis dalam bahasa pemrograman Python, bukan bahasa C di versi sebelumnya. Hal itu membuat instalasi dan integrasi pada perangkat IoT lebih mudah. Selain itu, komunitas PyTorch juga lebih besar dari

komunitas Darknet, yang berarti PyTorch akan mendapatkan lebih banyak kontribusi dan potensi pertumbuhan di masa depan. Karena ditulis dalam dua bahasa pemrograman yang berbeda pada dua kerangka kerja yang berbeda, perbandingan kinerja antara YOLOv4 dan YOLOv5 sulit untuk akurat. Namun setelah beberapa saat, YOLOv5 terbukti lebih berperforma daripada YOLOv4 dalam beberapa kasus, sebagian memenangkan kepercayaan dari komunitas *computer vision* di luar YOLOv4 [34].

Seperti yang telah disebutkan sebelumnya, arsitektur YOLOv5 mengintegrasikan inovasi terbaru yang mirip dengan arsitektur YOLOv4, sehingga secara teori tidak ada banyak perbedaan. Pengembang model YOLOv5 tidak menerbitkan makalah terperinci, tetapi hanya meluncurkan repositori di Github dan memperbaruinya di sana. Dengan menganalisis struktur kode pada file `.yaml`, model YOLOv5. YOLOv5 menggunakan *backbone focus structure* dan *CSP network*, *neck* yang digunakan adalah SPP dan PANet, dan *head* yang digunakan adalah *head* YOLOv3 menggunakan GIoU-loss.

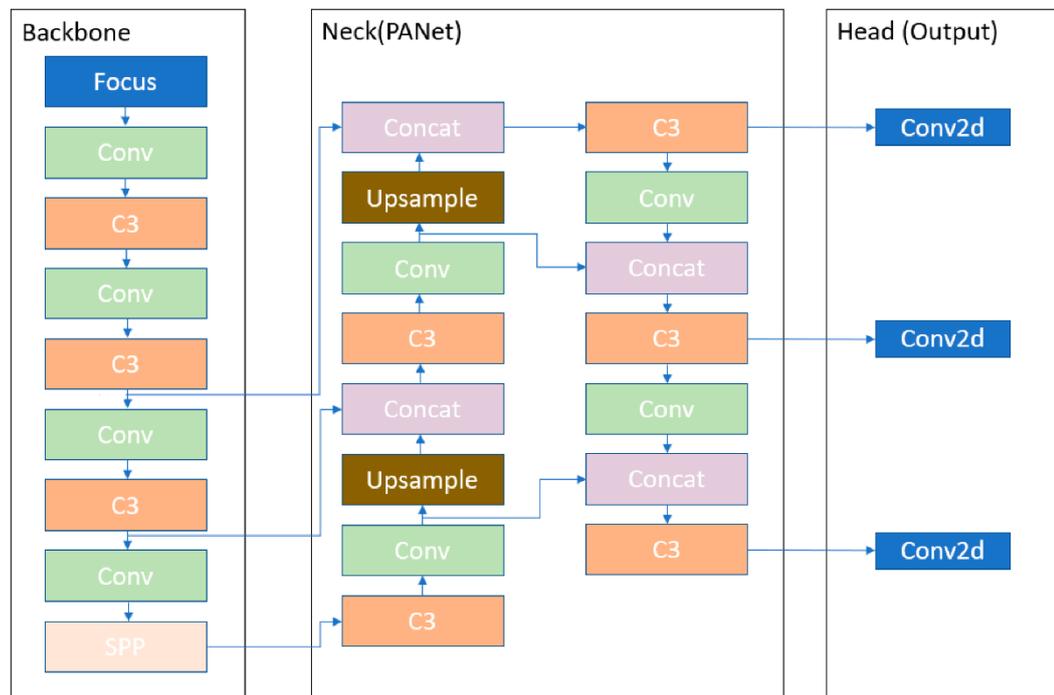
Insinyur *computer vision* biasanya menjalankan algoritma *clustering* kmeans pada dataset yang unik untuk memperoleh hasil *anchor box* yang paling baik. Kemudian, parameter ini akan dikonfigurasi manual ke dalam arsitektur YOLO. Glenn Jocher mengusulkan untuk mengintegrasikan proses pemilihan *anchor box* ke dalam YOLOv5. Oleh karena itu, jaringan YOLOv5 tidak perlu mempertimbangkan parameter ini pada dataset apapun sebagai *input*, hal ini akan membuat model otomatis belajar *anchor box* yang paling baik pada dataset tersebut dan menggunakannya selama pelatihan [36].

2.11.2 Arsitektur YOLO v5

YOLOv5 berbeda dari model rilis sebelumnya. YOLOv5 menggunakan PyTorch sebagai gantinya dari Darknet. YOLOv5 menggunakan CSPDarknet53 sebagai *backbone*. Backbone ini memecahkan repitisi informasi gradien di *backbone* besar dan mengintegrasikan perubahan gradien ke dalam peta fitur yang mengurangi

kecepatan inferensi, meningkatkan akurasi, dan mengurangi ukuran model dengan mengurangi parameter [37].

YOLOv5 menggunakan *Path Aggregation Network* (PANet) sebagai *neck* untuk meningkatkan arus informasi. PANet mengadopsi *Feature Pyramid Network* (FPN) yang baru mencakup beberapa lapisan *bottom up* dan *top down*. Hal ini dapat meningkatkan penyebaran level rendah fitur dalam model. PANet meningkatkan melakukan pembatasan (lokalisasi) di lapisan bawah, yang meningkatkan akurasi lokalisasi objek [38]. Selain itu, *head* di YOLOv5 sama dengan YOLOv4 dan YOLOv3 yang menghasilkan tiga keluaran peta fitur yang berbeda untuk mencapai skala *multi prediction*. Hal ini juga membantu meningkatkan prediksi objek kecil hingga besar secara efisien dalam model. Gambar dimisalkan ke CSPDarknet53 untuk ekstraksi fitur dan dimisalkan lagi ke PANet untuk fusi fitur. Arsitektur YOLOv5 ditunjukkan oleh Gambar 2.6.



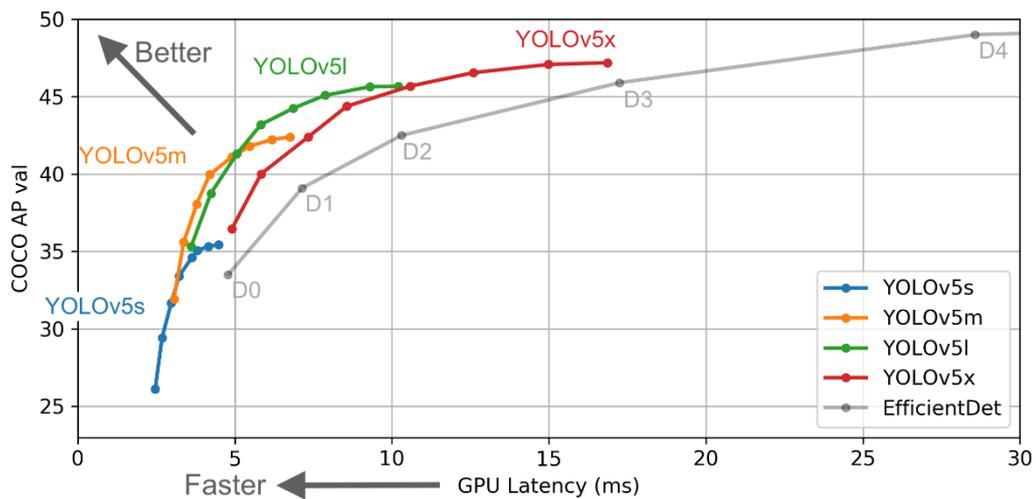
Gambar 2. 6 Arsitektur YOLOv5 [39]

Conv mengindikasikan lapisan konvolusi. C3 terdiri dari tiga lapisan konvolusi dan modul yang mengalir oleh berbagai gangguan. *Spatial Pyramid Pooling* (SPP) adalah lapisan *pooling* yang digunakan untuk menghilangkan batasan

ukuran tetap pada jaringan. *Upsample* dipakai dalam *upsampling* fusi *layer* sebelumnya di *node* terdekat. *Concat* adalah *layer* pengiris dan digunakan untuk mengiris *layer* sebelumnya. Tiga *Conv2d* terakhir adalah modul deteksi objek yang digunakan di head dari jaringan [36].

2.11.3 Tipe YOLOv5

YOLOv5 memiliki beberapa tipe dengan masing-masing memiliki perbedaan dari segi kecepatan deteksi dan performa mAPnya. Berikut gambar dari perbedaan dari masing-masing tipe YOLO v5 [40]. Pada Gambar 2.7 digambarkan bahwa semakin keatas arah grafik model maka akan semakin baik dalam mAP model tetapi semakin ke kiri maka semakin cepat untuk deteksi.



Gambar 2. 7 Peforma tipe model YOLOv5 [40]

Tipe Model YOLOv5 dilatih dengan data COCO dengan size 640 piksel. Dengan melihat dari *speed* atau kecepatannya, akan semakin besar bila params juga semakin besar. Params merupakan parameter *training*. Oleh karena itu komputasi akan lebih lama jika parameter semakin besar dan deteksi akan semakin lambat [20].

Pada Tabel 2.3 ditampilkan beberapa indikator yang bisa menjadi pertimbangan dalam memilih model.

Tabel 2. 3 Peforma tipe model YOLOv5 [41]

Model	Size (Pixels)	mAPval 0.5:0.95	mAPval 0.5	Speed CPU b1 (ms)	Speed V100 b1 (ms)	Speed V100 b32 (ms)	Params (M)	FLOPs @640 (B)
YOLOv5n	640	28.4	46.0	45	6.3	0.6	1.9	4.5
YOLOv5s	640	37.2	56.0	98	6.4	0.9	7.2	16.5
YOLOv5m	640	54.2	63.9	224	8.2	1.7	21.2	49.0
YOLOv5l	640	48.8	67.2	430	10.1	2.7	46.5	109.1
YOLOv5x	640	50.7	68.9	766	12.1	4.8	86.7	205.7

2.12 Regresi

Regresi merupakan metode analisis yang mengidentifikasi relasi atau hubungan diantara 2 variabel atau lebih. Regresi masuk ke dalam *supervised learning* yang berfungsi memprediksi nilai kontinu. Tujuan dari regresi untuk menemukan suatu fungsi yang memodelkan data dengan meminimalkan error atau selisih antara nilai prediksi dan nilai sebenarnya.

Metode regresi merupakan sebuah metode statistik yang dapat melakukan prediksi menggunakan pengembangan hubungan matematis antara variabel, yaitu variabel dependen (Y) dengan variabel independen (X) [42]. Variabel dependen merupakan variabel akibat atau variabel yang dipengaruhi, sedangkan variabel independen merupakan variabel sebab atau variabel yang mempengaruhi [43]. Prediksi terhadap nilai variabel dependen dapat dilakukan jika variabel independennya diketahui. Perlu diperhatikan dalam kasus regresi adalah nilai dari variabel dependen (Y) harus berupa nilai kontinu, bukan diskrit. Sementara itu untuk variabel independen (X) bisa berupa nilai kontinu maupun kategori, misalnya sedan, *hatchback*, *wagon*, *convertible*.

Ada 4 tipe model klasifikasi dalam regresi yaitu :

1. *Binary classification* yaitu jika ketagori dalam target variabel hanya ada dua, misalnya 0 dan 1, Yes dan No, X dan Y, dan sebagainya.
2. *Multi-class classification* memiliki lebih dari dua ketagori pada variabel targetnya.

3. *Simple regression* (regresi sederhana) adalah ketika hanya satu variabel independen yang digunakan untuk mempresiksi variabel dependen, bisa berupa linear (*simple linear regression*) maupun non-linear (*simple non-linear regression*).
4. *Multiple regression* (regresi berganda) yaitu ketika ada lebih dari satu variabel independen yang digunakan untuk memprediksi variabel dependen.

2.12.1. Regresi Linier

Dalam kasus regresi linier satu dimensi *input*, dimana misalnya *input* $x = [x_1, x_2, x_3, x_4, x_5]$ atau disebut variabel independen, dan *output* $y = [y_1, y_2, y_3, y_4, y_5]$ atau biasa disebut variabel dependen [44]. Persamaan linear satu dimensi *input* (sederhana) dapat ditulis sebagai berikut:

$$y = a + bX \dots\dots\dots 2.2$$

$$b = \frac{n\sum X.Y_1 - (\sum X)(\sum Y_1)}{n\sum X^2 - (\sum X)^2} \dots\dots\dots 2.3$$

$$a = \frac{\sum X^2 \sum Y_1 - \sum X \sum XY_1}{n\sum X^2 - (\sum X)^2} \dots\dots\dots 2.4$$

Keterangan :

a = Konstanta

b = Konstanta regresi

X = Variabel Independen

Y = Variabel dependen

Teknik umum untuk mengukur seberapa layak nilai yang dihasilkan oleh aproksimasi fungsi garis *regression* terhadap data *output* yang ada, adalah dengan menggunakan metode *R-squared* dan *Mean Absolute Percetage Error (MAPE)*.

2.13 Pengelolaan Citra Digital

Pengolahan citra digital pada awalnya hanya digunakan untuk mengubah citra analog menjadi digital dan perbaikan kualitas citra. Seiring perkembangan

peralatan-peralatan pendukung pengolahan citra maka penggunaan pengolahan citra digital menjadi lebih beragam. Melalui algoritma pengolahan citra, fungsi sensor penglihatan pada manusia diperlukan untuk dapat digantikan dengan *artificial vision sensor* (kamera). Saat kecepatan pemrosesan komputer yang meningkat pesat hal itu memungkinkan pemrosesan citra digital dilakukan secara *realtime*. Dan juga perkembangan memori memungkinkan citra analog dapat dikodekan menjadi citra warna digital yang mendekati warna aslinya [45].

Arti pengolahan menurut kamus besar Bahasa Indonesia (KBBI) adalah suatu cara atau proses mengusahakan sesuatu supaya menjadi lain atau menjadi lebih sempurna. Sementara itu, menurut KBBI citra adalah bentuk atau gambar, yaitu gambar yang diperoleh dengan bantuan sistem visual. Pemrosesan gambar secara keseluruhan melibatkan cara mengerjakan gambar untuk mengubahnya menjadi gambar lain yang lebih sempurna atau diinginkan. Dengan kata lain, image processing adalah proses memasukkan gambar dan menghasilkan keluaran berupa gambar sesuai dengan yang diinginkan. [45].

Pengolahan citra digital (*Digital Image Processing*) adalah sebuah cabang ilmu yang mempelajari tentang teknik-teknik mengolah citra. Citra yang dimaksud disini adalah gambar diam (foto) maupun gambar bergerak (yang berasal dari *webcam*). Sedangkan digital disini mempunyai maksud bahwa pengolahan citra/gambar dilakukan secara digital menggunakan komputer [46].

2.14 Citra

Menurut kamus Webster citra adalah representasi atau kemiripan dari suatu objek. Contohnya adalah foto sebuah pisang mewakili identitas buah pisang tersebut di depan sebuah kamera. Citra dapat berupa hasil fotografi, lukisan, atau gambaran serta corat-coret yang terjadi di kertas, kanvas, dan di layar monitor. Dapat dikatakan juga citra adalah sebaran variasi warna-warni, redup-cerah, dan/atau gelap-terang di suatu bidang datar. Formalitas pengungkapan dengan angka-angka yang mengindikasikan variasi intensitas kecerahan atau/dan warna pada arah mendatar dan tegak [45].

Secara matematis, citra merupakan fungsi kontinu (*continue*) dengan intensitas cahaya pada bidang dua dimensi. Agar dapat diolah dengan komputer digital, maka suatu citra harus dipresentasikan secara numerik dengan nilai-nilai diskrit. Reperesentasi dari fungsi kontinu menjadi nilai-nilai diskrit disebut digitalisasi citra.

Citra adalah suatu representasi (gambaran), kemiripan, atau imitasi dari suatu objek. Citra terbagi 2 yaitu citra yang bersifat analog dan ada citra yang bersifat digital. Citra analog adalah citra yang bersifat *continue* seperti gambar pada monitor televisi, foto sinar X, foto yang tercetak di kertas foto, lukisan, 9 pemandangan alam, hasil CT scan, gambar-gambar yang terekam pada pita kaset, dan lain sebagainya. Citra analog tidak bisa digambarkan dalam komputer, sehingga tidak bisa diproses di komputer secara *relatime*. Oleh sebab itu, agar dapat diproses di komputer, proses konversi analog ke digital harus dilakukan terlebih dahulu. Citra analog dihasilkan dari alat-alat analog, seperti video kamera analog, kamera foto analog, kamera, CT scan, sensor *rontgen* untuk foto *thorax*, sensor gelombang pendek pada sistem radar, sensor ultrasound pada sistem USG, dan lain-lain [47]. Sedangkan pada citra digital adalah citra yang dapat diolah melalui komputer.

2.15 Citra Digital

Citra digital merupakan representatif dari citra yang diambil oleh mesin dengan bentuk pendekatan berdasarkan sampling dan kuantisasi. Sampling menyatakan besarnya kotak-kotak yang disusun dalam baris dan kolom. Atau dapat dikatakan, *sampling* pada citra menyatakan besar kecilnya ukuran *piksel* (titik) pada citra, dan kuantisasi menyatakan besarnya nilai tingkat kecerahan yang dinyatakan dalam nilai tingkat keabuan (*grayscale*) sesuai dengan jumlah bit biner yang digunakan oleh mesin, dengan kata lain kuantisasi yang menyatakan jumlah warna yang ada pada citra [47].

Sebuah citra digital dapat diwakili oleh sebuah *matrix* dua dimensi $f(x,y)$ yang terdiri dari M kolom dan N baris, dengan x dan y adalah koordinat spasial, dan

amplitudo f di titik koordinat (x,y) dinamakan intensitas atau tingkat keabuan dari citra [47] dimana perpotongan antara kolom dan baris disebut piksel (*pixel = picture element*) atau elemen terkecil dari sebuah citra.

$$f(x,y) \approx \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,M-1) \\ f(1,0) & f(1,1) & \dots & f(1,M-1) \\ \vdots & \vdots & \vdots & \vdots \\ f(N-1,0) & f(N-1,1) & \dots & f(N-1,M-1) \end{bmatrix} \dots\dots\dots 2.5$$

Suatu citra $f(x,y)$ dalam fungsi matematis dapat dituliskan sebagai berikut:

$$0 \leq x \leq M-1$$

$$0 \leq y \leq N-1$$

$$0 \leq f(x,y) \leq G-1$$

dimana :

M = jumlah piksel baris (*row*) pada *array* citra

N = jumlah piksel kolom (*column*) pada *array* citra

G = nilai skala keabuan (*graylevel*)

Jenis-jenis citra dikelompokkan menjadi tiga [48] yaitu citra biner (monokrom), citra skala keabuan (*grayscale*) dan citra warna RGB (*true color*).

2.15.1 Citra Biner

Sebuah citra biner adalah sebuah citra digital yang hanya memiliki dua kemungkinan nilai untuk tiap *piksel*. Kedua warna tersebut adalah hitam dan putih [49]. Citra biner disebut juga dengan citra W&B (*White&Black*) atau citra monokrom. Hanya dibutuhkan 1 bit untuk mewakili nilai setiap piksel dari citra biner. Citra biner seringkali muncul sebagai hasil dari proses pengolahan seperti segmentasi, pengambangan, morfologi, maupun *dithering* [50]. Pembentukan citra biner memerlukan nilai batas keabuan yang akan digunakan sebagai nilai patokan. *Piksel* dengan derajat keabuan lebih besar dari nilai batas akan diberi nilai 1 dan sebaliknya piksel dengan derajat keabuan lebih kecil dari nilai batas akan diberi nilai 0. Citra macam ini banyak digunakan dalam pemrosesan citra, misalnya dalam rangka memperoleh tepi bentuk suatu objek [51]. Fungsi dari binerisasi

sendiri adalah untuk memperlancar proses mengenali pola, karena pola akan lebih gampang terdeteksi pada citra yang mengandung lebih sedikit warna.

2.15.2 Citra *Grayscale*

Citra *grayscale* merupakan citra digital yang hanya memiliki satu nilai kanal pada tiap pikselnya, atau bisa dikatakan bagian *red*, *green*, dan *blue* memiliki nilai yang sama. Nilai tersebut digunakan untuk menunjukkan tingkat intensitas. Warna yang dimiliki adalah warna dari hitam, keabuan, dan putih. Tingkat keabuan ini merupakan warna abu dengan bermacam tingkatan dari hitam hingga mendekati putih [52]. Citra skala keabuan mempunyai nilai minimum (biasanya=0) dan nilai maksimum. Banyaknya kemungkinan nilai minimum dan maksimum bergantung pada jumlah bit yang digunakan (umumnya menggunakan 8 bit). Contohnya untuk skala keabuan 4 bit, maka jumlah kemungkinan nilainya adalah $2^4 = 16$, dan nilai maksimumnya adalah $2^4 - 1 = 15$, sedangkan untuk skala keabuan 8 bit, maka jumlah kemungkinan nilainya adalah $2^8 = 256$, dan nilai maksimumnya adalah $2^8 - 1 = 255$.

Secara digital suatu *grayscale image* dapat direpresentasikan dalam bentuk array dua dimensi. Tiap elemen dalam array menunjukkan intensitas (*greylevel*) dari *image* pada posisi koordinat yang bersesuaian. Apabila suatu citra direpresentasikan dalam 8 bit maka berarti pada citra terdapat 28 atau 256 level *grayscale*, (biasanya bernilai 0 – 255), dimana 0 menunjukkan level intensitas paling gelap dan 255 menunjukkan intensitas paling terang. Bisa dikatakan Nilai 0 menyatakan hitam dan nilai 255 menyatakan putih [53].

Pada dasarnya proses ini dilakukan dengan meratakan nilai piksel dari tiga nilai RGB menjadi satu nilai. Namun karena ketiga warna pokok tersebut dianggap tidak seragam dalam hal kemampuan kontribusi pada kecerahan, ada yang berpendapat bahwa cara konversi yang lebih tepat adalah dengan menggunakan Persamaan [54].

$$Y = 0,299R + 0,587G + 0,114B \dots\dots\dots 2.6$$

Di mana Y adalah nilai kecerahan suatu piksel pada citra *grayscale*, dengan persentase 29,9% warna merah, 58,7% dari warna hijau, dan 11,4 dari warna biru.

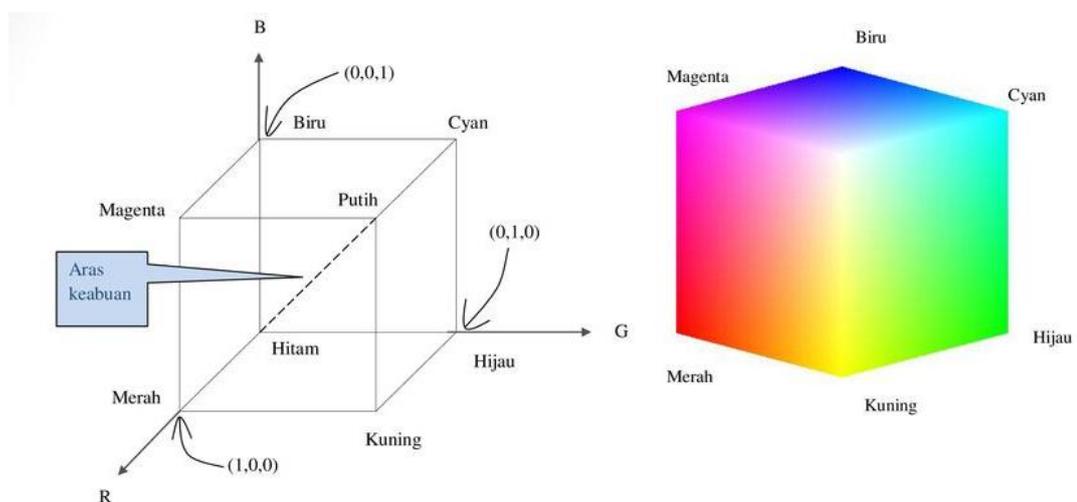
2.15.3 Citra RGB (*Red Green Blue*)

Citra berwarna atau biasa dinamakan citra RGB, merupakan jenis citra yang menyajikan warna dalam bentuk komponen R (*Red*), G (*Green*), B (*Blue*).

Tabel 2. 4 Warna dan nilai penyusun warna

warna	R	G	B
Hijau	255	0	0
Merah	0	255	0
Biru	0	0	255
Hitam	0	0	0
Putih	255	255	255

Setiap komponen warna menggunakan delapan bit (nilainya berkisar antara 0 sampai dengan 255). Dengan demikian kemungkinan warna yang dapat disajikan mencapai $255 \times 255 \times 255$ atau 16.581.375 warna. Tabel 2.3 menunjukkan contoh warna dan nilai R, G, B. Model RGB biasa disajikan dalam bentuk kubus tiga dimensi, dengan warna merah, hijau, biru berada di pojok sumbu (Gambar 2.8). Warna hitam berada pada titik asal dan warna putih berada di sumbu kubus yang bersebrangan.



Gambar 2. 8 Skema ruang warna RGB dalam bentuk kubus [55]

Fitur warna dapat diperoleh melalui perhitungan statistik seperti rerata dan deviasi standar. Sebagai contoh fitur-fitur tersebut dapat digunakan untuk kepentingan identifikasi maupun klasifikasi. Perhitungan dikenakan pada setiap komponen R, G, dan B.

RGB banyak digunakan dalam perancangan hardware seperti peralatan elektronik (televisi, monitor komputer, dan lain-lain) karena kemudahannya dalam perancangan, akan tetapi sebenarnya tidak ideal untuk beberapa aplikasi karena warna merah, hijau, dan biru berkorelasi erat dan sangat sulit diproses untuk beberapa algoritma [52].

2.16 Segmentasi

Teknik untuk membagi atau memisahkan citra ke dalam beberapa daerah (*region*) berdasarkan kemiripan atribut yang dimilikinya disebut dengan segmentasi [52]. Segmentasi juga disebut sebuah proses yang membagi sebuah citra menjadi sejumlah bagian atau objek. Segmentasi bukanlah proses tunggal yang dilakukan dalam pengolahan citra digital. Namun segmentasi merupakan proses yang penting dalam pengolahan citra digital. Pada proses segmentasi objek yang menjadi target akan diambil untuk proses selanjutnya. Ada dua karakteristik nilai derajat kecerahan citra pada Teknik segmentasi ini, yaitu *discontinuity* dan *similarity*. Pada *discontinuity*, citra dipisah pisah berdasarkan perubahan yang mencolok dari derajat kecerahannya, biasanya diterapkan pada deteksi tepi, garis, area dan sisi citra. Sedangkan pada *similarity* citra akan dipisah berdasarkan *thresholding*, *region growing* dan *region splitting* dan *merging*, biasanya diterapkan pada citra yang statis dan dinamis [56].

2.17 Thresholding

Thresholding proses untuk memisahkan foreground dengan background dari suatu citra [57]. Proses *thresholding* dilakukan dengan cara melihat perbedaan intensitas warna dari suatu gambar. *Input* untuk proses *thresholding* ialah gambar. Keluaran dari proses ini adalah citra biner. Citra hasil *thresholding* biasanya

digunakan untuk proses mengenali objek serta ekstraksi fitur. Metode *thresholding* secara umum dibagi menjadi dua, yaitu :

- *Thresholding* global

Thresholding dilakukan dengan mempartisi histogram dengan menggunakan sebuah *threshold* (batas ambang) global T, yang berlaku untuk seluruh bagian pada citra.

- *Thresholding* adaptif

Thresholding dilakukan dengan membagi citra menggunakan beberapa sub citra. Lalu pada setiap sub citra, segmentasi dilakukan dengan menggunakan *threshold* yang berbeda. *Thresholding* diimplementasikan setelah dilakukan proses perbaikan kontras citra menggunakan fungsi *Contrast-Limited Adaptive histogram Equalization* (CLAHE).

Pada proses *thresholding* digunakan sebuah parameter yang disebut dengan *intensity threshold*. Setiap *piksel* yang didapat akan dibandingkan dengan nilai *intensity threshold*, apabila *piksel* gambar lebih tinggi dari *intensity threshold* maka *piksel* akan diset menjadi putih. Sebaliknya, apabila lebih rendah maka akan diset menjadi *piksel* hitam. Dalam hal ini nilai *intensity threshold* yang akan digunakan adalah nilai tengah dari 0-255 yaitu 127. Dalam proses segmentasi menggunakan teknik pengambangan (*Thresholding*) yang akan menghasilkan citra biner [52]. Biasanya T dihitung dengan persamaan:

$$T = \frac{f_{maks} + f_{min}}{2} \dots\dots\dots 2.7$$

Dimana T adalah *threshold* (nilai ambang batas), *f_{maks}* ialah nilai intensitas maksimum pada citra, *f_{min}* adalah nilai intensitas minimum pada citra. Jika *f(x,y)* adalah nilai intensitas *piksel* pada posisi (x,y) maka *piksel* tersebut diganti putih atau hitam tergantung kondisi berikut:

$$F(x,y) = \begin{cases} 255, & \text{jika } f(x,y) \geq T \\ 0, & \text{jika } f(x,y) < T \end{cases} \dots\dots\dots 2.8$$

Jika nilai pada piksel lebih besar sama dengan nilai ambang batas maka nilai pada piksel tersebut menjadi bernilai 255, sedangkan jika nilai pada piksel l lebih kecil dari nilai ambang batas maka nilai pada piksel tersebut menjadi bernilai 0.

2.18 Python

Untuk mengimplementasikan proyek *Deep Learning*, yang mana salah satunya yaitu metode *Convolutional Neural Network (CNN)* diperlukan sebuah bahasa pemrograman yang stabil, fleksibel dan memiliki sekumpulan tools yang tersedia, baik itu berupa pustaka (*library*) maupun kerangka kerja (*framework*). Tujuannya agar saat proses implementasi lebih mudah dan cepat, sehingga tidak memakan banyak waktu maupun tenaga. Saat ini, terdapat bahasa pemrograman yang menawarkan itu semua, yaitu Python.

Python merupakan bahasa pemrograman yang interpretatif dan multiguna. Tidak seperti bahasa lain yang rumit dibaca juga dipahami, Python lebih menekankan keterbacaan kode guna membuat *sintaks* sehingga lebih mudah untuk dipahami.



Gambar 2. 9 Logo Python [58]

Dengan kode yang sederhana dan mudah diimplementasikan, seorang programmer dapat memprioritaskan pengembangan aplikasi yang sebenarnya, daripada sibuk mencari *syntax error* [59].

Python banyak digunakan oleh akademis dan masyarakat umum guna berbagai keperluan, mulai dari membuat program desktop atau situs web, membuat game, hingga keperluan statistik, IoT (*Internet of Things*), analisis, pemrosesan, dan

visualisasi data serta saat ini umum digunakan untuk membuat proyek AI, salah satunya CNN, dengan dukungan *library* dan *framework* yang tepat. Berikut adalah beberapa *library* Python umum yang digunakan di CNN.

1. Tensorflow

TensorFlow adalah *software library* yang dikembangkan oleh Tim Google Brain dalam organisasi penelitian Mesin Cerdas Google, untuk tujuan melakukan pembelajaran mesin dan penelitian jaringan syaraf dalam.

TensorFlow menggabungkan teknik pengoptimalan aljabar komputasi terintegrasi yang menyederhanakan penghitungan banyak rumus, yang merupakan masalah yang memakan waktu. TensorFlow dapat menggunakan informasi tentang objek atau target yang diidentifikasi untuk menjalankan perintah guna membedakan satu objek dari objek lainnya [60].

2. OpenCV

OpenCV Digunakan untuk tujuan pengolahan citra digital (*digital image processing*) dan *computer vision*. Contoh penggunaannya di CNN yaitu membaca data citra (jpg, png, dll), mengubah data citra menjadi sebuah array agar bisa diolah, mengkonversi citra (misal dari RGB ke *grayscale*), dan *resize* citra (misal dari 500 x 500 piksel diubah menjadi 224 x 224 piksel).

3. Sklearn

Sklearn digunakan sebagai tools ML dan model statistik berupa *classification*, *regression*, *clustering* dan *dimensionality reduction*. Contoh penggunaannya di CNN yaitu mengaplikasikan *k-fold cross validation* terhadap dataset, membuat *confusion matrix* dan *classification report*.

4. Numpy

Numpy digunakan untuk memproses/mengolah beragam keperluan array maupun *matrix*. Contoh penggunaannya di CNN yaitu reshape dimensi citra, menghitung rata-rata maupun standar deviasi akurasi model CNN, dan pengolahan berbasis *array/matrix* lainnya.

5. Pandas

Pandass digunakan untuk memanipulasi dan analisis data. Contoh penggunaannya di CNN yaitu menyimpan history pelatihan (berupa *accuracy* dan *loss*) kedalam sebuah format tabel CSV.

6. Matplotlib

Matplotlib digunakan untuk membuat visualisasi data. Contoh penggunaannya di CNN yaitu mem-plot dataset beserta labelnya dan mem-plot hasil *training* maupun *testing* model CNN kedalam sebuah grafik.

7. Seaborn

Seaborn digunakan untuk membuat visualisasi data yang lebih menarik dan informatif berdasarkan Matplotlib. Contoh penggunaannya di CNN yaitu membuat *heatmap* untuk *confusion matrix*, tujuannya agar lebih informatif sehingga mudah untuk dilakukan analisis.

2.19 Roboflow

Roboflow merupakan sebuah platform yang dikhususkan untuk membantu AI *engineer* dalam mengolah dataset untuk pengaplikasian projek *computer vision*.



Gambar 2. 10 Roboflow [61]

Roboflow menyediakan 20 library model yang dapat menjadi pilihan model saat akan membuat projek *computer vision*, yaitu YOLOX, Vision Transformer, YOLOv5, YOLOR, OpenAI Clip, Scaled-YOLOv4, Resnet32, EfficientDet-D0-D7, YOLOv4-tiny, YOLOv4 Darknet, EfficientNet, Detectron2, EfficientDet, Faster R-CNN, YOLO v3 PyTorch, YOLO v3 Keras, MobileNetSSDv2, YOLOv4, PyTorch, MobileNetV2 Classification, dan ResNet32 [62].

2.20 Google Colaboratory

Google Colaboratory merupakan sebuah alat yang mirip dan berdasarkan dari Jupyter Notebooks. Jupyter adalah alat *open source* berbasis *browser* yang mengintegrasikan bahasa pemrograman, *library*, dan alat visualisasi. Notebook Jupyter berfungsi baik secara lokal maupun di *cloud*. Setiap dokumen terdiri dari beberapa sel, di mana setiap sel berisi bahasa skrip atau kode *markdown*, dan *outputnya* disematkan dalam dokumen. Tipe *output* termasuk teks, tabel, grafik, dan grafik. Google Colaboratory (Colab) adalah proyek yang bertujuan untuk menyebarluaskan pendidikan dan penelitian pembelajaran mesin. Notebook Colab didasarkan pada Jupyter dan digunakan sebagai objek Google Documents: notebook dapat dibagikan dan pengguna dapat berkolaborasi pada *notebook* yang sama [63].

Kolaborasi menyediakan *runtime* Python 2 dan 3 yang telah dikonfigurasi sebelumnya dengan pembelajaran mesin dan library AI, seperti TensorFlow, Matplotlib, dan Keras. Mesin virtual di bawah *runtime* (VM) dinonaktifkan setelah jangka waktu tertentu, dan semua data dan konfigurasi pengguna hilang. Namun, notebook memungkinkan untuk mentransfer file dari *hard disk* VM ke akun Google Drive pengguna. Layanan Google ini menyediakan akselerasi GPU *runtime*, juga sepenuhnya dikonfigurasi dengan perangkat lunak yang sebelumnya diuraikan. Infrastruktur Google Colaboratory di-*host* di platform Google Cloud [63].

2.21 Jupyter Notebook

Jupyter Notebook (*file* yang berekstensi *ipynb*) adalah dokumen yang dihasilkan oleh Jupyter Notebook App yang berisikan kode komputer dan *rich text element* seperti paragraf, persamaan matematik, gambar dan tautan (*links*). Jupyter Notebook dikenal sebelumnya sebagai IPython Notebook dan dalam waktu dekat akan berevolusi menjadi Jupyter Lab [64].

2.22 Populasi dan Sampel

Populasi adalah wilayah generalisasi yang terdiri atas subyek atau obyek dengan kualitas dan karakteristik tertentu yang ditetapkan oleh peneliti untuk dipelajari dan ditarik kesimpulan [65]. Sampel adalah bagian dari jumlah dan karakteristik yang dimiliki oleh populasi [65]. Sehingga dapat dikatakan sampel merupakan suatu pengambilan data dari jumlah dan karakteristik yang dimiliki oleh populasi dimana hanya sebagian populasi saja diambil dan diperlakukan untuk menentukan sifat serta ciri yang dikehendaki dari suatu populasi.

Semakin besar jumlah sampel mendekati populasi makin kecil peluang kesalahan generalisasi. Kemudian makin kecil jumlah populasi makin besar kesalahan generalisasi (diberlakukan umum). Semakin kecil toleransi kesalahan, semakin akurat sampel menggambarkan populasi. Pada penelitian dengan batas kesalahan 5% berarti memiliki tingkat akurasi 95%. Pedoman menentukan jumlah sampel menurut pendapat Slovin dapat dilihat pada persamaan 2.9.

$$n = \frac{N}{1+N(e)^2} \dots\dots\dots 2.9$$

Keterangan:

N = ukuran populasi

n = ukuran sampel

e = *margin of error* / batas toleransi kesalahan (5%)

2.23 Confusion Matrix

Menurut Hahn dan Kamber [66] *confusion matrix* dapat diartikan sebagai alat yang berfungsi untuk menganalisis apakah suatu sistem baik dalam mendeteksi *tuple* kelas yang berbeda. Metode tersebut disusun dalam sebuah tabel yang terdiri dari deretan data uji yang diprediksi benar atau salah berdasarkan hasil deteksi sistem (model). Terdapat empat nilai pengukuran yang digunakan untuk mengukur kinerja model yang diuji, yaitu :

1. *True Negative* (TN) : jumlah data negatif yang terdeteksi benar

2. *False Positive* (FP) : data negatif namun terdeteksi sebagai data positif.
3. *True Positive* (TP) : data positif yang terdeteksi benar.
4. *False Negative* (FN) : kebalikan dari *True Positive*, sehingga data positif, namun terdeteksi sebagai data negatif.

Pengukuran kinerja yang dibuat menggunakan basis konsep dari *confusion matrix* seperti pada Tabel 3. Dimana kelas (+) merepresentasikan kelas actual data (manual). Sedangkan kelas (-) merupakan kelas predicted data (otomatis).

Tabel 2. 5 *Confusion matrix*

	Prediksi (+)	Prediksi (-)
Aktual (+)	TP	TN
Aktual (-)	FP	FN

Setelah dilakukan proses perhitungan dengan uji *confusion matrix*, nilai yang didapat akan dilakukan perhitungan lanjutan untuk mendapatkan hasil analisis performa sistem yang dibuat. Berikut persamaan yang digunakan :

1. *Presisi* merupakan perbandingan jumlah data yang relevan yang terambil oleh sistem dengan keseluruhan data yang terambil oleh sistem. perhitungan ketepatan klasifikasi pada jumlah data berlabel positif atau data kelas minoritas yang memang benar secara aktual merupakan kelas positif.

$$Precision = \frac{TP}{TP+FP} \dots\dots\dots 2.10$$

2. *Recall* (disebut juga *sensitivity*) adalah perhitungan ketepatan klasifikasi pada jumlah data positif yang teridentifikasi benar sebagai kelas positif. Didefinisikan juga sebagai perbandingan jumlah data yang relevan yang terambil oleh sistem dengan keseluruhan data relevan yang ada dalam sistem. *F-measure* sendiri adalah *relative importance* antar *presisi* dan *recall*.

$$Recall = \frac{TP}{TP+FN} \dots\dots\dots 2.11$$

$$F - measure = \frac{(1+\beta^2) \times Recall \times Precision}{\beta^2 \times Recall + Precision} \dots\dots\dots 2.12$$

3. *Accuracy* merupakan kedekatan hasil pengukuran (prediksi) dengan nilai yang sebenarnya (aktual). Sebuah sistem pengukuran dapat akurat dan tepat, atau akurat tetapi tidak tepat, atau tepat tetapi tidak akurat atau tidak tepat dan tidak akurat.

$$Akurasi = \frac{TP+TN}{TP+TN+FP+FN} \dots\dots\dots 2.13$$

4. *Specificity* Merupakan kebenaran memprediksi negatif dibandingkan dengan keseluruhan data negatif.

$$Specificity = \frac{TN}{TN+FP} \dots\dots\dots 2.14$$

5. *Error rate* adalah dihitung sebagai jumlah semua prediksi yang salah dibagi dengan jumlah total dataset.

$$Error = \frac{FP}{TP+FP} \dots\dots\dots 2.15$$

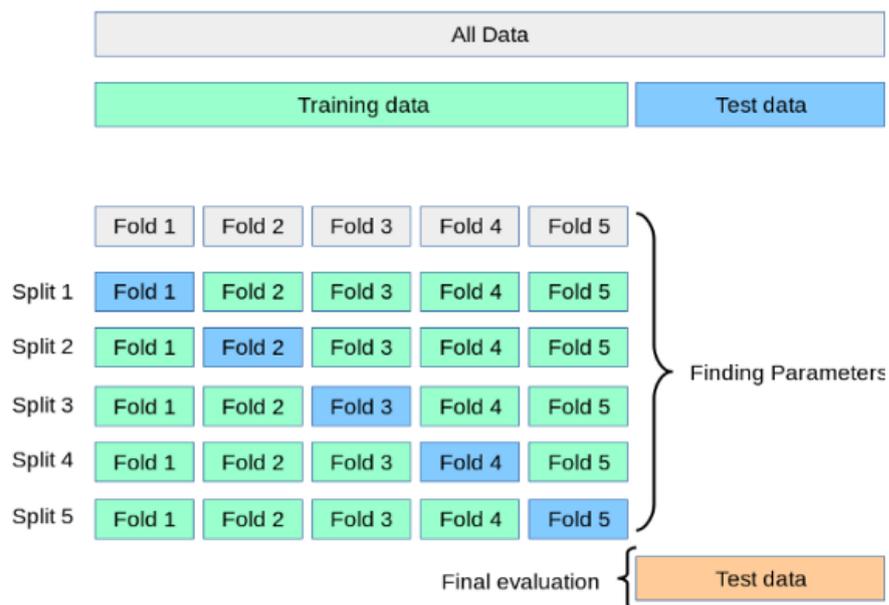
2.24 K-fold Cross Validation

Salah satu metode dari *cross validation* yang sering digunakan adalah *k-fold cross validation* karena metode ini secara umum akan menghasilkan model yang tidak bias. Sebagaimana telah dijelaskan dalam [67], *cross validation* merupakan salah satu metode dalam melakukan validasi model terbaik. Teknik ini akan menguji keefektifan dari model yang dibentuk dengan melakukan penyusunan ulang (*resampling*) pada data untuk membaginya menjadi 2 bagian yaitu data *training* dan data *testing*. Data *training* akan dipakai untuk melatih model sehingga model dapat memahami pola pada data dan untuk melakukan validasi terhadap latihan model tersebut, akan digunakan data *testing* sebagai pengujiannya.

Awalnya, metode ini akan membagi data menjadi k bagian (*folds*) yang sama besar. Nilai k dibebaskan kepada peneliti, tetapi disarankan tidak terlalu besar dan tidak terlalu kecil. Nilai k yang terlalu besar akan menghasilkan model yang tidak

bias, tetapi dapat membuat variansi menjadi besar sehingga dapat memicu terjadinya overfit. Nilai k yang terlalu kecil akan menghasilkan model yang serupa dengan metode *cross validation* biasa yang hanya membagi data menjadi *train – test* saja (dapat memicu terjadinya bias). Nilai k yang biasa digunakan adalah $k = 5$ atau $k = 10$ [68].

Setelah data dibagi menjadi k bagian yang sama besar, $k-1$ bagian akan digunakan sebagai data *training* untuk melatih model dan 1 bagian yang tersisa akan digunakan sebagai data *testing* untuk validasi model. Kemudian menghitung *mean* dari Koefisien determinasi *R-squared* untuk mengetahui apakah terdapat korelasi yang baik antara jumlah piksel dari lebar daun terdeteksi terhadap lebar daunnya, dan menghitung *Mean Absolute Percentage Error* (MAPE) untuk mengukur ketepatan atau akurasi suatu hasil prediksi pada model.



Gambar 2. 11 Ilustrasi *K-fold Cross Validation* [69]

1. Koefisien determinasi (*R-Squared*)

Menurut Agus Widarjono [70] Uji Koefisien Determinasi (*R-Squared*) merupakan jenis pengujian yang digunakan untuk menjelaskan besaran skala variasi dari variabel dependen yang dijelaskan oleh variabel independen. Selain itu, uji koefisien determinasi juga berfungsi untuk melihat seberapa baik garis regresi

yang dimiliki. Koefisiensi determinasi ini mengukur presentase total varian variabel dependen Y yang dijelaskan oleh variabel independen di dalam garis regresi. Nilai *R-squared* mempunyai interval antara 0 sampai 1 ($0 < R\text{-squared} < 1$). Semakin besar *R-squared* (mendekati 1), semakin baik hasil untuk model regresi tersebut dan semakin mendekati 0, maka variabel independen secara keseluruhan tidak dapat menjelaskan variabel dependen [71]. Koefisien determinasi dapat ditentukan dengan rumus :

$$R\text{-squared} = \frac{((n)(\sum XY) - (\sum X)(\sum Y))^2}{[n\sum X^2 - (\sum X)^2][n\sum Y^2 - (\sum Y)^2]} \dots\dots\dots 2.16$$

Keterangan :

R-squared = Koefisien determinasi

n = Jumlah data

Y = Variabel dependen

X = Variabel independen

Tabel 2. 6 Range nilai *R-squared* [72]

Nilai MAPE	Kriteria
> 0.67	Kuat
0.34 – 0.67	Moderat
0.19 – 0.33	Lemah

2. Mean Absolute Percentage Error (MAPE)

Mean Absolute Percentage Error (MAPE) adalah salah satu dari beberapa cara yang digunakan dalam evaluasi untuk mengukur ketepatan atau akurasi suatu hasil prediksi yang sangat umum digunakan. MAPE adalah suatu ukuran ketetapan *relative* yang digunakan untuk menghitung presentase hasil dari penyimpangan peralaman dengan cara menghitung kesalahan absolut pada tiap periode dibagi dengan hasil observasi nyata pada periode tersebut kemudian dihitung rata-ratanya [73]. Persamaan 2.17 digunakan untuk perhitungan mencari eror dari hasil prediksi.

$$\text{MAPE} = \sum_{t=1}^n \left| \frac{y - \hat{y}}{\hat{y}} \right| \times 100\% \dots\dots\dots 2.17$$

Keterangan :

MAPE = *Mean Absolute Percentage Error* (%)

n = Jumlah data

y = Nilai hasil aktual

\hat{y} = Nilai hasil prediksi

Semakin rendah nilai MAPE, maka dapat dikatakan model peramalan memiliki kemampuan yang baik. Nilai MAPE yang baik adalah di bawah 10% [74]. Range nilai untuk MAPE dapat dilihat pada Tabel 2.7.

Tabel 2. 7 *Range* nilai MAPE [74]

Nilai MAPE	Kriteria
<10%	Sangat Baik
10-20%	Baik
20-50%	Cukup
>50%	Buruk

BAB III

METODELOGI PENELITIAN

3.1 Tempat dan Waktu Penelitian

Penelitian ini dimulai pada Juni 2022 sampai dengan Desember 2022. Penelitian ini dilakukan di Laboratorium Terpadu Jurusan Teknik Elektro, Fakultas Teknik, Universitas Lampung.

3.2 Alat dan Bahan

Alat dan bahan yang digunakan pada penelitian penelitian ini adalah sebagai berikut :

1. Satu unit laptop Dell Latitude E5470 dengan spesifikasi *processor* Intel Core i5-6300U, memori RAM 8GB dan sistem operasi Windows 10.
2. Perangkat Lunak Python 3.10 (64bit).
3. Citra udara sebagai dataset.
4. Google Colab sebagai tempat untuk *training* dan *testing* dataset dengan keluaran model algoritma YOLO.
5. *Software* Visual Studio Code dan Jupyter Notebook sebagai *text editor* dan *running script program* dalam pengolahan citra dan regresi linier.

3.3 Prosedur Penelitian

Metode penelitian dalam pengerjaan penelitian ini adalah sebagai berikut:

1. Studi Literatur

Studi literatur merupakan tahapan awal penelitian untuk mempelajari suatu pustaka dasar teori yang digunakan untuk menunjang penelitian proposal. Hal

ini dilakukan untuk mendapatkan pemahaman tentang konsep, metode, yang berkaitan dengan *computer vision*, *machine learning*, *deep learning* dan regresi linier yang meliputi kebutuhan data awal untuk dapat dilakukan pengolahan, *tools* apa saja yang dibutuhkan, bagaimana melakukan metode *computer vision*, *machine learning*, *deep learning*, pengolahan citra dan regresi linier. Penelitian ini menggunakan dasar teori berupa buku, jurnal, dan penelitian terdahulu. Studi literatur bertujuan untuk mempelajari materi-materi yang terkait dengan topik yang dibahas dalam penelitian, yaitu mengenai *artificial intelligence*, *machine learning*, *deep learning*, *computer vision*, pengolahan citra dan regresi linier. Materi tersebut didapatkan dari buku, jurnal dan referensi dari website yang dapat dipertanggung jawabkan informasinya.

2. Pengumpulan Data

Kegiatan pengumpulan data dilakukan dengan mengumpulkan data-data yang diperlukan dalam pembuatan desain program, data citra lahan nanas.

3. Pembuatan Program

Pembuatan program ini untuk mengimplementasikan hasil studi literatur dan studi bimbingan yang dilakukan dalam menggunakan bahasa pemrograman Python yang sebelumnya data-data yang diperlukan sudah terkumpul.

4. Pembuatan Laporan

Pembuatan laporan ini berupa penulisan dari hasil yang sudah didapatkan dan dapat dipertanggungjawabkan terhadap penelitian yang sudah dilakukan. Pada pembuatan laporan ini terdapat dua bagian, yaitu laporan proposal untuk seminar proposal dan laporan akhir yang digunakan untuk seminar hasil.

5. Interpretasi dan Pembahasan

Setelah mendapatkan model yang sesuai dengan tingkat akurasi yang tinggi maka dilakukan interpretasi hasil dan pembahasan.

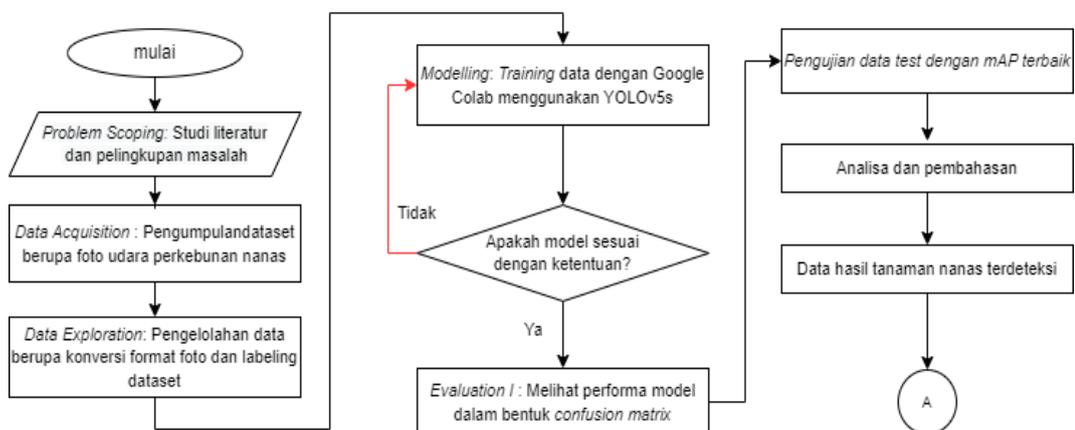
6. Kesimpulan dan saran

3.4 Perancangan Program

Perancangan program secara keseluruhan dilakukan dengan *software* Visual Studio Code, Jupyter Notebook, dan Google Colab dengan referensi dari berbagai sumber seperti *GitHub*, *StackOverFlow*, *Youtube*, dan lainnya. Perancangan program terdiri menjadi 3 bagian, yaitu perancangan program dengan metode YOLOv5, perancangan program pengolahan citra dan perancangan program regresi linier. Program dengan metode YOLOv5 bertujuan untuk mendeteksi dan menghitung objek yaitu tanaman nanas dengan output total objek terdeteksi sebagai jumlah populasi (data acuan untuk menghitung berapa objek sampel lebar daun yang harus dihitung) untuk sistem pengolahan citra. Program pengolahan citra bertujuan untuk mengolah citra menjadi citra biner agar mendapatkan keluaran berupa nilai lebar piksel putih yang digunakan sebagai variabel pada metode regresi untuk menentukan lebar daun terprediksi dalam satuan panjang centimeter. Selanjutnya hasil dari data lebar daun yang terprediksi akan dikonversikan ke dalam biomassa tanaman nanas.

3.4.1 Metode YOLOv5

Alur kerja dan metode dalam perancangan program metode YOLOv5 menggunakan *AI Project Cycle* yang merupakan tahapan atau metode kerja untuk membuat suatu proyek AI. *AI Project Cycle* untuk penelitian ini digambarkan dalam diagram alir pada Gambar 3.1 berikut.



Gambar 3. 1 Diagram Alir *AI Project Cycle* untuk metode YOLOv5

Berdasarkan Gambar 3.1 tahapan perancangan program metode YOLOv5 menggunakan AI Project Cycle pada penelitian ini adalah sebagai berikut:

1. *Problem Scoping*

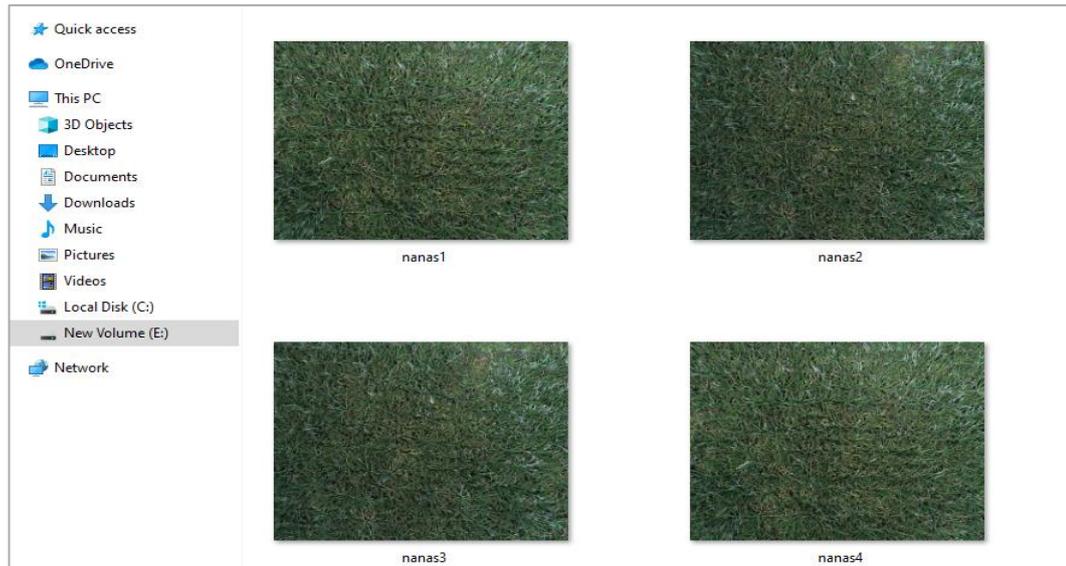
Proses ini dilakukan untuk mengidentifikasi masalah yang ada untuk membuat suatu solusi dengan menggunakan AI. Dalam problem scoping, dilakukan perincian dalam 4W, yaitu *What*, *Who*, *Why*, dan *Where* seperti pada tabel 3.1 berikut ini [75]. Proses ini dilakukan untuk mengidentifikasi masalah yang ada dan membuat solusi terhadap pemecahan masalah dengan AI. Pada tahap ini dilakukan mencari, mempelajari, serta memahami materi dari berbagai literatur yang berkaitan dengan masalah yang ada. Materi dan literatur yang dipelajari berasal dari berbagai sumber referensi atau sumber-sumber ilmiah lainnya seperti penelitian-penelitian sebelumnya, jurnal, artikel dan buku/*e-book*.

Tabel 3. 1 *Problem Scoping*

4W	Rincian
<i>What</i>	Cara <i>forcing</i> berdasarkan usia tanaman saat 12-14 bulan dinilai kurang efektif dalam mengefisiensi waktu panen
<i>Who</i>	Petani dan Pabrik Perkebunan nanas
<i>Where</i>	Perkebunan Nanas
<i>Why</i>	Selama ini PT Great Giant Pineapple menggunakan cara <i>forcing</i> berdasarkan usia tanaman saat 12-14 bulan. Cara <i>forcing</i> pada tanaman nanas dinilai lebih efektif jika menilai dari segi berat tanaman, agar mendapatkan waktu <i>forcing</i> yang tepat. Berat tanaman nanas yang baik untuk <i>forcing</i> adalah 2,3 - 2,5 kg. Karenanya diperlukan sistem untuk dapat mengestimasi berat tanaman nanas guna mengefisiensi waktu panen

2. *Data Acquisition*

Proses ini dilakukan untuk pengambilan dataset berupa foto udara lahan perkebunan nanas yang diperlukan untuk melatih model YOLOv5. Penelitian ini menggunakan foto udara yang diambil di perkebunan nanas PT. Great Giant Pineapple atas izin perusahaan. Data yang diambil terdiri dari satu kelas objek yaitu 'nanas'.

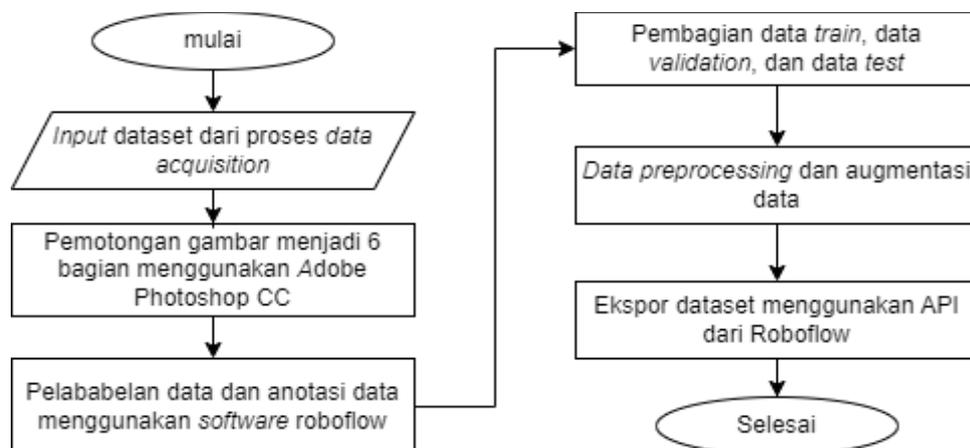


Gambar 3. 2 Dataset lahan perkebunan nanas

Data yang disiapkan berupa foto udara tanaman tebu dengan ukuran lahan 150 m^2 . Total foto udara yang akan diolah sebanyak 4 foto udara seperti tampak pada gambar 3.2.

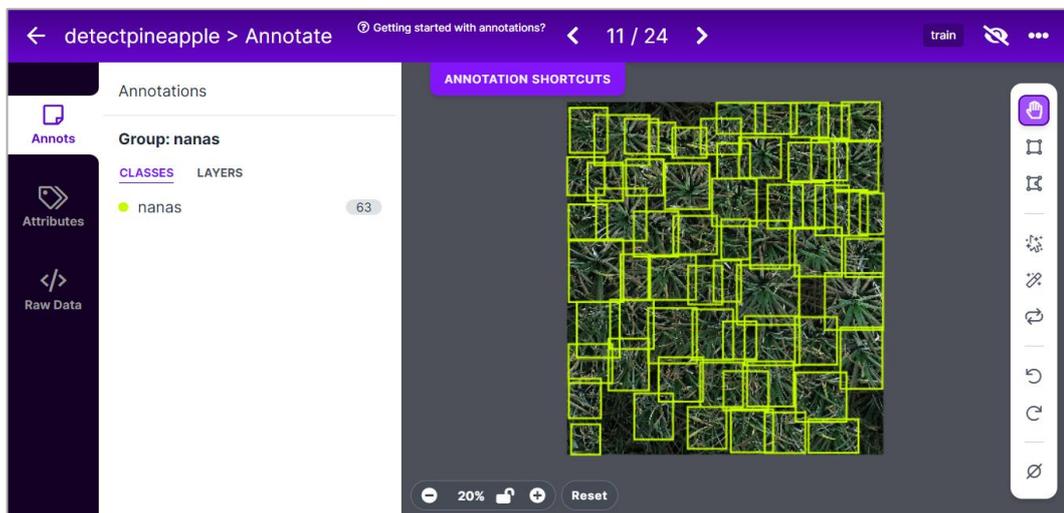
3. *Data Exploration*

Proses ini dilakukan untuk mengolah dataset yang telah terkumpul. Pengolahan foto dilakukan agar format setiap foto sesuai dengan prasyarat algoritma pendeteksian yang dibuat. Alur kerja pada *data exploration* dapat dilihat pada Gambar 3.3.



Gambar 3. 3 Diagram alir *data exploration*

Tahapan pertama pada *data exploration* memotong citra menjadi 6 bagian menggunakan *software* Adobe Photoshop CC. Hal ini dilakukan karena keterbatasan GPU yang dimiliki, karena jika ukuran terlalu besar akan mempersulit *training* dan membebani GPU. Dari hasil ini maka diperoleh 24 potongan gambar yang akan menjadi dataset. Kemudian pada 24 gambar dataset akan dilakukan proses pelabelan dan anotasi dengan *software* Roboflow. Pelabelan data dilakukan untuk memberikan nama atau label dari objek yang dianotasi. Anotasi data dilakukan untuk memberikan kotak penanda pada objek di suatu gambar yang biasa disebut dengan *bounding box* seperti pada gambar 3.5.



Gambar 3. 4 Labeling objek nanas menggunakan *software* Roboflow

Selain persiapan dataset dengan 24 gambar sebelumnya untuk membuat model, disiapkan juga data *test* berupa 1 citra asli berukuran 4864 x 3648 piksel untuk menjadi data tes. Setelah proses labeling selesai dilakukan, dataset dibagi menjadi 2 dengan ratio 70% data *train*, 30% data *validation* [76]. Sehingga total data dari masing-masing set yaitu 16 untuk *train* set, 8 untuk *validation* set.

Tahapan selanjutnya pada *data exploration* adalah tahap *preprocessing data* dan augmentasi data menggunakan *software* Roboflow. Detail *preprocessing data* dan augmentasi data yang digunakan dapat dilihat pada Gambar 3.5.

PREPROCESSING	Auto-Orient: Applied Resize: Stretch to 640×640 Filter Null: Require all images to contain annotations.
AUGMENTATIONS	Outputs per training example: 3 Flip: Horizontal 90° Rotate: Clockwise, Counter-Clockwise, Upside Down Bounding Box: Flip: Horizontal Bounding Box: 90° Rotate: Clockwise, Counter-Clockwise, Upside Down

Gambar 3. 5 Rincian *preprocessing* data dan augmentasi data

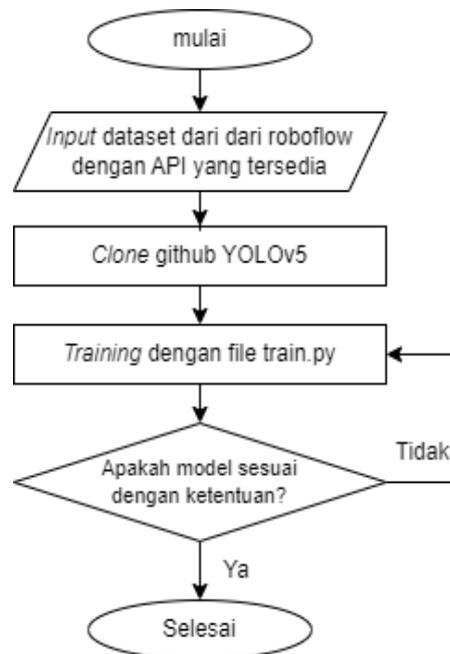
Tahap *preprocessing* data dilakukan untuk memastikan pembelajaran dan inferensi terjadi pada properti gambar yang sama. *Auto-Orient* digunakan agar menghapus gambar dari data EXIF sehingga gambar ditampilkan dengan cara yang sama seperti disimpan di disk. Data diatur ukurannya menjadi 640 x 640 piksel dengan mempertimbangkan dari resolusi gambar yang terkumpul dan waktu latih model, serta melihat dari *pretrained* model YOLOv5 yang melakukan *training* sebelumnya dengan ukuran gambar 640 piksel. *Filter null* digunakan untuk mengeliminasi data yang telah ditandai sebagai null atau tanpa objek pada suatu gambar agar tidak masuk ke dalam proses latih model.

Tahap augmentasi data dilakukan untuk mengatasi *overfitting* pada model dan juga menambah akurasi inferensi karena tahap ini membuat dataset menjadi lebih banyak dengan variasi yang berbeda-beda. Augmentasi data menggunakan 2 variasi yaitu Flip dan Rotation 90° dengan detail masing-masing tertera pada Gambar 3.5. Total dataset yang didapatkan sebesar 170 foto dengan pembagian masing-masing set sebesar 147 *train set* dan 23 *validation set*.

4. Modelling

Tahapan *modelling* adalah proses dimana *dataset* yang sudah dibuat akan dilatih untuk membuat sebuah model. Pelatihan model dilakukan dengan menggunakan *pretrained* model YOLOv5. *Pretrained* model digunakan agar waktu pelatihan lebih cepat dan performa model yang cukup baik untuk deteksi. Pelatihan *dataset*

ini dilakukan menggunakan Google Colab. Proses *modelling* dapat dilihat pada Gambar 3.6.



Gambar 3. 6 Alur proses tahap modelling

Dataset yang telah diolah sebelumnya akan diunduh melalui *tools* Google Colab. Dataset diunduh dengan menggunakan API yang disediakan oleh Roboflow. Model dan *source code* diambil dari situs resmi YOLOv5 melalui Github. Setelah dataset dan model YOLOv5 masuk ke Google Colab, maka selanjutnya dilakukan *training* model dengan menggunakan code `train.py`. Karena YOLOv5 ditulis dalam bahasa pemrograman Python maka file tersebut memiliki ekstensi `*.py`. Dalam melakukan pelatihan model, ada beberapa konfigurasi yang diatur seperti *image size*, *batch*, dan *epoch*. Pada tahap ini digunakan metode ‘*trial and error*’ untuk setiap konfigurasi skema *training* hingga didapatkan konfigurasi dengan skor akurasi tinggi. Metrik evaluasi yang diamati untuk menganalisis performa model adalah *Confusion Matrix*, *Precision*, *Recall*, mAP, loss, dan akurasi.

5. Evaluation

Dalam melakukan analisis performa model yang telah dilatih, dibutuhkan sebuah metrik evaluasi. Metrik evaluasi yang diamati untuk menganalisis performa model adalah *confusion matrix*, *precision*, *recall*, mAP, dan akurasi.

6. Pengujian dataset

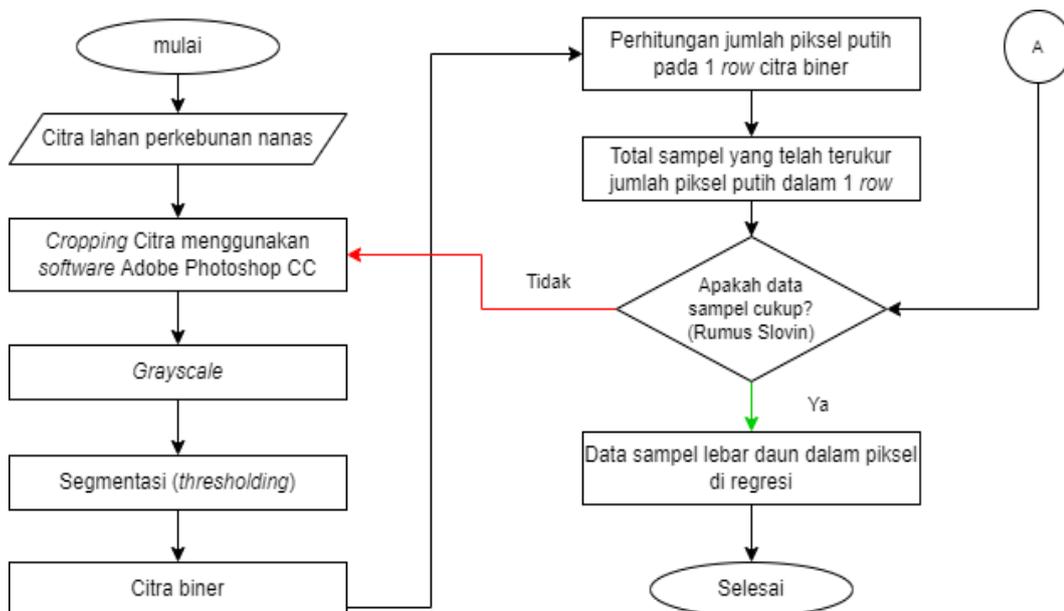
Pada proses ini dilakukan pengujian pada data *testing* dengan menggunakan model deteksi objek yang telah dilatih sebelumnya dengan menghasilkan mAP terbaik. Pengujian ini menggunakan parameter evaluasi berupa *precision*, *recall*, dan akurasi.

7. Analisa dan Pembahasan

Pada proses ini dilakukan analisa pada hasil uji yang didapatkan. Setelah dilakukan uji deteksi, maka data hasil prediksi akan dibandingkan dengan data hasil label. Kemudian Data yang didapat akan diolah untuk dihitung dan diproses sebagai jumlah populasi dari penelitian ini.

3.4.2 Pengolahan Citra

Citra foto udara diolah menggunakan teknik citra digital yang meliputi *cropping* citra, *grayscale*, segmentasi, *thresholding* hingga dapat dihitung jumlah piksel putih dalam satu baris. Diagram alir pengolahan citra pada penelitian ini ditunjukkan oleh Gambar 3.7.

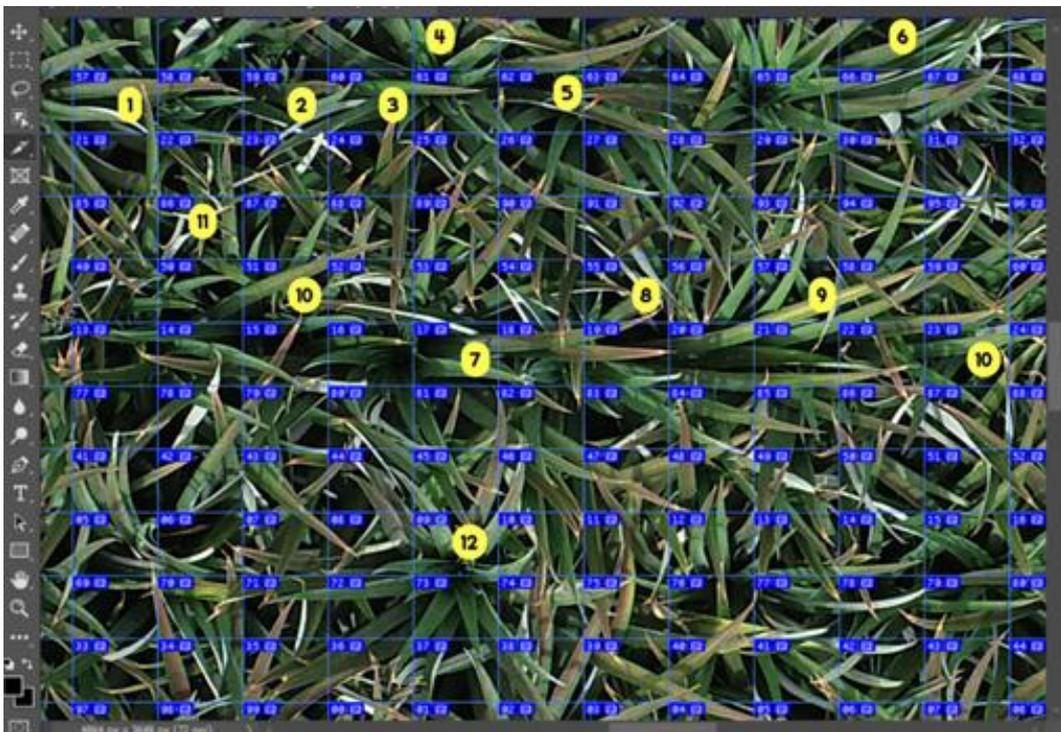


Gambar 3. 7 Diagram alir pengolahan citra

Berdasarkan Gambar 3.7 tahapan untuk mengolah citra pada penelitian ini adalah sebagai berikut:

1. *Cropping* citra

Pemotongan atau *Cropping* citra dilakukan dengan prosedur membagi citra dari ukuran yang asli 4864x3648 px menjadi 64x64 bagian (57x76 px per citra) menggunakan *software* Adobe Photoshop CC. Kemudian citra dipilih secara manual untuk menentukan mana citra yang mewakili 1 daun per tanaman dan dapat diukur lebar daunnya seperti yang ditunjukkan pada Gambar 3.8.



Gambar 3. 8 Memilih citra yang akan dihitung lebar daunnya

2. *Grayscale*

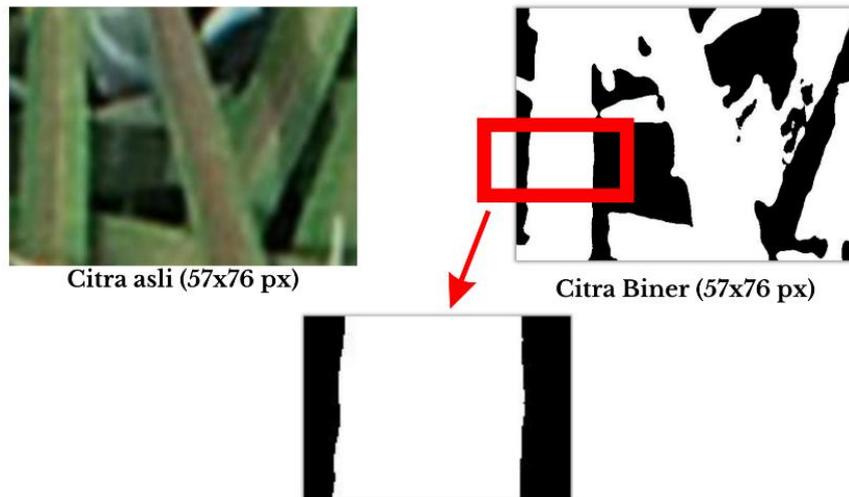
Pada tahapan *grayscale*, citra *cropping* yang sudah dipilih secara manual diubah menjadi citra putih dan gradiasi warna hitam yaitu dengan cara menghitung rata-rata nilai intensitas R, G, dan B dari setiap nilai piksel citra berwarna sehingga didapatkan satu nilai intensitas pada setiap piksel, dengan kata lain nilai $Red=Green=Blue$.

3. Segmentasi (*thresholding*)

Segmentasi dilakukan menggunakan metode pengambangan atau *thresholding*. Proses tersebut dilakukan dengan cara mengatur nilai *threshold* (ambang) pada bahasa pemrograman Python. Tujuannya adalah untuk memisahkan antara objek daun nanas dengan *background*.

4. Citra biner

Hasil dari *thresholding* adalah citra biner. Pada citra biner ini dilakukan *cropping* menggunakan *software* Adobe Photoshop CC yaitu dengan hanya memilih objek yang ada daunnya saja. Pada Gambar 3.9 menggunakan nilai *threshold* 70.



Gambar 3. 9 *Cropping* pada citra biner

5. Perhitungan jumlah piksel putih pada satu *row* citra biner dilakukan dengan bahasa pemograman Python dengan IDE Jupyter Notebook. Selanjutnya merekapitulasi jumlah citra biner yang telah dihitung jumlah piksel putihnya pada satu baris yang akan digunakan sebagai sampel (n) untuk menentukan apakah data ini dapat sudah cukup untuk dilakukan regresi atau belum.

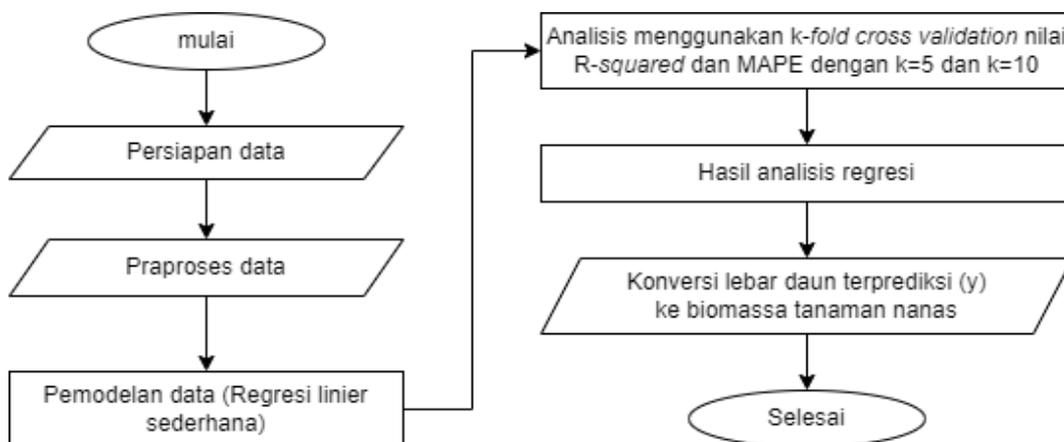
3.4.3 Populasi dan Sampel

Pada penelitian ini, populasi yang digunakan adalah jumlah objek hasil deteksi YOLOv5 dan sampel pada penelitian ini adalah jumlah citra biner yang telah

dihitung jumlah piksel putihnya dalam satu baris (*row*). Penentuan jumlah sampel yang digunakan dalam penelitian ini menggunakan rumus Slovin.

3.4.4 Regresi Linier

Diagram alir untuk regresi linier ditunjukkan pada Gambar 3.10. Regresi linier ini bertujuan untuk memprediksi lebar daun nanas dalam satuan panjang centimeter (cm) dan mengkonversikannya kedalam biomassa tanaman nanas.



Gambar 3. 10 Diagram alir regresi linier dan konversi lebar daun ke biomassa

Berdasarkan Gambar 3.10 tahapan regresi linier pada penelitian ini adalah sebagai berikut:

1. Persiapan data

Data yang digunakan untuk regresi linier adalah data sampel berupa lebar daun terprediksi dalam satuan piksel dan data lebar daun sebenarnya yang telah dihitung secara manual di lapangan.

2. Praproses data

Praproses data dimulai dengan menentukan atribut data yang digunakan dalam regresi linier yaitu lebar daun terprediksi (piksel) sebagai X (variabel independen) dan data lebar daun sebenarnya sebagai Y (variabel dependen).

3. Pemodelan data

Pemodelan data dilakukan dengan regresi linier sederhana untuk melihat hubungan keterkaitan antara variabel X dan variabel Y pada data yang akan

diprediksi sebelum masuk dalam proses analisis data menggunakan k-fold cross validation. Regresi linier dapat dihitung dengan $y = a + bX$.

4. Analisis data *k-fold cross validation*

K-fold cross validation digunakan menilai kinerja proses metode regresi linier dengan membagi sampel data secara acak dan mengelompokkan data tersebut yang dalam penelitian ini sebanyak nilai 5-fold dan 10-fold dengan menghitung nilai *R-squared* dan *Mean Absolute Percentage Error (MAPE)*.

Hasil dari regresi linier adalah nilai y terprediksi (lebar daun), jika y terprediksi memiliki *range* lebar 6-7 cm maka dapat dikonversikan ke biomassa tanaman nanas 2.3-2.5 kg.

3.5 Analisis Data

Analisis data dilakukan untuk mengukur kinerja yang diperoleh dari model selama pelatihan pada data sebelumnya. Analisis data yang digunakan dalam penelitian ini adalah uji *confusion matrix* dan *K-fold cross validation* dengan menghitung nilai *R-squared* dan MAPE.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Kesimpulan dari penelitian ini adalah:

1. Penelitian ini bertujuan membangun sebuah metode yang memprediksi ukuran lebar daun tanaman nanas. Metode yang dibangun terdiri dari metode YOLOv5, pengolahan citra untuk ekstraksi fitur jumlah piksel dari objek pada posisi tampak atas dan regresi linier untuk memprediksi lebar daun pada objek.
2. Pengujian pada regresi linier menggunakan *k-fold cross validation* dengan *5-fold* menghasilkan nilai rata-rata *R-squared* 85% dan MAPE 0.94%. Pengujian *10-fold* menghasilkan nilai rata-rata *R-squared* 81% dan nilai rata-rata MAPE 0.95%.
3. Hasil dari prediksi regresi linier menghasilkan sebanyak 180 helai daun yang memiliki lebar 6-7 cm dari 216 total seluruh data hasil terprediksi dengan persentase 83%. Sehingga tanaman nanas pada citra dapat dilakukan *forcing* karena biomassa tanaman nanas pada citra berkisar 2,3 – 2,5 kg.

5.2 Saran

Saran untuk penelitian ini adalah menggunakan lebih banyak dataset yang digunakan untuk mendapatkan hasil model yang lebih baik, menambahkan jumlah *epoch* pada saat *training* data sehingga menghasilkan hasil akurasi yang lebih tinggi dan menggunakan spesifikasi perangkat yang lebih tinggi yaitu dengan menggunakan komputer dengan *Random Access Memory* (RAM) yang tinggi dan

menggunakan *Graphics Processing Unit* (GPU) untuk mempercepat proses *training*.

DAFTAR PUSTAKA

- [1] K. P. RI, *Outlook Nanas Komoditas Pertanian Sub Sektor Hortikultura*. Jakarta: Kementerian Pertanian RI, 2016.
- [2] Badan Pusat Statistika RI, *Statistik Tanaman Buah-buahan dan Sayuran Tahunan Indonesia*. Jakarta: Badan Pusat Statistika RI., 2019.
- [3] Monavia Ayu Rizaty, “Nanas Jadi Komoditas Buah Unggulan dengan Volume Ekspor Tertinggi,” *Databoks*, 2021. <https://databoks.katadata.co.id/datapublish/2021/03/12/nanas-jadi-komoditas-buah-unggulan-dengan-volume-ekspor-tertinggi> (accessed Jan. 04, 2022).
- [4] Wahyu Pamungkas, “Nanas Lampung Kuasai Pasar Dunia,” *Lampost.co*, 2019. <https://m.lampost.co/berita-nanas-lampung-kuasai-pasar-dunia.html> (accessed Jan. 03, 2022).
- [5] Z. Rabi, F. U. Maigari, U. Lawan, and Z. G. Mukhtar, “Pineapple Waste Utilization as a Sustainable Means of Waste Management BT - Sustainable Technologies for the Management of Agricultural Wastes,” Z. A. Zakaria, Ed. Singapore: Springer Singapore, 2018, pp. 143–154.
- [6] A. N. Putra *et al.*, “Pineapple biomass estimation using unmanned aerial vehicle in various forcing stage: Vegetation index approach from ultra-high-resolution image,” *Smart Agric. Technol.*, vol. 1, no. 1, p. 100025, 2021, doi: 10.1016/j.atech.2021.100025.
- [7] I. Nurhabib, “Pengenalan dan perhitungan pohon kelapa sawit dengan deep learning menggunakan citra satelit,” INSTITUT PERTANIAN BOGOR, 2021.
- [8] Dr. Purba Daru Kusuma, “Definisi Machine Learning,” in *Machine Learning Teori, Program, Dan Studi Kasus*, 1st ed., Sleman, Yogyakarta: Deepublish, 2020.
- [9] H. Fitriyah, “Pengukuran Panjang-Berat Ikan dan Sayuran pada Budikdamber (Budi Daya Ikan dalam Ember) Menggunakan Visi Komputer dan Regresi Linier,” *J. SISKOM-KB (Sistem Komput. dan Kecerdasan Buatan)*, vol. 4, no. 1, pp. 8–14, 2020, doi: 10.47970/siskom-kb.v4i1.166.
- [10] R. Islamadina, N. Pramita, F. Arnia, K. Munadi, and T. M. Iqbal, “Pengukuran Badan Ikan Berupa Estimasi Panjang, Lebar, dan Tinggi Berdasarkan Visual Capture,” *J. Nas. Tek. Elektro dan Teknol. Inf.*, vol. 7, no. 1, 2018, doi: 10.22146/jnteti.v7i1.401.

- [11] R. Wan Nurazwin Syazwani, H. Muhammad Asraf, M. A. Megat Syahirul Amin, and K. A. Nur Dalila, "Automated image identification, detection and fruit counting of top-view pineapple crown using machine learning," *Alexandria Eng. J.*, vol. 61, no. 2, pp. 1265–1276, 2022, doi: 10.1016/j.aej.2021.06.053.
- [12] Z. Guan, A. Abd-Elrahman, Z. Fan, V. M. Whitaker, and B. Wilkinson, "Modeling strawberry biomass and leaf area using object-based analysis of high-resolution images," *ISPRS J. Photogramm. Remote Sens.*, vol. 163, no. September 2019, pp. 171–186, 2020, doi: 10.1016/j.isprsjrs.2020.02.021.
- [13] F. T. A. Muhammad Alfin Jimly Asshiddiqie, Basuki Rahmat, "Deteksi Tanaman Tebu Pada Lahan Pertanian Menggunakan Metode Convolutional Neural Network," *Inform. dan Sist. Inf.*, vol. 1, no. 1, pp. 229–237, 2020.
- [14] F. Ren, Y. Zhang, X. Liu, Y. Zhang, Y. Liu, and F. Zhang, "Identification of Plant Stomata Based on YOLO v5 Deep Learning Model," *ACM International Conference Proceeding Series*. pp. 78–83, 2021, doi: 10.1145/3507548.3507560.
- [15] M. K. Vishal *et al.*, "Leaf Counting in Rice (*Oryza Sativa* L.) Using Object Detection: A Deep Learning Approach," *Int. Geosci. Remote Sens. Symp.*, pp. 5286–5289, 2020, doi: 10.1109/IGARSS39084.2020.9324153.
- [16] J. C. Suárez Salazar, L. M. Melgarejo, E. H. Durán Bautista, J. A. Di Rienzo, and F. Casanoves, "Non-destructive estimation of the leaf weight and leaf area in cacao (*Theobroma cacao* L.)," *Sci. Hortic. (Amsterdam)*, vol. 229, no. August 2017, pp. 19–24, 2018, doi: 10.1016/j.scienta.2017.10.034.
- [17] M. A. S. Putra, "Perancangan Aplikasi Perhitungan Jumlah Pohon Mangga Berbasis Deep Learning Memanfaatkan Low-Altituderemote Sensing," UNIVERSITAS JEMBER, Jember, 2020.
- [18] C. Gupta, V. K. Tewari, R. Machavaram, and P. Shrivastava, "An image processing approach for measurement of chili plant height and width under field conditions," *J. Saudi Soc. Agric. Sci.*, vol. 21, no. 3, pp. 171–179, 2022, doi: 10.1016/j.jssas.2021.07.007.
- [19] C. J. Diji, "Electricity Production From Biomass In Nigeria:Options, Prospects And Challenges," 2013.
- [20] Lembaga Perindustrian Nanas Malaysia, "Morfologi Pohon Nanas." https://www.facebook.com/lpnm.gov/posts/morfologi-pokok-nanassecara-lazimnya-terdapat-4-bahagian-yang-boleh-dijadikan-be/3423602691031017/?locale2=id_ID&paipv=0&eav=AfbS2XPKEwAcnY0zTcC3oIcGDKd_kojvW3uX1ua9SZ2hWTmy8PjOuDoTPHWHIBPBi_I&_rdr (accessed Apr. 01, 2022).
- [21] D. P. Bartholomew, R. E. Paull, and K. G. Rohrbach, *The Pineapple. Botany, Production and Uses. University of Hawaii, Honolulu*. CABI Publishing, EE. UU, 2003.
- [22] A. Suryanto, Sitawati, A. Noor, E. E. Nurlaelih, and D. R. R. Damaiyanti, "PEMBERDAYAAN KELOMPOK TANI UNTUK PERSIAPAN BAHAN BAKU INDUSTRI NANAS (*Ananas comosus* (L) Merr.) DI

KECAMATAN NGANCAR DAN PLOSOKLATEN KABUPATEN KEDIRI,” *J. Din. Pengabd.*, vol. 6, no. 1 SE-, pp. 1–10, Nov. 2020, doi: 10.20956/jdp.v6i1.11499.

- [23] A. S. Pratikno, “Implementasi Artificial Intelligence dalam Memetakan Karakteristik, Kompetensi, dan Perkembangan Psikologi Siswa Sekolah Dasar Melalui Platform Offline Conference,” *Isbn*, no. September 2017, pp. 18–27, 2018.
- [24] I. Cholissodin, A. A. Soebroto, U. Hasanah, and Y. I. Febiola, *AI, Machine Learning & Deep Learning*. Fakultas Ilmu Komputer, Universitas Brawijaya, Malang, 2020.
- [25] K. P. Danukusumo, “Implementasi deep learning menggunakan convolutional neural network untuk klasifikasi citra candi berbasis GPU.” UAJY, 2017.
- [26] L. Deng and D. Yu, *Deep learning: Methods and applications*, vol. 7, no. 3–4. 2013.
- [27] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.
- [28] W. Andriyani, “KORELASI ANTARA ARTIFICIAL INTELLIGENCE, MACHINE LEARNING DAN DEEP LEARNING,” *Algoritma*, 2020. <https://algoritma.blog/artificial-intelligence-deep-learning/> (accessed Apr. 28, 2022).
- [29] H. Mooij, Germonda and Bagulho, Ines and Huisman, “Automatic segmentation of prostate zones,” 2018, doi: 10.48550/ARXIV.1806.07146.
- [30] Y. Lecun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015, doi: 10.1038/nature14539.
- [31] A. L. Katole, K. P. Yellapragada, A. K. Bedi, S. S. Kalra, and M. Siva Chaitanya, “Hierarchical Deep Learning Architecture for 10K Objects Classification,” pp. 77–93, 2015, doi: 10.5121/csit.2015.51408.
- [32] H. Lawaniya, “Computer Vision,” *IET Comput. Vis.*, May 2020.
- [33] Adhy Wiranto, “Tentang Computer Vision,” *Dev Comunnity*. <https://dev.to/adhywiranto44/tentang-computer-vision-1db4> (accessed Jul. 01, 2022).
- [34] Z. Q. Zhao, P. Zheng, S. T. Xu, and X. Wu, “Object Detection with Deep Learning: A Review,” *IEEE Trans. Neural Networks Learn. Syst.*, vol. 30, no. 11, pp. 3212–3232, 2019, doi: 10.1109/TNNLS.2018.2876865.
- [35] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016, vol. 2016-Decem, pp. 779–788, doi: 10.1109/CVPR.2016.91.

- [36] D. Thuan, "Evolution of Yolo algorithm and Yolov5: The State-of-the-Art object detection algorithm," 2021.
- [37] Z. Li, X. Tian, X. Liu, Y. Liu, and X. Shi, "A two-stage industrial defect detection framework based on improved-yolov5 and optimized-inception-resnetv2 models," *Appl. Sci.*, vol. 12, no. 2, p. 834, 2022.
- [38] X. Zhu, S. Lyu, X. Wang, and Q. Zhao, "TPH-YOLOv5: Improved YOLOv5 based on transformer prediction head for object detection on drone-captured scenarios," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 2778–2788.
- [39] U. Nepal and H. Eslamiat, "Comparing YOLOv3, YOLOv4 and YOLOv5 for Autonomous Landing Spot Detection in Faulty UAVs," *Sensors*, vol. 22, no. 2, 2022, doi: 10.3390/s22020464.
- [40] Ultralytics, "YOLOv5," *Github*. <https://github.com/ultralytics/yolov5> (accessed May 24, 2022).
- [41] G. Jocher, "YOLOv5 Documentation," *Ultralytics*. <https://docs.ultralytics.com/> (accessed Jul. 25, 2022).
- [42] T. Khotimah and R. Nindiyasari, "Forecasting Dengan Metode Regresi Linier Pada Sistem Penunjang Keputusan Untuk Memprediksi Jumlah Penjualan Batik (Studi Kasus Kub Sarwo Endah Batik Tulis Lasem)," *J. Mantik Penusa*, vol. 1, no. 1, pp. 71–92, 2017.
- [43] N. Kusumawati, F. Marisa, and I. D. Wijaya, "Prediksi Kurs Rupiah Terhadap Dollar Amerika Dengan Menggunakan Metode Regresi Linear," *J I M P - J. Inform. Merdeka Pasuruan*, vol. 2, no. 3, pp. 45–56, 2017, doi: 10.37438/jimp.v2i3.79.
- [44] I. Siradjuddin, "Machine Learning: Linear Regression," *Politek. Negeri Malang*, 2018, doi: 10.13140/RG.2.2.21765.65761.
- [45] M. K. Sri Ratna Sulistiyanti, FX Arinto Setyawan, *Pengelolaan Citra; Dasar dan Contoh Penerapannya*. TEKNOSAIN, 2016.
- [46] M. Pulung Nurtantio Andono, Totok Sutojo, *Pengolahan citra digital*. Yogyakarta: Penerbit Andi, 2017.
- [47] R. E. W. Rafael C. Gonzalez, *Digital image processing*, Third. Upper Saddle River, New Jersey 07458: Pearson Prentice Hall, 2009.
- [48] W. Kaswidjanti, H. Jayadianti, and E. A. Malik, "Aplikasi pengenalan bendera negara menggunakan histogram citra," vol. 2011, no. semnasIF, pp. 42–49, 2011.
- [49] S. Gao, L.-T. Chia, and X. Cheng, "Web image concept annotation with better understanding of tags and visual features," *J. Vis. Commun. Image Represent.*, vol. 21, no. 8, pp. 806–814, 2010.

- [50] A. W. Kusuma and R. L. Ellyana, "Penerapan Citra Terkompresi Pada Segmentasi Citra Menggunakan Algoritma K-Means," *J. Terap. Teknol. Inf.*, vol. 2, no. 1, pp. 65–74, 2018.
- [51] I. D. Kurniawati and A. Kusumawardhani, "Implementasi Algoritma Canny Dalam Pengenalan Wajah Menggunakan Antarmuka GUI Matlab," *Res. Surabaya*, 2017.
- [52] D. Putra, *Pengolahan citra digital*. Penerbit Andi, 2010.
- [53] A. Kadir and A. Susanto, *Teori dan Aplikasi Pengolahan Citra*. 2013.
- [54] U. Ahmad, *Pengolahan citra digital & teknik pemrogramannya*, 1st ed. Yogyakarta: Graha Ilmu, 2005.
- [55] L. B. Prianggodo and R. N. Rohmah, "Perancangan Object Tracking Robot Berbasis Image Processing Menggunakan Raspberry Pi." Universitas Muhammadiyah Surakarta, 2016.
- [56] G. Pangarso and R. Kusumaningrum, "SEGMENTASI CITRA OBYEK BUAH MENGGUNAKAN OHTA COLOUR SPACE DAN NIBLACK ADAPTIVE LOCAL THRESHOLD," Universitas Diponegoro, 2016.
- [57] B. S. Morse, "Thresholding," *Brigham Young University*. Brigham Young University, Provo, Utah, 2000.
- [58] Python.org, "What is Python? Executive Summary." <https://www.python.org/> (accessed Apr. 21, 2022).
- [59] J. Enterprise, *Python untuk Programmer Pemula*. Elex media komputindo, 2019.
- [60] M. Abadi *et al.*, "TensorFlow: a system for Large-Scale machine learning," in *12th USENIX symposium on operating systems design and implementation (OSDI 16)*, 2016, pp. 265–283.
- [61] Roboflow, "Give your software the sense of sight," *roboflow*. <https://roboflow.com/> (accessed Jul. 21, 2021).
- [62] S. Alexandrova, Z. Tatlock, and M. Cakmak, "RoboFlow: A flow-based visual programming language for mobile manipulation tasks," in *Proceedings - IEEE International Conference on Robotics and Automation*, Jun. 2015, vol. 2015, pp. 5537–5544, doi: 10.1109/ICRA.2015.7139973.
- [63] T. Pessoa, R. Medeiros, T. Nepomuceno, G.-B. Bian, V. H. C. Albuquerque, and P. P. Filho, "Performance Analysis of Google Colaboratory as a Tool for Accelerating Deep Learning Applications," *IEEE Access*, Oct. 2018, doi: 10.1109/ACCESS.2018.2874767.
- [64] T. Kluyver *et al.*, *Jupyter Notebooks-a publishing format for reproducible computational workflows.*, vol. 2016. 2016.
- [65] D. Sugiyono, *Metode penelitian pendidikan pendekatan kuantitatif, kualitatif dan R&D*, 6th ed. Bandung: Alfabeta, 2013.

- [66] J. Han and M. Kamber, "Data mining: concepts and techniques, 2nd," *Univ. Illinois Urbana Champaign Morgan Kaufmann*, 2006.
- [67] Sanjay.M, "Why and how to Cross Validate a Model?," *Towards Data Science*. <https://towardsdatascience.com/why-and-how-to-cross-validate-a-model-d6424b45261f> (accessed Sep. 10, 2022).
- [68] M. Kuhn and K. Johnson, *Applied predictive modeling*, 5th ed., vol. 26. New York: Springer, 2013.
- [69] D. C. Hsiou, F. Huang, F. J. Tey, T. Y. Wu, and Y. C. Lee, "An Automated Crop Growth Detection Method Using Satellite Imagery Data," *Agric.*, vol. 12, no. 4, 2022, doi: 10.3390/agriculture12040504.
- [70] A. Widarjono, *Ekonometrika: Pengantar dan aplikasinya*. Yogyakarta: EKONISIA, 2009.
- [71] W. Sulaiman, *analisis regresi menggunakan SPSS*, 1st ed. Yogyakarta: Andi, 2004.
- [72] W. W. Chin, "The partial least squares approach to structural equation modeling," in *Modern methods for business research*, vol. 295, no. 2, Mahwah, NJ, 1998, pp. 295–336.
- [73] A. De Myttenaere, B. Golden, B. Le Grand, and F. Rossi, "Mean absolute percentage error for regression models," *Neurocomputing*, vol. 192, pp. 38–48, 2016.
- [74] D. Lewis Colin, *Industrial and business forecasting methods : a practical guide to exponential smoothing and curve fitting*. Butterworth & Co Publishers Ltd, 1982.
- [75] D. De Silva and D. Alahakoon, "An artificial intelligence life cycle: From conception to production," *Patterns*, p. 100489, 2022.
- [76] B. Vrigazova, "The proportion for splitting data into training and test set for the bootstrap in classification problems," *Bus. Syst. Res. Int. J. Soc. Adv. Innov. Res. Econ.*, vol. 12, no. 1, pp. 228–242, 2021.